PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Design and Analysis of Load-Balancing Switch with Finite Buffers and Variable Size Packets

Yury Audzevich

Advisor:

Prof. Yoram Ofek

Università degli Studi di Trento

Co-Advisor:

Prof. Renato Lo Cigno

Università degli Studi di Trento

December 2009

Abstract

As the traffic volume on the Internet increases exponentially, so does the demand for fast switching of packets between asynchronous high-speed routers. Although the optical fiber can provide an extremely high capacity, the Internet switches still remain the main point of traffic bottleneck. The packet switching time may run up to nanoseconds in such routers with more than thousands ports, each processing at 10 GB/s. Even modern extremely fast processing units are not capable to satisfy these needs. It is well known that switching of such a high volume of traffic from input to output requires large buffers and fast processors to perform the header processing, complex scheduling and forwarding functions. Although a large number of switching architectures is presented on the market, the considerable part of them is either not scalable or reach their limits in power consumption and complexity. Therefore, novel and extremely scalable switching systems are essential to be investigated.

The load-balancing switching approach is simple, and therefore, may be capable of performing the switching and forwarding from all inputs to all outputs simultaneously with low complexity and high scalability. Since this simple approach has distributed topology (each component of the switch is controlled by an individual chip) and do not require fast switch control units, primarily because each stage is independent and it makes its own distributed calculations, it becomes a perfect candidate for the future practical deployment. The load-balancing switching architecture, considered in this thesis, is proved to have high potential to scale up while maintaining good throughput and other performance characteristics. Additionally, the load-balancing switching architecture can effectively resolve the important problem of packets mis-ordering which can appear due to the distributed structure of the system. Unfortunately, in the research conducted previously, some of the mentioned characteristics were obtained under a set of strong assumptions. In particular, it was assumed that all the packets transmitted through the system have equal length, traffic is admissible and central stage buffers are infinite. On the other hand, due to the distributed control the switch is not able to control and maintain a necessary amount of traffic transmitted from stage to stage inside the switch.

The following Ph.D. thesis analyzes behavior of the load-balancing switch equipped with finite central stage buffers. Due to this fact the LB switch will always have a possibility to drop a packet due to an overflow. In this work we first analyze the packet loss probability in the central stage buffers while considering packets of the same length (data cells). The

analysis will be performed for both admissible and inadmissible traffic matrices. The obtained results show that the packet loss can have a significant influence on the overall LB switch performance if inputs of the switch are overloaded.

In order to present more realistic scenario, the packet loss analysis was performed in the switch with variable size packets. It is considered that most of the internet switches are operating on the cell-based level (to increase buffer utilization), that means that arriving variable size packets are segmented at inputs and reassembled at outputs. The issue of possible cell and correspondingly a packet loss inside the switch can introduce some significant posterior problems to the load-balancing switch reassembly unit. In order to evaluate packet loss we assumed Markovian behavior to be able to use numerically efficient algorithms to solve the model. The mathematical model characterizing inhomogeneous input traffic presented inside the thesis gives the most precise way of packet loss probability evaluation. Unfortunately, the high complexity of this model results in irresolvably complex Markov chains even in case of very small switches. Consequently, as a next step, we performed the analysis with fast solution procedures using a restrictive assumption of identical stochastic processes at all inputs. The final results allowed us to conclude that a single cell drop at the central stage buffers cause the whole packet removal and, the packet loss probability inside the system can be extremely high in comparison with the corresponding cell loss. Another important issue observed from the analysis is the difference in packet loss probabilities depending on the traffic traversing path, e.g. sequential number of input, central stage buffer and output of the switch. This property makes more complex the evaluation of the loss probabilities for large switch sizes. The last but not the least issue observed by our analysis was the instability, congestion and large delays appearing at output re-sequencing and reassembly unit due to the the central stage packet loss.

In order to cope with such a behavior, we proposed the novel algorithms which are able to efficiently minimize/avoid packet loss at the central stage buffers of the switch. For instance, the novel minimization protocol is introducing an artificial buffering threshold at the central stage buffers in such a way that packets at the input stage are are dropped in case the actual central stage buffers occupancy is above the threshold. The results show that due to possible packet removal at the input stage of the switch, the overall packet loss probability is significantly reduced. Similarly to the loss minimization service protocol, the novel NoLoss load-balancing switch operates while using information from both inputs and central stage buffers, and allows a packet transmission through the switch only if the central stage buffers have enough space to accept it during the current and the following time slots. In order to minimize communication overheads, the algorithm was implemented by means of centralize controller. Finally, such kind of management helped us to reach the lower boundary in the overall packet loss probability and resolve some other important issues of the switch, like, for instance, the congestion problem of the output reassembly unit.

Keywords

[Scalable switch design, load-balancing switch, packets mis-ordering, cell loss, packet loss, congestion avoidance protocol, noloss switch]

Acknowledgments

I am extremely grateful to my advisor Prof. Yoram Ofek for his incredible wisdom, immense knowledge, motivation and enthusiasm provided during my Ph.D. study in Italy. His guidance helped me in all the time of research, and his continuous support encourage me not only in professional but also in moral respect. I could not have imagined having a better advisor for my Ph.D. study.

Besides my advisor i would like to thank to my co-advisor Prof. Renato Lo Cigno, for his relentless support, supervision, advices and his continuous involvement into my research activities and assistance in resolution of non-scientific concerns. I am grateful in every possible way and hope to keep our collaboration in the future.

Many sincere thanks go in particular to Prof. Miklós Telek for his continuous help and guidance. I am really grateful to him for the offered opportunity of visiting and working with the members of the Stochastic Modelling lab at the Technical University of Budapest. Both Prof. Miklós Telek and all the members of his group have created inexhaustibly energizing, motivating and supportive atmosphere in the lab. My special thanks are dedicated to my friend and colleague Levente Bodrog for sharing his experience with me in stochastic modelling, his productive discussions and readiness to help.

Finally, i would like to express my sincere gratitude to Prof. Bűlent Yener without whom the investigations on the topic of this thesis might not even have been initiated. His experience, wise advices and assurance of success have enlightened me in the first glance of the research in the load-balancing switching field.

For all good time and support during my Ph.D. years I am grateful to my colleagues and friends, especially to: Danilo, Vladimir, Dmitry, Olga, Nikolay, Ivan, Andrey, Raman, Thang, Huong and Marcin.

Last but not the least, I would like to thank my family for their invaluable help and support: my parents, and my wife Tatsiana.

This work is dedicated...

To the memory of my departed uncles, grandparents and Yoram.

Contributions and publications

This work has been developed in collaboration with various people and in particular with: Yoram Ofek, Renato Lo Cigno, Miklós Telek, Bűlent Yener, Levente Bodrog, Danilo Severina, and Giorgio Fontana.

This thesis makes the following contributions:

- Presents an overview of the currently deployed switching technologies on the market;
- Reviews the scalability limitations of the load-balancing switching architectures;
- Performs analysis of the load-balancing switching architectures with fixed and variable size packets;
- Presents mathematical models and simulator for the packet loss evaluation inside the switch;
- Proposes design and analysis of protocols which minimize/avoid the central stage packet loss of the load-balancing switch;

Part of the material of the thesis has been published (to appear) in various conferences, journals and technical reports (in order of appearance):

- [69]: Y.Audzevich, Y. Ofek, M. Telek and B. Yener. Analysis of Load-Balanced Switch with Finite Buffers. In *IEEE GLOBECOM' 08*, Dec. 2008.
- [6]: Y. Audzevich and Y. Ofek. Assessment and Open-issues of the Load-balanced Switching Architecture. In *IEEE FGCN'08*, Dec. 2008.
- [4]: Y. Audzevich, L. Bodrog, M. Telek, Y. Ofek, and B. Yener. Variable Size Packets Analysis in Load-Balanced Switch with Finite Buffers. In *Technical report, TU Budapest*, Apr 2009.
- [5]: Y. Audzevich, M. Corra, G. Fontana, Y. Ofek and D. Severina. Energy Efficient All-Optical SOA Switch for the Green Internet. In *FOTONICA'09*, May 2009.
- [3]: Y. Audzevich, L. Bodrog, Y. Ofek and M. Telek. Scalable model for packet loss analysis of Load-Balancing switches with identical input processes. In *IEEE ASMTA'09*, Jun 2009.

- [2]: Y. Audzevich, L. Bodrog, Y. Ofek and M. Telek. Packet Loss Analysis of Load-Balancing Switch with ON/OFF Input Processes. In *EPEW'09*, Jul 2009.
- Y. Audzevich and Y. Ofek. Overview and Evaluation of Load-Balancing Switches. Submitted to *Computer Networks* Journal.
- Y. Audzevich, L. Bodrog, Y. Ofek and M. Telek. Packet Loss Minimization in Load-Balancing Switch. Submitted to *IEEE ASMTA '10*.
- Y. Audzevich, G. De Blasio, R. Lo Cigno, M. G. Frecassetti, F. Granelli and D. Kliazovich. SoPSim: Simulation of Circuit Emulation over Packet for Microwave Radio Links. Submitted to *IEEE EuWiT '10*.

Whenever results of any of these works are reported, proper citations are made in the body of the thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

According to the latest statistics, the amount of users connected to the Internet is growing extremely fast every year. For instance, the World Internet Usage Statistics organization reports that during the period from 1992, when Internet was pulled to the general public, up to the present time the number of Internet users became 100 times larger. In addition to that, demand on the "bandwidth-hungry" applications is increasing every year. Various types of applications represented in the modern network cover all the aspects of human life. In particular, the network resources are widely used in business, science, and mostly at entertainment. The number of internet users can be easily divided in both the regard to the time spent online and the amount/type of applications they use. Technological developments in both the electronics and telecommunications have extended broadband capacity of private networks from a limited "Internet browsing" function to a fundamental entertainment capabilities while exploring services such as Video-on-Demand and IPTV.

1.1 The Context

In order to provide the increasing demand to the network resources novel technologies for realization of transmission medium as well as for the Internet routing are introduced.

In spite of the cabling costs, the optical fiber is still remain the fastest and most reliable medium for data transmission. The situation is completely opposite with regard to the the Internet switching/routing. The traditional ways of packet switching are designed to connect multiple area networks (LANs, WANs, etc.) and forward asynchronous traffic between the communication links by means of routing protocols. Usually the transmission decision is evaluated by means of centralized controller, which is not considered to be a scalable solution for switch sizes of at least 1000 ports. Moreover, in order to maintain operating such an architecture, a considerable financial (extremely high power consumption [43]) and administrative (by involving network administrators) investments should be constantly done. In this context, switches with distributed control are looking more attractive. In such a system, the packet scheduling decision is achieved by a separate block which is independent from all other system's blocks. Although the on-chip control is not always considered to be simple, it gives the possibility to easily scale up the system without significant drop in performance.

In this thesis we focus on the representation of the various switching/routing architectures based on the distributed control. The initial part of this thesis, e.g. chapter 2, will show the technologies used for a scalable switch/router implementation. The main task of each network router can be simply reduced to routing and forwarding of incoming information. It is considered that a router is operating in both control plane (the process of the most optimal interconnection decision is performed) and forwarding plane (where the information is actually sent). However, it is possible that due to internal router topology, the complete routing decision cannot be specified at the control plane at once. This rule is mostly related to the routers implementing non-deterministic or adaptive forwarding. In contrast to the assignment of a fixed traversing route, such routers can dynamically decide which hop(s) is(are) more preferable for optimal information forwarding. The main drawback of routers with non-deterministic routing can be the capacity expensive communication overheads, which in some cases can be even larger than the amount of information transmitted. Additionally, such routers are a subject of extremely large traversing delays and low throughput.

In contrast, the forwarding decision which is performed in distributed routers with deterministic management can be specified after a decision making phase. The load-balancing (LB) switch can be attributed to such a category of routers. It was firstly presented in [16, 17, 40] where it is implementing extremely simple ingress-egress interconnection scheme, realized in round-robin run of crossbars. In general LB switch is composed of three stages, each of the stages contains a certain amount of buffering. The interconnection between the stages is performed by means of crossbar switches which are synchronized to the rest of the system. Due to the fact that decision about packet forwarding is deterministic at each time point, the system does not use any additional overheads for the forwarding decision phase. All these features deploy both extremely high scalability and good performance characteristics of the LB switch. All the positive features as well as novelty of the proposed approach makes the LB switch the main focusing point of this thesis.

The initial research on the topic of load-balancing was tackling only specific set of problems under the predefined set of assumptions (described also in section 2.2). Among the first significant results shown in [17] and [18] was the fact that under certain assumptions the switch can achieve high throughput (up to 100%) and low packet traversing delay. However these results were obtained under consideration that all the packets have equal length, traffic is admissible and central stage buffers are infinite. On the other hand, the important issue of packets mis-ordering was investigated into details in [15, 40, 46, 57, 75]. It is important to mention that some of the architectures to resolve packets mis-sequencing require extra control, introducing different overheads (communication and computational), that basically increases the control complexity of the LB switch. However, keeping correct sequence of packets through the system avoids unnecessary retransmissions of packets in the network protocol layer [10].

1.2 Problems, Solutions and Thesis Structure

The initially considered set of assumptions makes impossible to analyze the LB switch behavior under conditions existing in the real Internet router [23]. In particular the maximum system's throughput (100%) can be reached only in the ideal case with infinite central stage buffers. Taking into account the fact that practical networking devices will always have a limited space for packet storage and forwarding, in the future we perform analysis only for LB switches with finite amount of buffering (Chapter 3).

The first attempt to analyze (by means of simulations) the central stage packet loss was performed in [65]. In contrast to this work we propose in [69] a novel mathematical model, which allows to evaluate central stage packet loss for any set of parameters N - switch size, B - buffer size and traffic matrix A - which is also called an arrival rate matrix. The results of packet loss evaluation as well as the model itself will be presented in chapter 4. The first attempt of packet loss probability evaluation had one significant drawback. Throughout the paper [69] we assumed that all the packets arriving to the inputs are of the same size (also named as data cells). It is well known that due to a large variety of online applications, the Internet traffic is transporting packets with variable length. In order to transmit such a packet through a standard Internet switch arriving packets are segmented into a fixed size data cells at ingress ports and reassembled at egress ports after cell-by-cell forwarding. Holding this assumption for the LB switch we have performed the analysis of the packet loss probability with the assumption of variable size packets [4]. The mathematical model presented in chapter 5 section 5.2 shows extremely high complexity of given analysis. Since the LB switch itself is considered to be a highly scalable architecture, it is quite important to yield solution which is capable to converge extremely fast for large switch sizes and have a good precision. In order to satisfy these needs and simplify the mathematical model, some restrictive assumptions were applied to the arriving traffic. In particular we assumed the identical behavior, e.g. identically distributed arriving traffic, of all inputs. As a result the final complexity of the method was described as $O(N+1)$ (chapter 5 section 5.4), which allowed us to perform evaluations for switches with large number of ports. The obtained mathematical model allowed us to evaluate packet loss probabilities for various packet and interpacket lengths. It was shown by results that the packet loss of the switch is always higher than the corresponding cell loss. This can be explained by the fact that a single cell drop is causing the whole packet removal.

In addition to the mathematical model, as a tool to better understand and optimize performance and/or reliability of the LB switch, the LB switch simulator was also extensively used. In contrast to the modelling, the simulation tool allowed to explore obtained results and get understanding of the future behavior of the system when conditions close to real, e.g. heavy-tailed traffic matrices or limited amount of buffering at the output, are applied (like it is presented in section 5.5.4).

In addition to the problem of cell and packet loss at the central stage buffers, the analysis reveals a set of novel issues related to the re-sequencing and reassembly units (RRU). In particular, the amount of memory resources at RRU that can be held by congested data cells is expanding with increase of the central stage packet loss. Due to the fact that the LB switch does not perform any control on the traffic transmitted form a stage to another one, the data cells of incomplete packets (dropped at the central stage) arrive to outputs and should be handled by the processor of RRU. As a result of such unreliable packet transmissions a certain number of memory locations in the buffer is wasted by the cells of incomplete packets waiting for reassembly.

To resolve the system instability caused at RRU a set of novel protocols for the LB switch was proposed (see chapter 6). In order to reduce buffers capacity wastage produced by the central stage buffers congestion, it is preferable to predict a possible packet drop in inputs in order to remove it as a complete unit. Two types of algorithms were proposed to resolve this issue. The packet loss minimization service protocol, in section 6.1, bounds the amount of traffic transmitted from inputs to central stage while positioning a virtual threshold in central stage queues. This countermeasure, although, does not avoid central stage packet loss completely, but allows to reduce the amount of incomplete packets in the system. The second approach presented in section 6.2, in contrast, implements a management scheme which completely avoids central stage packet loss. Due to this fact, the overall packet loss of the system is always lower than in the traditional system. Finally these solutions allow to allocate the amount of buffering resources at each stage in order to perform the efficient reassembly process.

Chapter 2

State of the Art and Related work

Internet represents a complex topology asynchronous network of a very large numbers of queues that is continuously expanding in size and capacity. IP routers should be periodically upgraded to support the growing Internet capacity, enlarging exponentially every 18 month [40]. None of the recently operating single stage routers is capable of sustaining thousands of operational ports with simple control mechanism. Incoming asynchronous packets require fast routing processors and large buffering structures in order to lookup header [11] and hastily make fast output transmission decision. The forwarding table, that is stored in processor unit, looks for destination entries that best matches the destination network address of the packet. Practically, the router operating at high speeds are capable of performing millions of lookups per second. For routers of our interest, with size more than thousands ports, and supporting at least 10 GB/s per port rate, the packet switching time value may run up to nanoseconds [45]. Even the super fast processors are not capable to satisfy these needs. It is accepted estimation that single stage routers almost reach their capacity, and cannot scale up to the large sizes. Routers with distributed control [68] are meeting the requirements of designing a new scalable router. Usually distributed routers do not require relatively fast processing units [33] because each stage is independent and it makes its own self-calculations. Eventually, some disadvantages are highlighted on the use of distributed switching systems, and some of these problems may introduce significant changes in the operation stability. Some of the proposed solutions require more complex interactions with these distributed switches that can particularly affect the system scalability. As a result, we reveal the fact that not all distributed schemes are yet scalable and suitable for real Internet conditions support. One of the major motivations of this chapter is to analyze the distributed architectures represented so far in literature. This chapter aims at providing a comprehensive analysis of the switching architectures and main concepts behind switching architectures common to all switch vendors. The chapter begins by looking at most popular switching solutions and representation of their mechanisms to switch data. Then, we chose the most scalable solutions capable to provide good performance characteristics in real networks.

To start the discussion we refer to output buffer switches [36], which, at the moment are not of the great interest from the scalability prospectives. In spite of the fact, that switch is able to provide guaranteed throughput, the use of large memory speedup makes it non realizable for more than hundred of ports. Completely another interest is raised by input buffer switches where some scheduling decision algorithms are performed. To eliminate head-of-line blocking in the inputs, buffers with random access and virtual output queuing [36] structures get under the way. Apparently, input buffer switch requires scheduling algorithm in order to match and interconnect inputs with outputs. It is well known that both centralize controller or a distributed management schemes can be used to resolve ingress-egress matching problem. As our interest is given mostly to distributed-control algorithms (e.g. operating on per-port basis), next, we will present two particular implementations on the example of Parallel Iterative Matching (PIM) [62] and iSLIP [49] algorithms. We concentrate our attention on the input buffer switch defining two main implementation branches. The input buffer switches with fix routing schemes (like PIM and iSLIP) will be described first. However, the significant interest will be given also to representation of schemes with non-deterministic packet routing like presented in [8, 48, 54, 55].

As a completely different approach, the Time-driven switching (TDS) [7] deploys pipeline forwarding inside the network using global common time reference. Thanks to coordinated universal time, all the switches in the network are synchronized and use negligible amount of buffering for packet transmission. TDS scheduler reserves transmission resources for a dataflow in advance, thus allowing packet header processing to be excluded.

The remaining part of the thesis, is dedicated to the novel LB switching architecture recently appeared in the research [32]. The careful attention is devoted to the fact that the LB switching architectures are considered to be highly scalable, have a simple distributed control and almost no scheduling overheads. Our goal is to present a different perspective on load-balancing switches, while considering their advantages and drawbacks. In spite of the fact that quite large number of publications was presented on the topic, in this chapter we introduce several new issues which were not mentioned previously. In the following, we first show initial LB switching architecture with the basic assumptions and main disadvantages. Next we shortly list the most crucial drawbacks related to the LB switching architecture. Among these, we focus on the most important issues like system scalability and packets mis-sequencing. Additionally, we analyze how this two issues are influencing one another and can be maintained all together in the appropriate level. We merge all available solutions for packets mis-sequencing into a single section with extra descriptions on the proposed solutions. In the section devoted to the systems' scalability analysis we evaluated computation and communication overheads which can appear in the proposed designs. In addition, we introduce a novel problem of cell/packet loss inside the switch. This issue has strong dependency on the system stability and its performance.

2.1 Classification of Switching Architectures

By definition, a network switch is considered to be a device that performs transparent ingress-egress bridging at up to speeds of a hardware. Topologically each switch can be viewed as a network located in a box. All the systems making internal switching can be further classified as input or output buffered depending upon the position of buffers. As it was mentioned before, our main requirements are to find switching architectures being capable of scaling up to at least of thousands ports and support of minimum 10 Gb/s per port rate.

Although a switch using output queuing has better throughput and delay performance characteristics than switches using input queuing, the hardware costs can be enormous for large output buffer switches. Moreover, output buffer switches are fairly complex and require large speedups of transmission buffers due to simultaneous arrival of more than one input packet for the same output [36]. On the other hand, it has been also shown that the throughput of a switch with input queuing is only 0.586 of the full capacity. The main reason of this poor throughput is due to head of line (HOL) blocking where packets at the head of input queues contend for the same output while the packets destined for free outputs are waiting in a line. Due to the implementation simplicity and total costs, it is still feasible to build large switches using input buffering techniques. Some ingenious solutions were proposed to solve HOL blocking problem [50].

In particular, instead of using first-in-first-out (FIFO) policy in each input the virtual output queuing (VOQ) scheme creates per output queues at each input. In such a way each packet occupies a virtual queue corresponding to the output, and none of the packets are blocked. Moreover, input buffer switch requires common scheduler. In practice, scheduler's complexity is in the strict relation with the system performance. Henceforward, our attention will be dedicated to algorithms distributing decision complexity among the ports, allowing the neglect of centralized control for an input buffered switch. In the following sections we will represent also the load-balancing [17, 18, 40] and Time-Driven [7] switching principles. In the upshot, some relevant to this thesis architectures, like Metaring [21] and MetaNet [54, 55] and, switching architectures build upon Combinatorial Designs [71, 72], will be depicted. Table 2.1 presents the classification of the most space-time scalability appropriate solutions in the research.

The Internet routers are making their routing decisions based on the knowledge of the network's topology and it's conditions. In a simple configuration, fixed or adaptive routing schemes are possible. Further, we will apply these principles of fix and non-deterministic routing for the "internal network-based" routers of the input buffer switch.

Space ***Time***	**Sending over predefined routes**	**Sending over non-deterministic routes**
Schedule	**Time-Driven Switching** [7] (Small memory, Connection oriented, good for streaming media support)	**NOT practical**
Without routing schedule	**input buffer switches with:** 1) PIM [62], 2) iSLIP [49], 3) Load-Balancing switching [17, 18, 30, 39] (Large memory, low loss, long delay).	**Single input buffer switches with placed "in a box" approaches of:** 1) Hot-potato Routing [8], 2) Deflection Routing [48], 3) MetaNet and Convergence Routing [54, 55, 71, 72].

Table 2.1: Potentially scalable switching architectures

A packet transferring with predefined route can be possible in the routers with schedulers or in the systems where predefined configuration is applicable. On the other hand, the architectures where packet traverses along a non-deterministic route can be present. Deflection routing [48] and MetaNet Convergence routing [54, 55], if to be positioned "in a box", can be a good example of a non-deterministic routing. Figure 2.5 shows the use of Hot-potato routing [8] and all the other non-deterministic routing-schemes located into a single input buffer router. Table 2.1 strictly differentiates and underline importance of these two routing approaches.

2.1.1 Distributed designs with fix routes

Parallel Iterative Matching [62]. The following matching algorithm was implemented in Input buffering switch, using memory with random access [62]. The algorithm finds conflict-free pairing of inputs to outputs, only for pairs with a queued cell to transmit between them. Parallel iterative matching uses parallelism, randomness, and iteration to accomplish matching efficiently. The operational steps during one iteration are the following (Figure 2.1). First unmatched input sends request to every output for which it has buffered cell. If unmatched output receives any request, it chooses one randomly to grant. The outputs notify each input whether its request was granted. Finally if an input receives some grants, it chooses one to accept and notifies that output. It's possible that some inputs and outputs remain unmatched. In this case the algorithm runs another iteration excluding operations with previously matched inputs. The explicit schedule is built for each input-output pairings for each slot in a frame. Scheduler can be extended to allocate resources fairly when network is overloaded. Finally, distributed calculations make the system scalable.

iSLIP algorithm [49]. iSlip algorithm is based on PIM, with the difference that it uses

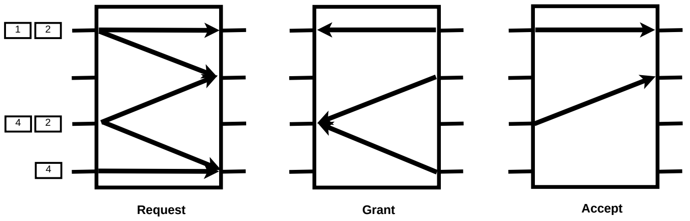


Figure 2.1: Parallel Iterative Matching: one iteration

rotating priority (”round-robin”) arbitration to schedule each active input and output. This fact improves performance, so for uniform traffic in can achieve 100% throughput with single iteration. In fact iSLIP is similar to round-robin matching algorithm with the difference that its not moving grant pointers unless grant is accepted. The operational idea is represented in Figure 2.2. It was found in [49] that for $N \times N$ switch it takes about $log_2 N$ iterations to converge. It has high throughput for uniform traffic. The algorithm is simple and for small switches arbiter can be placed on a single switch. For $N > 1000$ ports switch will have large computation overhead, and might not be highly scalable.

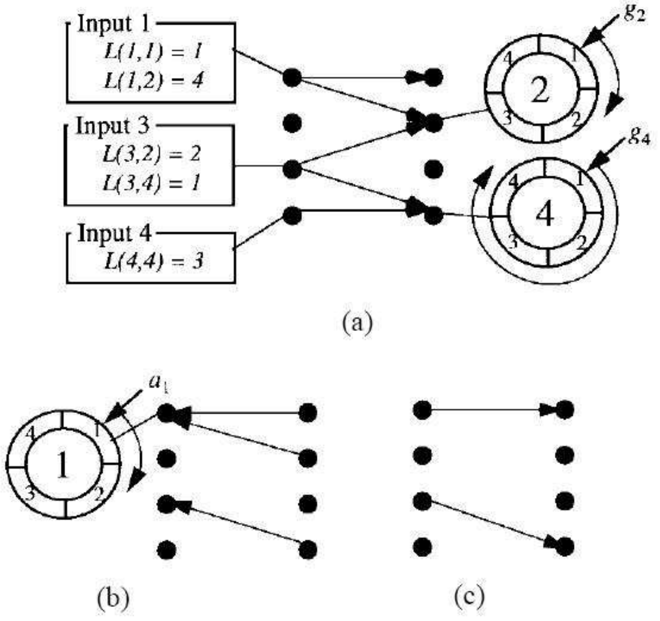


Figure 2.2: Representation of iSLIP matching algorithm

Load-Balancing switching architecture [17, 18]. The architecture is shown in Figure 2.3. It’s represented as a stage of buffers positioned in between of two identical crossbar switches. Each line card buffer at an intermediate input in the central stage is partitioned into N separate First-In-First-Out queues, one for each output. The authors of the basic scheme assume the packets of the switch to be of the same size, they call them simply packets [17, 40]. Each linecard inside the switch is synchronized and time

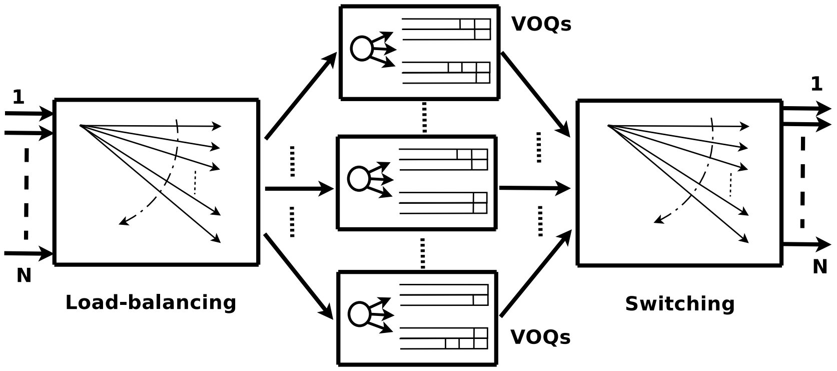


Figure 2.3: The basic LB switch

slotted. This implies that the only one cell can arrive and depart switch during the time slot. Finally each linecard is able to support strictly equal rate flows.

The operational idea behind the distributed $N \times N$ size architecture is to load-balance or spread uniformly the cells from the input along the VOQs of central buffering stage. Since every component is synchronized, input crossbar periodically interconnect each input to independent buffering units at the central stage. There, the cells positioned in the corresponding to the output port number VOQ. Later on, the VOQ will be served by the second crossbar switch to the output. Input and output crossbars configured similarly with the periodic round-robin connection pattern, linking input, central and output stages. The configuration is not based in the occupancy of the queues, both switching stages walk through a fixed sequence of configurations.

The basic LB switch is promised to be highly scalable. Due to predefined properties it should have a 100% throughput for a broad class of arrival admissible traffic (any traffic distributed uniformly between outputs) even without centralized scheduler. The basic scheme also promises to provide a low average delay even under heavy load and bursty traffic (mathematical representation can be found in [17, 18]). Absence of centralized control mechanism greatly reducing hardware complexity of the switch which makes it easier to implement. Therefore, this architecture was considered as a main candidate for our detailed investigation for the future.

2.1.2 Distributed designs with non-deterministic routes

MetaNet convergence routing [54,55]. The structure and operational ideas of MetaNet were first represented in [54]. It is a scalable local area network with an arbitrary topology. The principles of MetaNet are derived from two primitives. The first one refers to exchange of in-band hardware control signals in the network. Another property stands for virtual rings (VR) embedding (Rings and Thread links). These mechanisms are used to linearize the arbitrary topology network for provision of control and global order. Each VR can be constructed according to bidirectional buffer insertion or slotted ring [21].

The spatial reuse provides the ability to concurrently transmit over distinct segments of the ring. Hardware control signals are used to enable independent control functions in the network. Packets in the MetaNet can make short-cuts toward it destination (see Figure 2.4) in order to decrease routing distance. Another possibility is to make jumps on threads. Due to dynamic convergence routing ***MetaNet has no packet loss and capable of supporting real time traffic***.

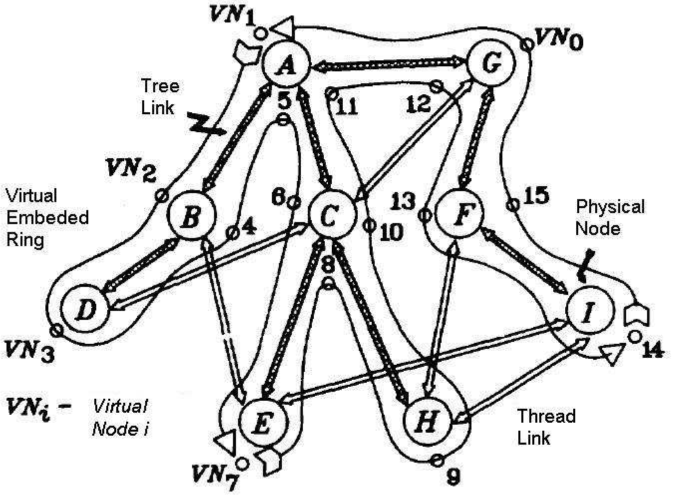


Figure 2.4: Tree embedded ring; short-cut ($VN_2 - VN_4$) and jump ($C - G$)

Manhattan street network [48]. The Manhattan Street Network (MSN) [48] is a self-routing regular topology, originally proposed for local and metropolitan area network applications. An MSN is characterized as a two-connected regular mesh with the nodes connected as rows and columns. Each node consists of a 2 × 2 crossbar switch that connects incoming links to outgoing links. The Clockwork Routing Scheme is a time-slotted system that enhances the MSN. It's includes a simple routing mechanism employed at intermediate nodes that prevents resource contentions, requires no re-sequencing of packets at the destination node and has comparable throughput to conventional routing schemes.

Deflection or "hot-potato" routing [8]. The nodes of Hot-potato routing network almost don't have any buffers to store packets in before they are moved on to their final destination. The single input buffer routers are used for operation. Each packet is constantly transferred until it reaches its final destination. The link bandwidth can support maximum one packet which is bounced around like a "hot potato," sometimes moving further away from its destination because it has to keep moving through the network [8]. This is the contrast to the "store and forward" routing where the network allows temporary storage at intermediate locations. The use of input buffer switch for Hot-potato routing is depicted in Figure 2.5.

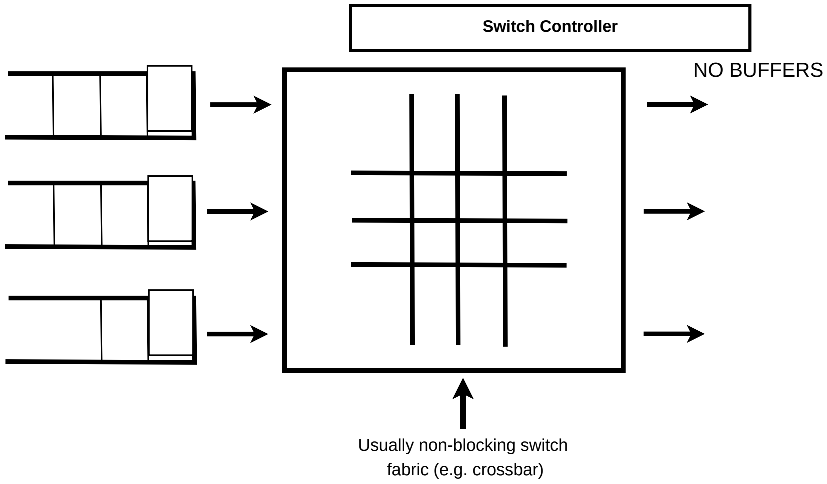


Figure 2.5: Use of single input buffer switch for Hot-potato, MSN and MetaNet routing

2.1.3 Time-Driven switching

Time-Driven switching [7]. In Fractional lambda Switching ($F\lambda S$), a concept of common time reference (CTR) using UTC (coordinated universal time) is introduced [7]. A UTC second is partitioned into a number of time-frames. Time-frames are switched at every $F\lambda S$ node with reference to the global CTR. A group of k time-frames forms a time-cycle; l continuous time-cycles are grouped into a *super cycle*. To enable $F\lambda S$, time-frames are aligned at the inputs of every $F\lambda S$ switch before being switched. After alignment, the delay between pair of adjacent switch nodes is an integer number of time-frames. The $F\lambda S$, using a global time scheduler, controlling occupancy of time frames. That helps to make provisioning for the future packet transmissions. The architecture reduces packet overhead processing, make usefully bandwidth allocation and efficiently support real-time services.

2.2 The Load-Balancing Switching Architecture

In the previous section, a great deal of attention was given to the distributed switching systems as well as to some routers using distributed control. The property to scale is considered to be the central focusing reason for this thesis. In particular, we paid a great amount of interest to the newly proposed LB switching structure. In the above section, we presented some basic knowledge related to the architectural operation principles and assumptions.

In the following section, we will present into details the assumptions and promises given to the initial architecture. We continue with operational principles of the system and short overview of open issues. Further, the detailed analysis of scalability and mis-sequencing problems will be done. In particular, we will analyze scalability of more

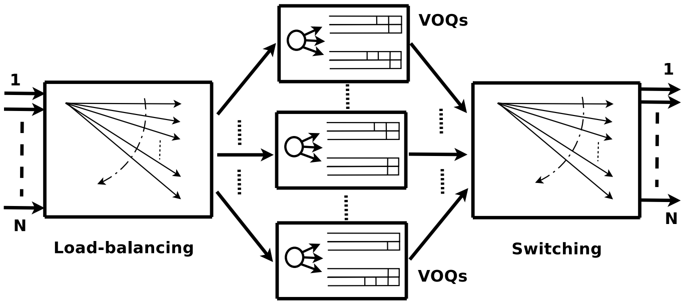


Figure 2.6: LB switch with single-stage buffering

complex switches with implemented feedback exchange links and which, correspondingly, require extra computation and communication overheads for proper functioning.

2.2.1 The architecture, assumptions and open issues

The basic single-stage buffering LB switch is shown in Figure 2.6 and consists of sets of buffers that are positioned in between of two identical crossbar switches [17, 40]. Each crossbar operates through similar predefined interconnection pattern in time according to the following rule:

$$j = (i + t)\ mod\ N, \tag{2.1}$$

where N is the number of ports in the switch, i is an input(intermediate input at the central stage) number which is interconnected with intermediate input at the central stage (output). Time inside the switch is slotted and each stage assumed to be synchronized with other stages. The transmission of a packet (or a cell) between the stages could be done only in the boundaries of a time slot, i.e., only one packet during each time slot can arrive or depart. Each buffer in the central stage is organized as a set of N queues in a way that there is one queue (VOQ) associated with each output.

The operating idea behind the N-by-N switch size with a single-stage buffering architecture [17] is to load-balance packets from the inputs along the VOQs of the central buffering stage. Then, the packets are sent to the destination output ports. Arriving packets are switched instantly and there are no buffers inside the crossbars [17, 18].

The main purpose of this section is to highlight the initial LB switch features and to provide the proper background knowledge for the comparative analysis in this chapter further on.

Assumptions. Concerned researchers, such as [16–18, 39, 40] have **assumed** that all of the LB switch packets are of the same size and they simply call them *packets.* Although in the real Internet world packets have variable length, yet these assumptions have been idealized and several studies were built upon.

With this thesis, we will be calling packets of the same size as ***cells*** and, we assume them to be immediate multipliers of the variable size packets. For LB switch support of variable size packets the multi-stage buffering scheme can be used. Usually, input stage buffers are constructed in the similar manner like the central stage VOQs, in order the Head-of-Line blocking to be avoided [18, 40]. Another significant statement is relevant to line card synchronization. For each line card the common slotted time is used. Indeed, this assumption implies that the only one cell can arrive and depart switch during the time slot. Finally, each line card is able to support strictly equal rate flows. As one of the major assumptions to permit achievement of high throughput [39] for the initial single/multi-stage LB switch is traffic **admissibility**. Precise definition of this traffic type can be found in the section regarding *input traffic matrixes*.

The listed above fundamental assumptions can be found in each further work regarding LB switching. Definitely, recent schemes can use some specific assumptions due to their complexity (like symmetric interconnection pattern for crossbars); if there any of them will be found, they will be mentioned in the corresponding part.

Promises. Taking into consideration publications [16, 17, 39, 40] the LB switch is promised to be highly scalable (due to low scheduling complexity), and have 100% throughput for a broad class of arrival traffic like stationary, stochastic and weakly mixing input sequences [38, 53]. Absence of centralize control in initial schemes greatly reducing overall communication and computational overheads while providing cells arrival in their original order [39, 40]. It's important to note that high throughput guarantees could be granted only in the case of infinite central stage buffers [18, 39] and assumptions of traffic admissibility. Further on, the degradation of throughput for the LB system with finite buffers will be examined in [65]. Initial LB scheme also assures to give a low average delay for the heavy traffic load under uniform, bursty and hotspot traffic [17, 39, 40]. Support of priority schemes and multicasting with fan-out splitting are realistic to implement in LB switch [39, 40]. Due to low hardware complexity, practical construction of the LB switch while using electronic components [40] (buffers and crossbars) is feasible [1, 32].

Recent modifications [15, 46, 57, 75] of the LB switch introduce additional overheads and make architectures more complex. However, the promises given to these architectures under highlighted conditions are equivalent (high scalability, low delay under various traffic matrices, high throughput). In following we would like to study the fulfillment of the guaranties when the size of the switch is considerably large (10^3 -10^4 ports).

Making the upshot of this section we repeat the significance of the background. Each of the next sections and chapters will rely on the standard assumptions represented above for traditional switch. Unfortunately, some of the claims given by authors in initial papers [17, 18, 40] are true for some specific set of assumptions. Therefore the scope of our interests concerns also behavior of the system in the conditions which are close to real ones.

Open issues

Cells mis-sequencing in the output. The first problem defined by early papers of C. S. Chang, N. McKeown and I. Keslassy is an issue of cells mis-sequencing experienced at output stage of the switch. A set of solutions were proposed. Conditionally, we distinguish two different approaches to overcome the problem. The first set of algorithms are competing against cell mis-sequencing in the input and central stages. These algorithms use a special interconnection scheme to gain some feedback information about the cells in the input. Among the examples we highlight: the Padded Frames [34] algorithm used for initial scheme, the Mailbox Switch [15], the Concurrent Matching Switch [46] ,the frame-aggregated concurrent matching switch [47], the Contention and Reservation Switch [75], the two-stage switch with novel feedback mechanism [73], and the three-stage switch [74]. The second approach introduces algorithms using so called re-sequencing buffers in the output stage. As a rule, these operational algorithms are much simpler, however, hardware implantations require large buffering structures. The latter architectures and algorithms are the Uniform Frame Spreading [39], First Ordered Frames First [40], and the Byte-Focal Switch [57].

Scalability. The initial architecture (see Figure 2.6) was promised to be scalable. Indeed, the system is not operating well mainly due to mis-sequencing problem [39, 40]. The proposed solutions as a rule promote some modifications of the system hardware or operational algorithms, bringing in extra communication and computation overhead and making system not scalable any more (like [46, 57, 75]). Being focused on the scalability issue as the main one, the next section will represent a comparative study between these novel architectures.

Support of various traffic matrixes. The run-time performance measurements made by authors of the traditional LB switch [17,18], as well as for the Mailbox switch [15], the Contention and Reservation switch [75], the Byte-Focal switch [57], the Concurrent Matching [46] switch, and others is mostly based on assumptions of traffic ***admissibility***. Under these conditions the LB switch is capable to provide extremely high throughput, reaching 100% in the ideal case (central stage buffers are infinite). However, the burstiness of the real traffic [25, 44, 58, 63] can be the measure characteristic influencing the throughput. In most of the presented results, it is assumed that the LB switch traffic with equal size packets can be characterized by a single parameter - arrival rate matrix. As our analysis shows next, the initial scheme is not capable to overcome throughput loss for any kind of arrival traffic matrix if the central stage buffers are finite. Moreover, under a ”many-to-one” hot-spot traffic (introduced in [56] and section 4.5), the system will experience extremely large packet loss probabilities as it will be shown in Section 4.1. To characterize the LB systems with variable size packets at least three parameters should be defined (for characterization of geometrically distributed packet length and inter arrival periods). According to the analysis performed in Section 5.1, the initial set of parameters of traffic patterns has a high influence on the overall performance of the system.

Throughput. In fact the throughput of the system is not always stable, and depends on the admissibility of incoming traffic. Mathematical analysis of the throughput and average delay in [17, 40] presents practical interest for the future study. I. Keslassy in [31] had introduced the mathematical description of the interconnection capacity (and correspondigly the maximum possible throughput that interconnection can provide) related to the LB switch. The method allows the LB switch to be compared with ring, torus and hypercube interconnects. As a final result, it was proven in [31, 39] that for a given interconnection capacity, the load-balancing mesh has the maximum throughput.

Variable size packets support. Due to large variety of online applications, real Internet traffic use variable size packets. Since all the previous research is focused on the analysis of the LB switching architectures with fixed size data cells, the question of variable size packets support remains open. In the following sections we will consider this issue as one of the main focusing points for the future research.

Multicast and broadcasting of fixed and variable size packets. The amount of streaming Internet resources (like IPTV) has significantly increased recently. Traditionally these resources were using multicast and broadcasting strategies to perform data distribution. In this context the support of multicast policy inside a network switch (like for instance in [12]) goes as the important requirement. The authors of [16, 29] argue that LB switch can multicast cells with fan-out splitting for first-come-first-served policy. By definition [29], *"the fan-out splitting implies transmitting cells to destination one at a time slot"*. The proof is made for ATM switches which operating in the manner similar to cell-based LB switch.

The question of broadcasting and multicasting is still an open issue for the initial Birkhof-von Neumann switch [18, 40] as well as for Mailbox switch [15] and other resent architectures [46, 57, 75] using Round-Robin crossbars configuration principles described above.

No-fan-out-splitting case for LB switch was denoted in [60]. By definition *"no-splitting is an extreme where multicast cell is sent to all outputs in its fan-out in a single slot. Each input needs to maintain a separate VOQ for every multicast flow."* The polynomial time algorithm for the moderate number of multicast flows was presented in [60]. The static approach use the knowledge of multicast patterns and rates and decides in advance how each flow should be split (in the scheduler). The dynamic approach implies different flow division during different time slots. Through the algorithm presented in [60] authors use static approach, due to the fact that dynamic splitting requires an exponential number of VOQs even with known multicast pattern. The algorithm is also polynomial in the size of the switch when the number of multicast flows $O(\log N)$ or smaller. Finding an appropriate scheduler for a given set of flows is NP-hard in general and depends on the amount of unicast and multicasting in the system.

Fault tolerance of the system and ways of practical realization. As it was shown in [31, 39, 40], the initial LB switch architecture is able to recover from the system failure in case of linecard missing during a time period of 50 ms. In order to redistribute the arriving traffic with equal rate at each link, some reconfiguration algorithms and two implementation designs (all-optical and hybrid optical-electronic architectures) were proposed by I. Keslassy et. al. in [39,40]. For scalability purposes, the authors of [1,32,39] propose subdivision of the LB switch architecture into groups (so called *hierarchical mesh implementation* and *mesh decomposition as a sum of matches*). Each of newly proposed designs can be implemented using just optical components. However the designers can face the problem of extremely high complexity of all-optical buffers (in particular FIFO queues) [13, 14]. Practically these structures can be emulated with a set of simple 2×2 switches and fiber delay lines. For instance, in [14] the three-stage realization of optical FIFO queues was presented. It is noted that the queue with buffer of $2n - 1$ cells was realized by using $2n$ 2×2 switches with the total fiber length of $3 \cdot 2^{n-1} - 2$ meters, which is considered to be a complex solution for a single FIFO queue of size $2n - 1$.

2.3 LB Switch Designs Preventing Cells Mis-Sequencing

The amount of mis-sequencing occurred during traversing the switch has an important impact on the overall performance of the network [66]. This problem is also crucial for the LB switch operating with variable length packets, where unbounded amount of mis-sequencing can lead to throughput degradation and enlargement of average switch traversing delay. In general, cells will be mis-positioned if the related central stage VOQs are occupied at different levels.

First we introduce schemes where algorithms for preventing mis-sequencing at input and central buffer stages are implemented. Several algorithms use specific crossbar interconnection scheme to provide some feedback information about the outstanding cells in the central stage. Such methods can have extra communication and computation overhead but usually provide good delay characteristics similar to the original LB switch. Finally, we conclude the discussion about the schemes which use re-sequencing buffers at the output buffer stage.

2.3.1 Uniform frame spreading

The uniform frame spreading (UFS) is the first approach proposed by I. Keslassy [39] for the the multi-stage buffering LB switch [40]. The algorithm, doesn't require any additional cell reordering hardware and is not using any re-sequencing buffers in the output. The main idea of the UFS is to spread equally cells to all the VOQs destined to the same output. The input stage of the switch composed from the N FIFO queues, so the buffering structure is the same as in the central stage. Arriving cells destined to the appropriate output are kept in the input buffers until there are N of them in the queue (make up a *full frame*). During the next N time slots the cells are spread along the corresponding VOQs of the central stage. It is assumed that initially there are no cells in the central stage and the cells become head-of-line of the relevant VOQ. This policy is applied for each input, so finally each cell departs the switch in order.

In spite of the fact that this distributed algorithm shows good results for a heavy traffic load, it is evident that the throughput can be low in case of light load. If a frame is not *full* it is necessary to wait undefined time to complete a frame, this situation may cause *starvation*.

2.3.2 Padded frame algorithms

The **the Padded Frames (PF)** and **improved PF (PF+)** algorithms proposed in [34] are considered as a generalization of the previous scheme. In similar manner the multi-stage buffering scheme is used.

In the new algorithm the absence of *full frame* will not lead to starvation. The algorithm selects the largest nonempty queue with say $F < N$ cells and inserts $N - F$ empty cells to obtain a padded frame. Right after, frames are spread between the central stage buffers as in the UFS scheme. After arrival to the output fake cells are immediately dropped. Definitely, this may create instability in the system during the light loads when the amount of padding is very high.

Analysis of traversing packet delay and throughput shows that UFS, PF as well as *full ordered frames first* (FOFF) algorithms are strongly dependent on incoming traffic pattern and the load of the switch. While considering admissible traffic [17, 39] with matrixes close to self-similar [44, 59], PF scheme shows the best delay characteristics, while the UFS scheme [39] has starvation problems under light loads and while small bursts are received. The modified versions of the original LB switch to prevent the cells out-of-order issue were proposed in [15, 46, 57, 75]. Following sections will explain the operating principles of these designs.

2.3.3 The mailbox switch

The mailbox switch architecture is presented in Figure 2.7 and consists of two crossbar switch fabrics and buffers between them. The rule for predetermined interconnection of crossbars is the following $(i + j)\ mod\ N\ =\ (t + 1)\ mod\ N$. The patterns are periodic with the round-robin interconnection, specifically, inputs and outputs are connected to each other exactly once in every N time slots. The general assumptions for this design is similar to the one presented in Section 2.2.1. Buffers are called mailboxes, there are N mailboxes each of it contains N bins with F cells (here "cell" has meaning of a buffering unit) in it (each cell can store exactly one equal size packet). The switch uses a single FIFO queue for each input.

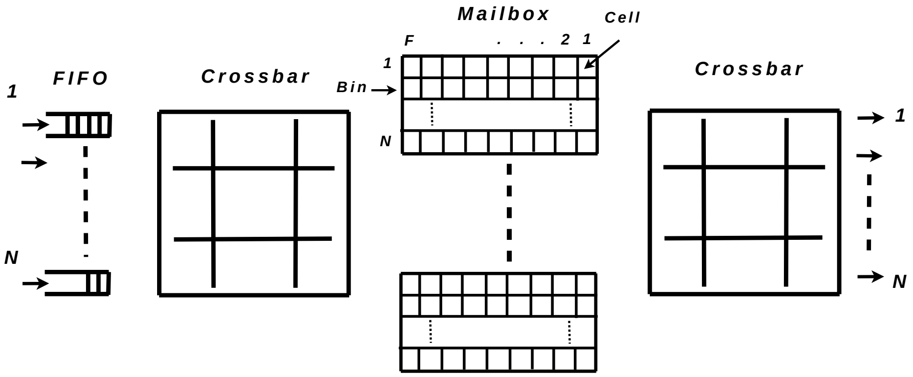


Figure 2.7: The Mailbox switching architecture [15]

The key idea of the mailbox switch is to use a set of symmetric connection patterns (Figure 2.8) to create a bidirectional link for transferring the information about the packet departure times using construction properties of the switch (each input and output is assumed to be implemented at the same line card). The information contains the value of virtual waiting time (VWT) $V_{i,j}(t)$ for each traffic flow that is the departure time of the last packet from input i to output j in the mailboxes. When a packet from i to j becomes the head-of-line packet of a FIFO queue at time t, the mailbox switch locates the packet in the mailbox such that it will depart not earlier than $t + V_{i,j}(t)$. Having this timing information, the switch can schedule packets so that they depart in the order of their arrivals.

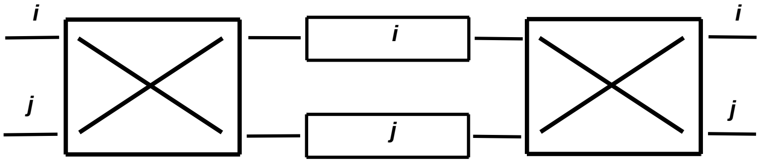


Figure 2.8: Example of symmetric interconnection pattern [15]

While this scheme requires searching the mailbox for an available location to be read after the VWTs, it's simpler than some other approaches. The drawback of this scheme is a HOL (head of the line) blocking. In the searching phase in order to find an empty cell for placing equal size packet, there might be several tries until an empty cell is found. An unsuccessful try may be viewed as a "collision" (since no free cell in the bin is found) and waiting time should be increased by N time slots (till the next try). This kind of input

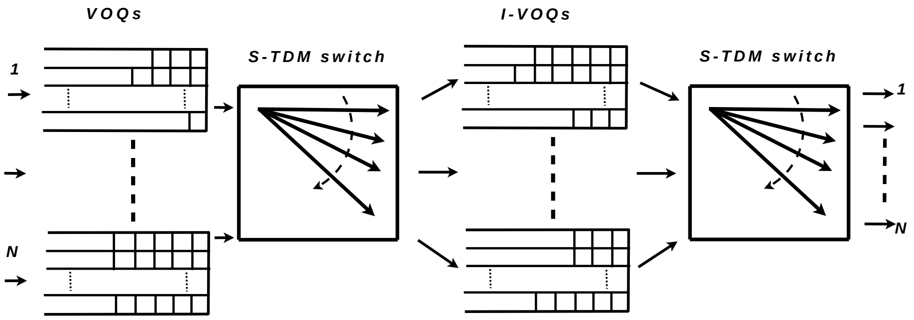


Figure 2.9: The Contention and Reservation switch [75]

packet blocking will result in low throughput and large delay. In order to solve these problems the authors implemented search of empty cells with limited number of forward and backward tries [15].

2.3.4 The Contention and Reservation switch

In [75], a novel **Contention and Reservation switch (CR switch)** design was proposed to address the cell reordering problem (Figure 2.9). It is considered to be an improvement of the Mailbox switch (Figure 2.7) proposed in [75]. The basic assumptions about the packet size, synchronization and flow rate are similar to the initial scheme presented in (Section 2.2.1). The CR switch like the Mailbox switch uses symmetric interconnection pattern to provide feedback bidirectional links from central buffers to the connected input buffers. The idea behind this, is to use different operational modes for different input stage loading of the switch. The algorithm presents the combination of the good features of the UFS scheme under heavy load and Mailbox switch in the light load without throughput reduction.

It is done by means of new VOQ with Insertion (I-VOQ) mechanism which allows to replace head-of-line packets(HOL). I-VOQ can distinguish three types of packets in different modes which allows to keep central stage buffers occupancy always similar. Input stage buffers are implementing VOQs and sort arriving packets according to the destination. Fake packets are always generated when the I-VOQ is empty, so there is always a packet in the queue. In the light load, when input VOQ occupancy is not high (during a N time slots period), input queue contention packets (number is less than N) are served to the I-VOQs. Contention packets arriving to the I-VOQs can replace fake packets and can be stored ONLY in HOL position. If it is not possible, contention packet is blocked in the input and retransmitted later. Reservation packets are generated when a sequence of N packets can be found in the input VOQs and always placed in the non-occupied tail position of the queue. In order to avoid packet loss authors of [75] assume that central stage buffers are infinite. The simulation results represented in [75] demonstrate CR switch delay superiority in comparison with the UFS and PF schemes.

2.3.5 The Concurrent Matching switch

The Concurrent Matching switch (CMS) uses two identical stages of fixed optical configuration meshes (Figure 2.10). The optical mesh presents Space Division Multiplexing (SDM) approach instead of Time Division Multiplexing (TDM) of crossbars. Input stage of the switch is implemented using VOQs structure. The intermediate inputs equipped with coordination slots and virtual counters (each line card maintains N^2 virtual counters, one counter for each flow). The main assumptions are similar to the ones shown in the Section 2.2.1.

Incoming cells are buffered in input VOQs of corresponding line card. Instead of instant cell load-balancing among the second stage, first each input sends the request tokens cyclically to the central stage, so the counters in the central stage are updated immediately about the traffic state in the input. Right after each intermediate input is solving the matching problem (while using only local counters information). Based on the matching result, the grant tokens are sent to each input back and appropriate counters are decremented. Only after this step, the corresponding VOQs in the input are allowed to send cells to the intermediate inputs, where they are stored temporarily in the coordination slots. Finally, during the next N time slots cells depart to the corresponding outputs of the switch.

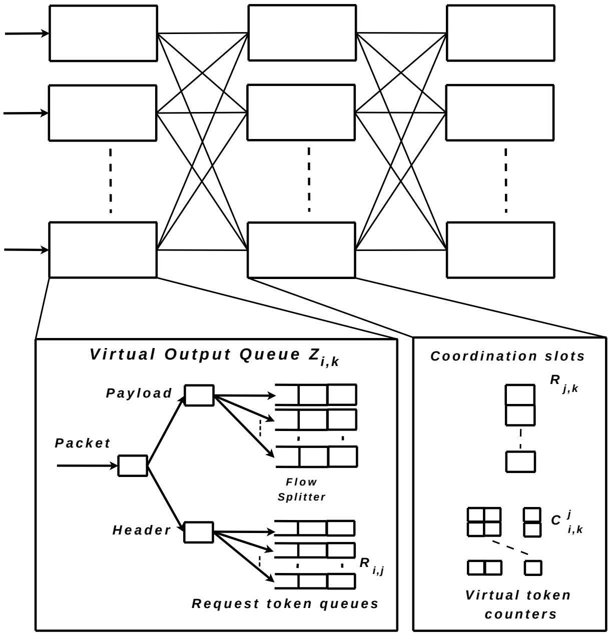


Figure 2.10: The Concurrent Matching switch [46]

2.3.6 Re-sequencing in the multi-stage buffering scheme

In this section, we analyze the architectures that are overcoming cells mis-sequencing problem at the output stage. First we discuss the basic approach with re-sequencing outputs, showing difficulties related to hardware implementation. Similar to the UFS algorithm, examined in previous section, we will present a modified scheme, called first ordered frames first (FOFF), which bounds the amount of cells mis-sequencing, but still require output reordering. Finally, the novel Byte-Focal switch that is using virtual input queuing (VIQ) structure for output reordering will be presented.

In the original multi-stage buffering scheme [18] cells from the same flow in the input are spread in the round-robin mannerto the corresponding VOQs (Figure 2.11) and are served with the first come first served (FCFS) policy (same as FIFO). The advantage of it is a traffic mixing along the input VOQs with respect to their departure times. A jitter control mechanism is used to delay cells to some maximum predefined value at the first stage so that flows entering the second stage are simply equally time shifted flows of the original ones. A set of theorems presented in [18], with consideration of two types of scheduling policies (first come first served and earliest deadline first) demonstrate system implementation complexity comparable with FOFF re-sequencing algorithm, which is described next.

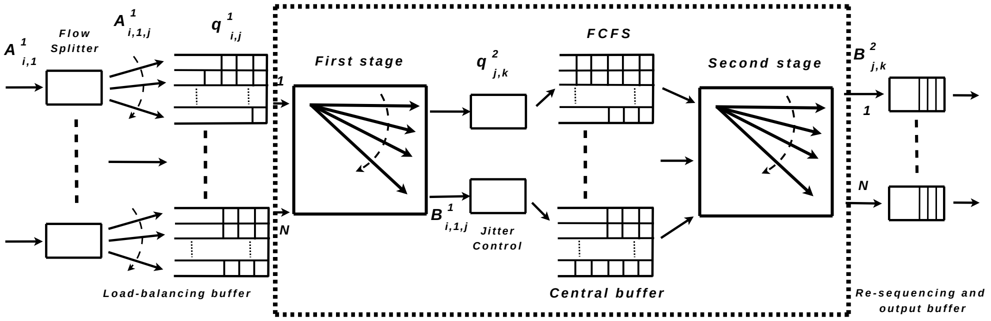


Figure 2.11: LB switch with multi-stage buffering [18]

The FOFF was described in [39, 40] and applied to the multi-stage buffering LB switch. The FOFF algorithm operates in the manner similar to the UFS but allows to sent a sequence of cells even if there are less than necessary to create a full frame. Every cycle of N time slots each input examines the availability of a full frame. Sending a full frame has a priority over the uncompleted ones. During the next N time slots cells of a frame are spread uniformly over the central stage [40]. If full frames are not available the algorithm checks other non-empty queues with number of cells $K < N$. During the next N time slots the transmission to the corresponding line cards will happen. The pointer will keep track of the last line card which received a cell, so that the next cell will be send only to the next VOQ. It is clear that the switch can have mis-sequencing only in the light

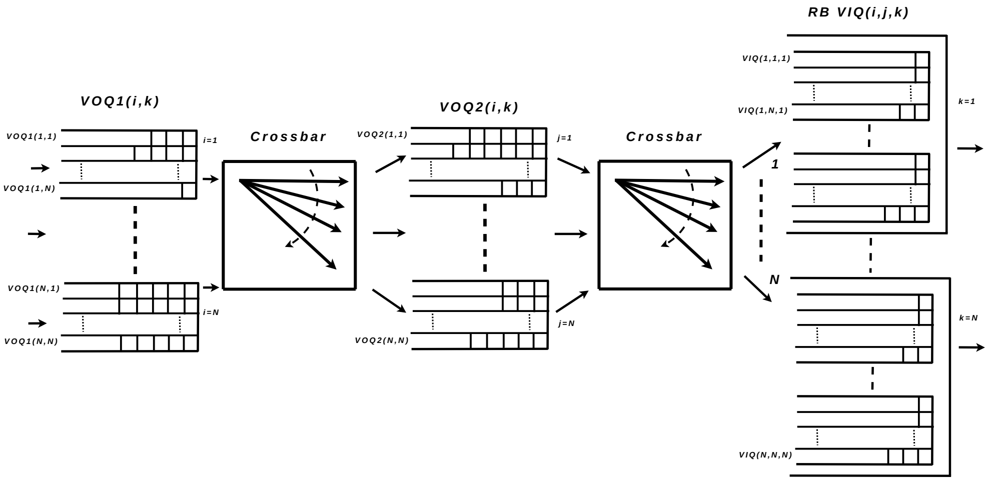


Figure 2.12: The Byte-Focal switch [57]

traffic load when the full frame cannot be created. But as it was shown in [39] the amount of mis-sequencing is always bounded. For that reason the output stage is equipped with output and re-sequencing buffers which remove remaining amount of mis-sequenced cells. Each output buffer uses N FIFO queues with size that is no less than $N^2 + 1$ cells.

2.3.7 Byte-Focal switch

The novel Byte-Focal Switch [57] (Figure 2.12) is equipped with two sets of VOQs in input and central stage ($VOQ1$ and $VOQ2$ correspondingly) and *VIQs*, which are used as re-sequencing buffers at the output. At each output there are N sets of VIQs, each set is related to some input port i. Within each VIQ set, there are N logical queues with each queue corresponding to a second stage j as well. Cells from input i designed to output k via second stage input j are finally stored in $VIQ(i, j, k)$, and its obvious that the cells in the same $VIQ(i, j, k)$ are in order. The advantage of using VIQs is that the complexity of finding and serving cells in sequence is $O(1)$.

Since each $VOQ1(i)$ is made of a set of $VOQ1(i, k)s$ the way in which one $VOQ1(i, k)$ should be chosen among the others have to be considered (only one queue can transmit to the central stage during N time slots). Indeed, three solutions were proposed, among them *the round-robin scheme* (A), *the longest queue first*, (B) and *the dynamic threshold scheme* (C) were shown in [57]. First solution gives fairness (A) for every queue but behave poorly with non-uniform traffic. Another two are able to handle problematic traffic but have extra computation overhead (B). The (C) scheme provides optimized delay performance results for a wide range of traffic patterns, packet lengths and packet length distributions.

As was shown, the original LB switch and recent more complex solutions are capable to efficiently solve the out-of-sequence problem introducing different level of complexity and overheads. As mentioned, the solutions with re-sequencing in the input stage are considered to have extra computational and communication overheads. On the other hand, the architectures solving out-of-sequence issue with output re-sequencing can also have high hardware complexity like in [18] or due to the fact that large amount of buffers is used [57]. In the following section we will analyze all possible overheads and algorithmic complexity of all mentioned above schemes.

2.4 Evaluation of the System's Scalability in Space and Time

The initial LB switch described in [17] has a simple control and considered to be a highly scalable solution. However, as a switching fabric it is proposed to use bufferless crossbar switches, which are known to be poorly scalable when the switch size is large(10^3 ports). Contrarily, LB switch has a predefined round-robin interconnection pattern and do not require all possible states interconnections from a crossbar. This fact gives a possibility to replace crossbars with less complex Clos [19], Benes [51] and Banyan network-based schemes in [15, 17, 64] together with Space Division Multiplexing switches (fixed mesh) in [35, 39, 42]. These ideas are introduced in Section 2.4.1.

In some newly proposed switches [15], [75], [46], [73] feedback links are used to propagate additional information about the state of the other stages of the switch. Take, for example, the Mailbox switch where information about cells departure times is transmitted from central stage to the input. As a rule, some extended controllers are used to process specific information. In fact, the computational and communication overhead can be a bottleneck for the large switches. In the section 2.4.2 we will derive some conclusions on the ability of such designs to scale up.

2.4.1 Scalability limitations regarding the basic scheme

The basic single-stage buffering LB switch architecture [17, 18, 40, 65] is implemented using two crossbar switches with a set of buffers in between. The switch does not have any feedback between the stages, as well as not doing any complex calculations during a time unit. The LB switch is making transmission of packets with a fixed size (through this section, we call them simply *packets*). Due to simple control the architecture don't have any communication and computation overhead even for large switch sizes($N = 10^3$ ports). As it is mentioned above, the schedulers used in the bufferless crossbars can be a point of bottleneck when N is large. In order to reduce the initial system complexity C.-S. Chang et al. [15] alternatively have proposed to replace crossbar switches with simplified three-stage Clos (like for instance [19]), Benes network structures [18, 51] or other less complicated structures [35] and 2x2 crossbars. Special attention was paid for implementation structure based on Banyan network architecture [64]. As mentioned, round-robin serving policy requires only N matrixes for each input to be realized. Practically, the composition of $(N \log_2 N)/2$ 2 x 2 switches is required to build $N \times N$ Banyan network.

The way to implement large N x N symmetric (the condition $(i + j)$ mod $N = (t + 1)$ mod N should hold true) TDM switches with number of ports $N = 2^k$ was presented together with the Mailbox switch in [15]. Each N x N TDM switch $N = pq$ consist of two stages. The first stage consists of p q x q symmetric TDM switches and the second stage consists of q p x p symmetric TDM switches. The stages are connected by the perfect shuffle, i.e. the $l - th$ output of the $k - th$ switch at the first stage is connected to the $k - th$ input of the $l - th$ switch at the second stage.

As an alternative to the TDM switches, authors of [30, 31] proposed to use a fixed optical mesh(sometimes with WDM). The crossbars are replaced with the two optical meshes where one mesh running twice as fast. It is assumed that the inputs, the output and the central buffers are implemented on the same line card. The packet traversing rate is equal to $2R/N$ [39]. Each input is simply multiplex (not only in time but also in space) incoming packets among the central buffers, in the similar manner like packets are demultiplexed at the output. For LB switch multiplexing and demultiplexing should be also uniform. The throughput characteristics of the LB switch while using different optical mesh interconnections are presented in [31].

2.4.2 Scalability of recent architectures, computation and communication overhead

The Mailbox switch

The Mailbox switch was depicted in Figure 2.7. The main goal of the Mailbox switch is to service packets in order through the switch. In the following architecture the scalability can be limited by the following factors: (1) crossbars switches and (2) computational and communication overhead. As mentioned, the Mailbox switch uses symmetric interconnection pattern in order to create a bidirectional communication link between an input and output stage, since they usually implemented in the same line card. To service packets in order, the packet departure time is fed back to the input stage, so the next packet will have all the required information about the transmission time. Authors in [15] propose several strategies, which in fact have different computation and communication overhears. In the following we examine in the detail possible overhead which can have a switch with large number of ports (10^3-10^4) and different packet size.

Generic architecture. Generic Mailbox switch, while using symmetric interconnection, feed back a VWT to the packet in the input stage. The virtual waiting time $V_{ij}(t)$ defines period that packet from flow (i, j) has to wait in a specific mailbox and even in specific bin position (once it was transmitted from the input stage), in order to be delivered to the output stage in sequence. As soon as transmission of the packet is done to the central stage bins it is possible to calculate in advance packet service and departure times (since all the crossbar interconnections are deterministic and periodic). The operating phase for the Generic architecture is the following:

1 step *Retrieve mails:* According to current interconnection between output j and mailbox $h(j,t)$, the transmission of the packet from corresponding bin j (of mailbox) is happening. The waiting time for the packets placed in some other mailboxes can be easily determined as soon as the placement is done.

2 step *Send mails:* The transmission of the packet from input i to mailbox $h(i,t)$ and bin j (according to crossbar interconnection) occurs. The packet is not allowed to depart before $t + V_{ij}(t)$.

3 step *Update VWTs:* The flows that do not send mail during time slot t update their VWT for time slot $t+1$.

Depending on the arriving traffic if the arriving packet can find an empty cell in the mailbox (say at the position f, which is less than F-length of the buffer), than transmission is successful, otherwise packet will be blocked at the HOL position of the input stage. More precisely, for each placement of the packet inside the central stage bin, the interconnected input will receive $\log_2(FN)$ bits of information. Moreover, at each input port i, the information about $V_{ij}(t)$ for all outputs is kept.

Lets assume that the switch size is 10^3 ports and the equal size packets are $L = 128$ bits(16 bytes). As it is usually assumed that central stage buffer length should be at least slightly larger than the switch size, we will obtain that $\epsilon = \log_2(NF)/L = \log_2(10^3 10^3)/100 = 20/128 = 0.156$ or 15.6% of the overhead. Please note that the amount of the overhead will increase in case if buffers are larger than the value considered. The same one can apply to larger packet size (say 1500 B). Communication overhead can influence on the performance characteristics of the Mailbox switch also if feedback links are experiencing unexpected delays in information transmission. On the other hand, during this scenario Generic Mailbox switch doesn't have any large computation overhead, since it needs only increment/update waiting time counters (up to N for each input) for the next packet transmission.

The Mailbox switch with cell indexes. In spite of the Generic approach which is not defining the value of the position f where the cell will be placed inside the bin, in approach with cell indexes switch tries to *minimize* value $f_{ij}(t)$ such that the packet will not depart earlier than $t + V_{ij}(t)$. In general $f_{ij}(t)$ is called cell index of VWT $V_{ij}(t)$. In addition to $f_{ij}(t)$ a new counter $g_{ij}(t)$ for flow (i,j) is introduced. In this case all the information of $f_{ij}(t)$ and $g_{ij}(t)$ is kept in the input for all possible outputs(from $j = 1, \dots, N$). Accordingly the operating phases are modified as follows:

1 step *Retrieve mails:* Similar to the previous case.

2 step *Send mails:* Packet from input i is transmitted to specified mailbox together with $f_{ij}(t)$. Then the packet is placed to bin j and to some cell in the range $max(f_{ij}(t), 1)$. If placement was successful than the index is assigned (say f), otherwise $f = 0$ and error message transmitted to i output;

3 step *Update VWTs:* If the transmission was successful, than $f_{ij}(t+1)$ can be easily defined by f at time slot t and $g_{ij}(t)$ is reset to N. Otherwise if transmission is not successful than again $f_{ij}(t+1) = f_{ij}(t)$ and $g_{ij}(t+1) = g_{ij}(t) - 1$.

Lets make small investigations on the overheads of presented model. Presented approach has smaller amount of communication overhead due to transmission of only cell indexes. Taking into account previous example ($N = 10^3$ ports, $F = 10^3$ cells and packet $L = 128$ bits) we obtain $\epsilon = \log_2(F)/L = \log_2(10^3)/128 = 10/128 = 0.0781$ or twice less than in previous example. On the other hand each input now have to keep track and update (every time slot) both $f_{ij}(t)$ and $g_{ij}(t)$ counters (in general up to $2N$ values during a time slot). Since that the computation overhead of the system will be doubled.

The Mailbox switch with a limited number of forward tries. In the previous approach it can happen that in order to find an empty cell for a packet from a flow several tries might be done. For each unsuccessful try the VWT will increase by N time slots (like collision). Accordingly if the collisions are happening quite often, packets will not occupy available capacity of mailboxes. In order to avoid that authors in [15] put a limit on additional waiting time δ in case of collision such that $f_{ij}(t) \in \min(f_{ij}(t) + \delta, F)$. Please note that δ is a maximum increment of the index of cell waiting time. For the presented case the overheads are similar like in previous approach. The implementation improves systems' throughput and delay.

Mailbox switch with a limited number of forward and backward tries. The problem of finding empty cell can be solved by mean of backward tries(it is bounded by $N\delta_b$ time slots), e.g. if free cell position is looked not only in the forward but also in the backward direction. The new constraint for cell indexes $f_{ij}(t)$ is in the interval from $\max(f_{ij}(t) - \delta_b, 1)$ to $\min(f_{ij}(t) + \delta, F)$.

It is easy to see that the amount of communication and computation overheads do not change, since only the interval of $f_{ij}(t)$ was changed. Based on [15], the last two approaches make improvement in throughput and delay characteristics of the Mailbox switch. Due to the HOL blocking, throughput of 100% cannot be reached and remains low (around 58%) unless last two approaches are used. Increase of δ will show growth in throughput but the delay parameter will also increase more intensively. The total amount of buffers used in the Mailbox switch is $PN + FN^2$ cells, where P is a length of an input VOQ buffer.

The Contention and Reservation switch

Next we examine **the Contention and Reservation switch** introduced in [75] that merge the advantages both of the Mailbox switch and the Uniform Frame Spreading. The CR switch is operating in two modes: the contention mode and the reservation mode.

The operating switch needs to send one bit of data to inform whether the transmission of contention packet was successful or not. The feedback information is sent from the central stage buffers to the inputs. The communication overhead of the system during a time slot is just one bit of information and not depend on the switch size. Taking into account previous example ($N = 10^3$ ports, $F = 10^3$ and packet size is $L = 128$ bits) we obtain $\epsilon = 1/L = 1/128 = 0.00781$ or 0.781% of overhead. Communication overhead can be significant if transmission delays between the stages are large.

Due to existence of the VOQs in each input, each cycle of N time slots CR switch have to search the queue in order to decide if the packets are reservation or contention. So the queue should keep the pointer to the queue which will served in the nearest future. After the specific queue was chosen and served successfully, input have to update its pointers for the next cycle transmission (Success:Persist/Failure:Advance(SPFA) scheme). However several other possibilities to update pointers are possible [75]. Since the other algorithms cannot provide best delay characteristics, we don't consider them in our analysis. This is in fact the only computation overhead which switch can experience during the operation. As mentioned, there are different strategies to serve the queue, however the way to choose the input VOQ will change packets traversing delay and performance, but not the amount of the computation overhead during a time slot (for $N = 10^3$, each inputs updates one counter, which have to choose one VOQ among the N ones). The total amount of buffers used in CR switch is $FN^2 + GN^2$ packets, where F is a length of an input VOQ buffer and G is a length of a central stage VOQ (considered to be infinite).

The Concurrent Matching switch

The Concurrent Matching switch(CMS) similarly to the previous switches makes packets reordering in the input and central stage. In fact, authors of [46] propose to use the fixed mesh instead of crossbars which can be scaled to a very high port counts due to WDM. Note, if size of the switch is large(e.g. 10^3-10^4 ports), the amount of optical links used (without WDM) could be also significant ($N^2 = 10^6 - 10^8$), and creates a scalability limitations.

The switch operates with five phases, and each takes N time slots: packets arriving phase(assume arrival is finished at n^{th} time slot). After arrival packets are kept in input VOQs. At the end of $n + N - 1$ time slot corresponding tokens are already sent to the central stage (packets are sent after matching decision). In the matching phase (that takes also N time slots) the virtual token counters get incremented (each line card has N^2 counters placed) and matching decision is made on the base of global information. The CM switch guarantees 100% throughput if any stable matching algorithm is used and traffic is admissible. Authors in [46] have introduced CMS operational stability for SERENA [26], Maximum Weight Matching (MWM) [50] and iSLIP [49] scheduling algorithms. Based on the decision the grant token are sent back to the inputs (by the end of time slot $n + 3N - 1$), where corresponding input VOQs are chosen for transmission. Please note that according to the matching decision and arriving traffic not only one VOQ from each input can be chosen(like it is done in UFS [39], FOFF [40], etc.), but several ones. In response, the packets are first sent to the central stage coordination slots(by the end of time slot $n + 4N - 1$) and later depart to the output(by the end of time slot $n+5N-1$). The transmission delay of the packets from the input to the output is taking $(n + 5N - 1) - (n + 3N) = 2N - 1$ time slots, and independent of time slot n.

The CM switch, similarly to basic LB switch [40] can be implemented in a set of linecards. The linecards are interconnected while using four fixed rate channels. Two channels are substituting crossbar switches and used for data transmission from input to the output. Two remaining channels are used for transmission of request and grant tokens together with information about the matching decision. Each request and grant token can be presented in $log_2 N$ bits. Lets evaluate the communication overhead which input have during a cycle of N time slots together with amortized overhead during a single time slot. During two cycles of $2N$ time slots switch should send N request and N grant tokens from input to the central stage and back. Taking into account previous example when $N = 10^3$, $F \geq 10^3$ and $L = 128$ bits, we obtain during a cycle overhead of $2N \log_2 N$ bits or amortized during each time slot $(2N \log_2 N)/2N = \log_2 N$ bits. Than the ratio of the overhead to the packet size is $\epsilon = (\log_2 N)/L = 10/128 = 0.0781$ or 7.81% similar to Generic Mailbox switch - section 2.4.2.

Computation overhead is fully depends on the performance of chosen matching algorithm. It's clear that N^2 counters should be checked for the correct match during a period of N time slots. That means that complexity of each algorithm can be amortized by the factor of N. Some of the matching algorithms may have insufficient time in practice to obtain a matching decision during these period of time. Lets present most popular matching algorithms and evaluate computation overhead for each. MWM [50] is strongly stable and can achieve 100% throughput. Unfortunately, in order to converge in worst case it takes around $O(N^3)$ iterations, which can be insufficient for large switch sizes. Another stable matching algorithms like Parallel Iterative Matching (PIM) [62] and iSLIP [49] have to perform from $log_2 N$ to N iterations during a time slot, which might not be scalable for large N as well (in particular, when $N = 10^3$ ports). Authors in [46] proposed to use SERENA [26] matching algorithm which has complexity of $O(N)$(amortized $O(1)$). This algorithm is showing both stability and good performance characteristics. The total number of buffers used in CM switch is $FN^2 + N$ packets, where F is a length of an input VOQ.

In summary we conclude, that the computation overhead of the CM switch is fully depend on the matching algorithm chosen. In particular it might be a negative issue for a very large switch size (e.g $N = 10^3 - 10^4$ ports).

The Frame-Aggregated CMS

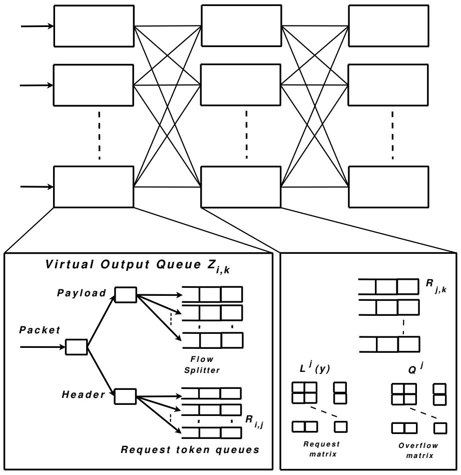


Figure 2.13: The Frame-Aggregated CM switching architecture [47]

Frame-aggregated Concurrent Matching Switch [47] (FA-CMS, presented in Figure 2.13) is very similar to previously presented CMS architecture, with some minor differences in the implementation and operating ideas. The main goal of this architecture is to achieve $O(N \log N)$ delay, since previously the average delay was bounded to $O(N^2)$. The new architecture goes through the same phases like CMS (packet arrival and transmission of request tokens; transmission of grant tokens; transmission of packets through both meshes), however operations of FA-CMS are presented in terms of superframes. Superframe φ is composed of N consecutive frames or NT consecutive time slots, where T is a minimum frame size [52]. In the central stage of new FA-CMS two counters are implemented: request and overflow ones.

Although new FA-CMS architecture [47] is operating in superframes and sends batches of tokens during superframes, the communication complexity is the same like in CMS. It is clear that the new switch is operating in different time scale, but the amount of data sent during a single time slot remains similar like in CMS switch. In particular, each input of the CMS during request phase sends N tokens during N time slots,on the other hand FA-CMS during the same phase sends maximum NT tokens during a superframe or NT time slots. Since that the communication overhead of FA-CMS during a time slot will be transmission of $log_2 N$ bits. Using previous example when $N = 10^3$, $F \geq 10^3$ and

$L = 128$ bits, $\epsilon = (\log_2 N)/L = 10/128 = 0.0781$ or 7.81% for our example.

If computation overhead of CMS was based on the complexity of matching algorithm used, than in FA-CMS there is no need to decompose request matrix by means of matching algorithms [62] [49] or edge coloring [22]. The only operations which should be done is to update request and overflow matrices and generate grant tokens according to Step 2. Since each central stage needs to generate at most one token, the processing takes constant time. The total amount of buffers used in FA-CMS is $GN^2 + TN^2$, where G is the length of input VOQs.

The Byte-Focal switch

The Byte-Focal switch [57] has some similarities with the initial LB architecture. It has a set of VOQs at input and central stages and the set of VIQ re-sequencers in the outputs. Two crossbars use deterministic and periodic patterns and synchronously change configuration. The scalability limitations of crossbars and ways of replacement with Banyan were described previously.

The Byte-Focal switch is not implementing cyclic interconnection pattern and doesn't have any feedback between stages, so there is no communication overhead at all. Since input stage buffers are implemented as a set of VOQs, the computation overhead can appear while searching the appropriate VOQ for transmission. The authors of [57] have presented a several schemes for service of packets in VOQs. The simplest round-robin scheme needs just to keep pointer on the VOQ which was just served and update it for the next transmission. The other schemes, such as Fixed threshold and Dynamic threshold schemes, perform search of the queue with fixed or dynamically changing threshold during a period of N time slots. If several queues with required threshold were found the algorithm picks one in round-robin manner. In particular for $N = 10^3 - 10^4$ ports N states should be checked (in each input during N time slots) before transmission and appropriate decision is made. As it is shown in [57] the presented schemes change traversing delays of packets under different traffic matrices.

Buffering units at the output stages make algorithm for packet re-sequencing simple since packets from specific input, central stage VOQ and destination are kept in separate VIQs at the output. In general output-and-resequencing buffer use N^3 VIQs and N departure queues (DQs) for packet re-sequencing. The total amount of three-dimensional buffering used in the system can be a limitation for a large switch sizes. In particular for $N = 10^3 - 10^4$ ports $2N^2G + N^3H + NK$ buffers will be used, where G is a length of VOQ1 and VOQ2, H is a size of VIQ and K is a size of DQ.

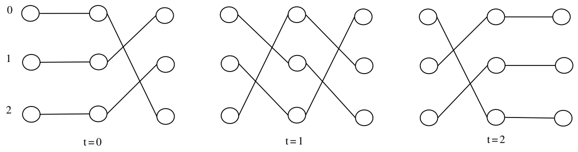


Figure 2.14: Staggered symmetry interconnection pattern for $N = 3$ [73]

The Feedback-based two-stage switch

The Feedback-based two-stage switch [73] is implemented using set of VOQ buffers in the input and central stage and two crossbar switches, similarly to basic multi-stage buffering LB switch [18]. Two crossbars are configured using different deterministic and periodic sequences. In [73] several interconnection patterns were examined, and the major focus was done on the pattern which provides staggered symmetry and in-order packet delivery. In particular, first crossbar interconnecting input and output uses classical round-robin scheme $j = (i + t) mod N$. The second crossbar sequence is constructed according to principles of staggered symmetry e.g. $k = (j + N - 1 - t) mod N$, so interconnection rotation in crossbars is happening in opposite order(Figure 2.14). If during a time slot t central stage buffers j are interconnected to output k than, according to staggered symmetry in the $(t + 1)^{th}$ time slot input k will be connected to middle-stage j. According to this property the current status of VOQs(j,k) (for $k = 0, \ldots, N - 1$) can be added to the header of packet from central stage to the output (like it is implemented in IP headers [11]). As each input, central stage and output usually are implemented in the same line card, than the status of the VOQ in time slot t is feed backed to the input so during the time slot $t+1$ input knows occupancy of the central stage queues. Another important property of the feedback-based two-stage switch is that each input i during a cycle of N time slots is always connected to the same output k (anchor output). Authors in [73] have proposed to use a single-packet-buffer in the central stage in order to provide the same delay for every packet in any central stage port. Similar delay will guarantee in-order packet delivery for the packets within the same flow. The dedicated feedback information can be represented by N bits and must be sent each time slot. For $N = 10^3$ ports and with average data packet size of $L = 100$ bits, the overhead will be 10 times larger than the packet size,e.g. $\epsilon = N/L = 10^3/10^2 = 10$. If assume that central stage buffers are implementing the queues with length F than at least $\log_2 F$ bits of information should be dedicated for status representation of each middle-stage VOQ. Please note that during a time slot the feedback information is transmitted also from the input to output stage. Since the switch has N VOQs at each input, different service policies for choosing each VOQ can be defined. However this policies do not introduce any significant computation overheads for the switch. In particular Round-robing(RR) serving policy just needs to keep counter on the previously selected queue, and chose the next non-empty queue to serve during the next cycle. Longest queue first (LQF) performs a search of the longest queue, and served it during the next cycle. Earliest Departure First(EDF) selects the queue which has earliest departure time(minimum delay) at the middle-stage port. The total amount of buffering used in the switch is $GN^2 + N^2$ packets, where G is a length of an input VOQ.

The three-stage switch in [74] implementing the same principles previously presented in [74] with the difference that additional switch fabric is connected to the outputs. The switch does not make any impact on the previously presented overheads, throughput and packet order. The main focus of this architecture is to cut down the traversing delay of the switch.

Chapter 3

Scientific Questions

Due to the large variety of online applications, real Internet traffic is composed of packets of a variable length. Accordingly to the last observations, the Internet traffic has trimodal structure [58], with the main packet sizes of 40, 1300 and 1500 B. To accommodate a rate of optical fiber links, input buffer switching architectures are commonly deployed. In order to increase a buffer utilization inside the switch, segmentation of variable size packets is used. It is assumed that variable size packets are segmented into small data cells (like ATM cells) at the ingress stage and reassembled at the egress stage before being transmitted. Moreover, bufferless crossbars, which are widely applied as a switching fabric, are considered to efficiently operate only with fixed-size cells. Noteworthy, a crossbar scheduling is quite complex, when number of ports is large. A newly proposed crossbars with an internal buffering reduce scheduling complexity, but may cause problems with fairness [20, 37, 67]. However, these types of crossbars are not considered in this thesis.

The review on the state of the art in the field of load-balancing switching given in previous chapter shows good potential of this architectures mostly due to it's scalability. On the other hand, the solutions given in chapter 2 most commonly deal with packets of the same size (cells) and infinite buffers. This dissertation presents analysis of the load-balanced switch with novel set of assumptions and examines new possible problems which appear due to considered assumptions.

We start with the analysis of the packet loss probabilities for the LB switch with finite central stage buffers and fixed size packets arriving to the inputs. Next, we focus our attention to the packet loss problem when variable size packets are contending inputs. Finally, we will present algorithms with can efficiently avoid/minimize the internal packet loss. The analysis of mentioned issues makes a groundwork for this thesis and is given in more detail in the next chapters.

This section is organized as follows. Section 3.1 and 3.1.1 shows the considered architecture, discusses the important issues of loss, performance and traffic distributions inside the system. The second part of section 3.1 is focused on the ways to resolve the mentioned above issues. Finally, the last section of the chapter will describe future research directions which were accomplished in this thesis.

3.1 The Considered Open Issues

The switching architecture. The considered LB switch is presented in Figure 3.1. Since all the buffers inside the switch are finite, a cell loss can appear in the system due to the overflow. Let's assume that each input is equipped with FIFO buffers, where arriving variable size packets are kept. Before transmission of packets through the system happens, packets are segmented into a small data cells. The segmentation overhead for each cell includes the information about a sequence number of the packet it belongs to and self-identification number inside this packet. The segmentation process allows the bufferless crossbars to be used for load-balancing (crossbar 1) and switching (crossbar 2) of cells to the next switching stage. Switching fabric of crossbars has predefined interconnection sequence and is presented by the formula (2.1). All elements of the system are synchronized (inputs, outputs, crossbars) in order to perform simultaneous cells transmission. Similarly to initial LB switch [17, 40] we assume that no feedback links are implemented between the stages.

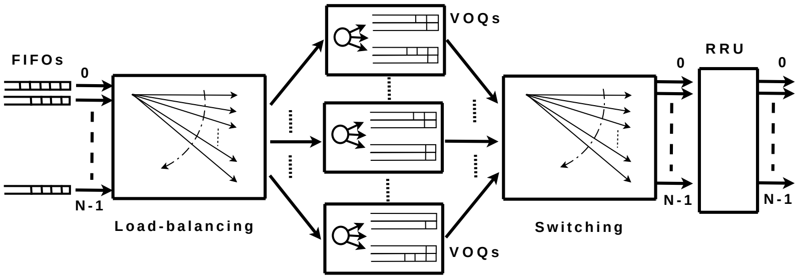


Figure 3.1: The considered LB switching architecture

3.1.1 Cell and packet loss

As mentioned above, cells inside the switch can be dropped when the intermediate buffer is full. Since each cell is essential for a packet reassembly, a single cell drop will result in the whole packet removal. Therefore, it is expected (and it will be proven below) that packet loss is always larger than the corresponding cell loss.

Several types of the packet/cell loss can be introduced in the LB switch. Input buffer packet loss may occur, when packets arrival rate is higher than the service rate inside the switch. If input buffer is full, it is preferable to drop the entire packet before segmentation and rely on the network protocol(like TCP [10]), which can request retransmission of that packet later on. Such a countermeasure helps to avoid buffering capacity wastage inside the system.

The second type of the packet loss takes place at central stage buffers. Since the LB switch - Figure 3.1 - implements no feedback, arriving cells are spread to the central stage without respect whether transmission will be successful or not. On the other hand, the switch is operating in a way that cells belonging to the same packet are distributed in a round-robin manner between the central stage buffers. Since these buffers are independent and do not communicate, the task of removing of the remaining cells (belonging to the same packet) from the central stage buffers can be insoluble. In the next chapters, we attempt to analyze non-intentionally caused packet loss at the central stage, while collecting the statistic at the output reassembly stage. For packet loss monitoring by means of simulations, we introduce new fields inside a cell header. In particular, *drop field* is a boolean variable which shows whether a cell is dropped or not. *Stage dropped field* is usually updated in correspondence to the stage where the cell was dropped.

The packet loss at the central stage can be subdivided into other types. *The overall packet loss* is evaluated as a ratio with a number of packets dropped inside the system to the number of transmitted packets. To better understand internal features of the system, we also investigate the *packet loss at the central stage VOQs*. The mathematical model for evaluation of packet loss inside the VOQ was presented in [2, 3] and in chapters 4 and 5. As it will be mentioned in the model, the probability of packet loss is strictly depends on a *chosen traffic path* (input, VOQ and output) and a *crossbars interconnection policy*. The details of this phenomenon were described in [2]. Both types of the central stage packet loss are analyzed using simulation model and are presented in chapter 5. Lets estimate the dependency between the packet and cell loss probabilities given above.

Packet loss boundaries. Let p be the parameter of the geometric distributed packet length, and $L = \frac{1}{p}$ is the average packet size. K denotes the total number of packets sent. If l_c presents the number of lost cells and l_p the number of lost packets then p_c the cell loss probability is defined as

$$p_c = \frac{\text{number of lost cells}}{\text{total number of cells}} = \frac{l_c}{LK} = \frac{l_c}{\frac{1}{p}K} = \frac{pl_c}{K} \tag{3.1}$$

and p_p the packet loss probability is defined as

$$p_p = \frac{\text{number of lost packets}}{\text{total number of packets}} = \frac{l_p}{K} \tag{3.2}$$

The boundaries on the fraction of the number of lost cells and the number of lost packets are

$$1 \leq \frac{l_c}{l_p} \leq L = \frac{1}{p} \tag{3.3}$$

which tells that at least one and at most L cells in a packet is lost (in average). The second inequality (3.3) results in (then dividing both sides by K)

$$\frac{pl_c}{K} \leq \frac{l_p}{K}$$

which, using (3.1) and (3.2), gives the lower bound of the packet loss

$$\begin{aligned} p_c = \frac{pl_c}{K} &\leq \frac{l_p}{K} = p_p \\ p_c &\leq p_p. \end{aligned} \tag{3.4}$$

3.1.2 Ways to avoid packet loss

As mentioned, it is always preferable to avoid the VOQ packet loss at the input stage and remove cells of a ”broken” packet (the packet with at least one dropped cell) in order to save the buffering capacity of the switch. This is possible only by means of centralized control which is performed over all central stage buffers or by means of some sophisticated interaction between all stages. For the LB switch, the packet loss of the specific VOQ (on the central stage) depends on the 1) *incoming traffic* at each input during a time slot and 2) *the occupancy of the other VOQs* (each VOQ has different number of cells in it) corresponding to the specific output. Knowing and controlling these two parameters it is possible to avoid packet loss in the central stage VOQs. Lets refer to some simple examples.

The LB switch with feedback (where feedback is implemented between the input and central stage) will request the list of occupancy states from the corresponding VOQs before cell transmission. If some of VOQs are congested there will be a possibility to block a cell in the input stage (create a back pressure) till the buffer is full or to send this cell to the available non-congested buffer. No cell will be dropped in this case.

The drawback of given example lies in the degradation of system’s performance (since Head-of-Line blocking (HOL) appears). On the other hand, such an scheme gives a possibility to predict cell drop while utilizing centralized control (which collects the information about input traffic and states of all the VOQs) every time slot.

It is obvious that the ways to avoid packet loss directly at the central stage buffer are not trivial and even in case of success these solutions influence the other performance characteristics of the LB switch (mis-sequencing, delay, and throughput). Therefore, it is always preferable to introduce such an algorithm with predicts central stage buffers congestion before the entire packet start its transmission from an input. In chapter 6 two possible solutions for packet loss minimization and avoidance are proposed.

3.1.3 Re-sequencing and reassembly

A packet loss that appears at the central stage influences the operating stability of the reassembly unit. It strongly depends on the packet delay on the central stage and on the chosen traffic pattern. On the other hand, packets which cannot be re-sequenced (reassembled) are always present in the system, caused by cell loss. Due to a distributed structure of the switch, the output re-sequencer will never know, whether the cells waiting for re-sequencing will be successfully reassembled or not (remaining cell of the packet may never arrive!). Since the buffer capacity at each output is limited, each output RRU have to keep track also on the re-sequencing (reassembly) process of other flows (up to N flows from all inputs). Therefore, the re-sequencing and reassembly unit can become another point of congestion and system's instability.

J. Turner in [66] has reviewed several schemes that are the most commonly used in the cells re-sequencing. In the LB switch the reassembly unit can execute functions of the re-sequencing unit (at the output stage). Alternatively these units can be implemented separately, if re-sequencing is done at input and central stages. In the latter instance the reassembly unit needs only to rebuild the packet from the sequence and send it out from the system. There are two well known approaches to implement RRU. **The first approach** controls the sequence numbers injected into the header (at the input) of the arriving cell according to the destination. Thus, the re-sequencer at the output waits for the arrival of all sequence and only than allows a packet reassembly and its departure from the switch. If a cell is delayed, the RRU will keep the remaining cells for undefined amount of time creating congestion at the output, and wasting buffering capacity.

The second approach implies addition of a time stamp field to each cell, before sending cells further. These types of re-sequences usually hold cells in the re-sequencing buffer until the difference between the current time and the time stamp exceeds some predefined threshold. That is why this kind of re-sequencers is rarely the subject of congestion. However, if a cell from a sequence is delayed inside the switch, the output re-sequencer can drop the whole sequence before the missing cell arrival.

In the proposed switch we suppose that re-sequencing unit per output is constructed from a set of N VOQs of size B_R. Such a scheme allows monitoring of the re-sequencing and reassembly process of up to N packet flows from all N inputs. Each VOQ will store the packets from specific inputs. In our study of sections 5.5.3 and 5.5.4 we investigate the dependency of the buffering amount in the central stage B on a packet re-sequencing delay. Latter instance of delay gives the minimum amount of buffers B_R necessary for re-sequencing unit. Please note, that for our analysis we consider only the packets which are successfully passed through the switch. In the switch without the feedback, it is expected that incomplete packets can also ask for reassembly operation. In order to prevent buffer wastage the threshold time (similar to time-to-live value of an IP header) for a cell/packet presence in the reassembly unit should be set (as presented in second approach above).

3.2 Future Research Directions

The following thesis is focused on the LB switching architectures which due to a distributed and simple control are able to support switches of extremely large sizes. Throughout the previous chapters we have presented not only the state of the art issues appraised by other researchers but also proposed several research directions for the future analysis. In particular we highlighted such issues like the internal mis-sequencing, the scalability issues, the problem of cell and packet loss, and set of problems related to the packets re-sequencing and reassembly. Using the fact that the real implementation of the LB switch will always rely on a finite amount of buffering, we presumed that a non-zero packet loss will appear at the central stage VOQs due to buffers congestion. Therefore the characterization of the internal loss both for fixed and variable size packet is considered to be the main point of this dissertation. Apart from this, the amount of packet loss has a strong dependency on the packet traversing delays at the central and output stages which gives a second interesting characteristic to investigate. In order to argue consistency of our estimates in the next chapters we will try to give detailed explanations and perform an extensive analysis to all the problems introduced.

Chapter 4

LBS with Equal Size Packets

The latest analysis performed on the intensive and evergrowing user demand on the Internet resources identifies a set of critical issues with which Internet can face in the recent years. Although the bandwidth dependent applications (like video streaming and interactive video gaming, peer-to-peer file transfer and file sharing) are already on the market and explore Internet services extensively, it is quite possible that today's Internet infrastructure is not ready for appearance of novel "killer applications". That is why the use of novel networking trends and packet switching technologies is essential.

Most of the packet switching technologies are forwarding packets from the ingress to the egress port while using substantial computation resources for decision making, header processing and packet storage. In this chapter we focus our attention to the LB switching architecture, which promises simple distributed control with almost no communication and computation overheads and large set of performance benefits [17, 40]. In particular, the architecture guarantees high throughput and small packet delay under certain assumptions.

In this chapter we analyze the cell loss probability in the central stage buffers while considering Single-stage switch [17] with finite amount of buffering. The chapter organization is as follows. Sections 4.1 and 4.2 present a short overview of operational principles related to LB switch as well as assumptions used for theoretical model. Section 4.3 analyzes the loss probability of the 2 x 2 LB switch with admissible input traffic, then in Section 4.4, we proceed with the general analysis of N x N switch. Finally, in Section 4.5 some numerical analytical results are presented.

4.1 The Considered LBS Architecture

The basic LB switch architecture shown in Figure 4.1 presents a stage where buffers positioned in-between two identical crossbar switches [16, 17, 40]. Each buffer at each intermediate input (central stage) is partitioned into N separate VOQs, one for each output in order to avoid Head-of-Line blocking and reduce throughput losses.

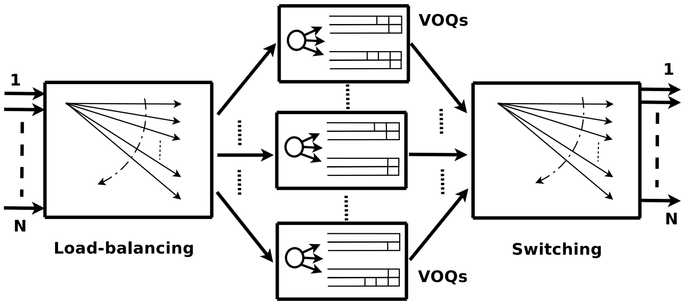


Figure 4.1: Single-stage buffering LB switch

The operating idea behind the N x N size single-stage buffering architecture [17] is to load-balance the packets (cells) from inputs along the VOQs of central buffering stage. Then, the packets are sent to the related output. Each crossbar set up a periodic connection pattern connecting the input to the central stage. The connection time for each packet is predefined and the rate is equal to $1/N$. Both crossbars operate identically and walk through a fixed sequence of interconnections according to the rule $j = (i + t)modN$. Arriving packets are switched instantly and there are no buffers inside the crossbars [17, 18]. Some packets can checkout from the system in a disorganized fashion because of the FIFO policy. Researchers have attempted to solve this problem by proposing a multi-stage buffering design [18]. In contrast with the single stage scheme a set of buffers are also available in the input and output stages. Consequently, solutions to resolve the cells that are out-of-order were presented in [39, 40]. Re-sequencing buffers within the output stage has promoted the existence of flow-splitters in the input stage and, jitter-control mechanism in the central stage.

4.2 Assumption and Traffic Model

Authors of [16–18, 39, 40] have assumed that all of the LB switch packets are of the same size and they simply call them packets. Although in the real Internet world packets have variable length, yet these assumptions have been idealized and several studies were built upon. Within this chapter, we will be calling *packets of the same size as cells* and, we assume them to be integer multipliers of the variable size packets. Another significant assumption is that each linecard uses the same common slotted time. Indeed, this assumption implies that only one cell can arrive at an input and depart from an output of the switch during a time slot. Each linecard is assumed to support equal rate flows. As one of the major assumptions to permit achievement of high throughput [39,65] for the initial single-stage LB switch is traffic admissibility. Precise definition of this traffic model can be found below. For the analysis of single-stage switch with finite buffers the same assumptions are used. In order to verify the analytical results an N x N single-stage LB switch simulator was written.

We describe the traffic load of the switch with an arrival probability matrix (4.1) of dimension N x N:

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,0} & a_{N-1,1} & \dots & a_{N-1,N-1} \end{pmatrix} \tag{4.1}$$

The i, j element of the arrival probability matrix $a_{i,j}$ is the probability that a cell arrives to input i which is destined to output j in a given time slot. Indeed, we assume that the number of cells (and their input output port assignment) arrives to the switch in consecutive time slots are independent identically distributed random variable. This way matrix A completely characterizes the traffic process. It is defined to be *admissible* if for stationary ergodic arrival process [9] $A'(n) = [A'_{i,j}(n)]$ (denote the number of cells that have arrived at input i destined to output j) together with law of large numbers:

$$\lim_{n\to\infty} \frac{A'_{i,j}(n)}{n} = a_{i,j}, \tag{4.2}$$

the equations (4.3) (4.4) hold. Since no more than one cell can arrive to a given input we have

$$\sum_{j=0}^{N-1} a_{i,j} < 1, \quad i = 0 \dots N-1 \tag{4.3}$$

and we say that output j is not overloaded if

$$\sum_{i=0}^{N-1} a_{i,j} < 1. \tag{4.4}$$

When (4.4), does not hold the traffic load is referred to be inadmissible [28]. Inadmissible traffic load results in infinite queue length and delay in case of infinite buffer switches, but it can be analyzed in the same way as admissible in case of finite buffer switches.

4.3 The Packet Loss Analysis: 2 x 2 LBS case

This section describes behavior of the simple 2 x 2 case LB switch depicted in Figure 4.2 and presents its mathematical analysis.

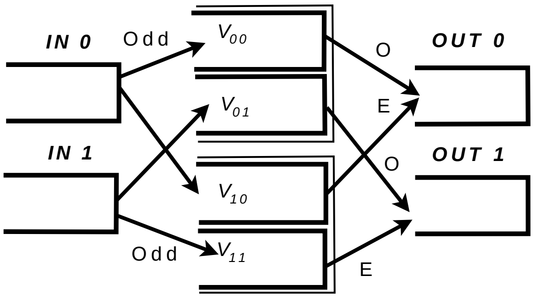


Figure 4.2: 2 x 2 LB switch

The arrival probability matrix for the 2 x 2 LB switch is

$$A = \begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{pmatrix} \tag{4.5}$$

The goal of our analysis is to describe the occupancy of a virtual output queue in the central stage of the switch in time. In particular, we consider $V_{0,0}$ for our study. Let $vq_{i,j}(n)$ be the occupancy (the number of cells) in the VOQ from input port i to output port j at time slot n. $vq_{i,j}(n)$ can change in every time slot, from 1 to n. As we are considering the case of 2 x 2 switch we describe the evolution of the occupancy function during two time slots, because $V_{0,0}$ has a kind of periodic behaviour with a two time slots period. We sort time slots to odd and even ones. An example of possible LB switch arrivals and VOQ occupancy is presented in the Table 4.1 and Figure 4.2.

ports\timeslot	*Odd*	*Even*	*Odd*
Out_0			1
$V_{1,0}$	0	1	1
$V_{0,0}$	0	1	2
In_1	$(a_{1,0})$ 0	$(a_{1,0})$ 0	0
In_0	$(a_{0,0})$ 0	$(a_{0,0})$ 0	1

Table 4.1: LB switch occupancy during three time slots

Table 4.1 represents the case when during the first *Odd* time slot the system is empty and only two cells arrive to the input ports (both destined for *output*0) according to input probabilities $a_{0,0}$ and $a_{1,0}$. During the next (*Even*) time slot the cells from the *Odd* time slot are moved to the corresponding $VOQs$ (e.g. $V_{0,0}$ and $V_{1,0}$) and two more cells arrive to the inputs (again destined to *output*0). Finally, during the third time slot due to crossbar round-robin switching properties only one cell is sent to *output*0, so one of the $VOQs$ queue starts building up ($V_{0,0}$ or $V_{1,0}$). Accordingly, for the time slot $n+1$ the recursive occupancy function will be the following:

$$vq_{0,0}(n+1) = max[vq_{0,0}(n) + I_{n \text{ is odd}}\alpha_{0,0} + I_{n \text{ is even}}\alpha_{1,0} - I_{n \text{ is odd}}, 0], \tag{4.6}$$

$func.\backslash probability$	$(1-a_{0,0})(1-a_{1,0})$	$(1-a_{0,0})a_{1,0}$	$a_{0,0}(1-a_{1,0})$	$a_{1,0}a_{0,0}$
$\alpha_{0,0}$	0		1	
$\alpha_{1,0}$	0	1	0	1
$\alpha_{1,0}+\alpha_{0,0}$	0	1	1	2
$\alpha_{1,0}+\alpha_{0,0}-1$	-1	0	0	1
$Probability$	p_{-1}	p_0		p_1

Table 4.2: Possible variation of the function values versus arrival probabilities

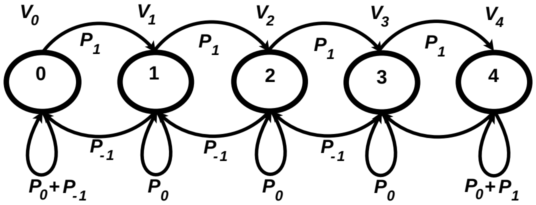


Figure 4.3: State-transition diagram for $2x2$ LB switch with central buffer of size $B = 4$

where $\alpha_{i,j}$ is the binary random variable representing the number of cells arrived to input i and destined to output j in the given time slot, and I is the indicator operator with two possible values:

$$I_{condition} = \begin{cases} 1, & if\ condition\ is\ true; \\ 0, & otherwise. \end{cases}$$

Depending on the actual value of n (4.6) simplifies to

$$vq_{0,0}(n+1) = max[vq_{0,0}(n) + \alpha_{0,0} - 1], \quad \text{if } n \text{ is odd},$$

$$vq_{0,0}(n+1) = vq_{0,0}(n) + \alpha_{1,0}, \ \text{if } n \text{ is even}.$$

Putting together the effect of two consecutive time slots we have

$$vq_{0,0}(n+2) = max[vq_{0,0}(n) + \alpha_{1,0} + \alpha_{0,0} - 1, 0]. \tag{4.7}$$

We analyze the behavior of the VOQ based on (4.7), which describes its evolution in time. Since $\alpha_{0,0}$ and $\alpha_{1,0}$ are independent binary random variable, $vq_{0,0}(n+2)$ is bounded as follows $vq_{0,0}(n) - 1 \leq vq_{0,0}(n+2) \leq vq_{0,0}(n) + 1$. The possible values of $\alpha_{0,0}$ and $\alpha_{1,0}$, the associated variation of $vq_{0,0}(n+2)$ and the corresponding probabilities are presented in the Table 4.2.

Indeed (4.7) describes a *Discrete Time Markov Chain* (DTMC) whose state transition probabilities are p_{-1}, p_0, p_1 (from Table 4.2). The state transition graph of this DTMC is presented in Figure 4.3 for $V_{0,0}$. The process can have three possible transitions. With probability p_0 the process stays in the same state, while with probability p_1 it moves to the next state (which represents one more cell in the buffer) and with probability p_{-1} it returns to the previous state. The three-diagonal one step state transition probability matrix, P, composed of the probabilities, p_i, is

$$P = \begin{pmatrix} p_{-1}+p_0 & p_1 & 0 & \dots & 0 & 0 \\ p_{-1} & p_0 & p_1 & \ddots & 0 & 0 \\ 0 & p_{-1} & p_0 & p_1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & p_{-1} & p_0 & p_1 \\ 0 & \dots & \dots & 0 & p_{-1} & p_0+p_1 \end{pmatrix}. \tag{4.8}$$

When the buffer is finite, B, the evolution of the DTMC is bounded at the buffer size. Figure 4.3 presents the transition graph of the case when the buffer size is 4. In this case packet loss occurs with probability p_1 when buffer is full.

Let V_i denote the steady state probability that i cells are present in the VOQ and V the row vector composed by the stationary state probabilities, i.e., $V = \{V_0, V_1, \dots, V_B\}$. The stationary distribution of the DTMC can be obtained as the solution of the linear system ([41])

$$V\ P = V,\ VH = 1, \tag{4.9}$$

where H is the column vector whose elements are equal to one.

For this simple DTMC structure (4.13) has a closed from solution [41],

$$V_i = \frac{1-\rho}{1-\rho^{B+1}}\rho^i,$$

with $\rho = p_1/p_{-1}$. For finite buffer VOQ with buffer size B the loss probability of the system is simply the probability that an arriving cell finds the system full. This way the loss probability, L, is the stationary probability of the final state, V_B, multiplied with probability p_1, i.e.,

$$L = V_B p_1 = \frac{1-\rho}{1-\rho^{B+1}}\rho^B p_1. \tag{4.10}$$

Based on (4.10) the loss probability can be calculated from the traffic matrix, A. The simulation and analytical results for 2 x 2 LB switch are presented in section 4.5.

4.4 The Packet Loss Analysis: N x N LBS case

In this section we present the mathematical model used for cell loss probability analysis of N x N LB switch ($N \geq 2$) with finite buffers. Please note, that as a basement to the following mathematical analysis, the *batch-Geo*$/D/1/K$ queueing model was used [27,61]. We model the system as a time homogeneous discrete-time,discrete-state Markov chain.

This model was modified in correspondence to the LB switch operation principles and just present an analysis of loss probability of a central stage VOQ during its operation, without complete derivation of queue length and unfinished work distributions. Figure 4.4 illustrates the architecture of the N x N LB switch.

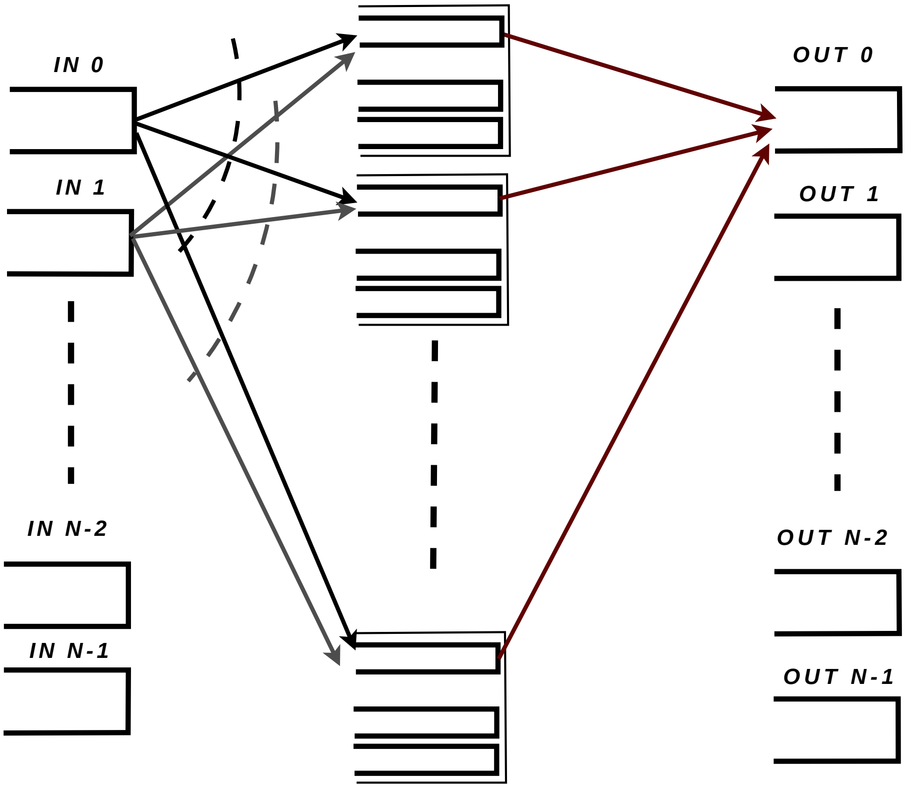


Figure 4.4: N x N LB switch with finite buffers

The goal of our analysis is to describe the occupancy of a VOQ in the central stage of the switch in time. In particular, we consider $V_{0,0}$ for our study. Let $vq_{i,j}(n)$ be the occupancy (the number of cells) in the VOQ from input port i to output port j at time slot n. $vq_{i,j}(n)$ can change in every time slot, from 1 to n. As we are considering the case of N x N switch we describe the evolution of the occupancy function during N time slots, because $V_{0,0}$ has a kind of periodic behavior with a N time slots period.

The evolution of $vq(n)$ is characterized by the following equation

$$\begin{aligned} vq_{0,0}(n+1) = max[vq_{0,0}(n) + I_{(n\ mod\ N=0)}\alpha_{0,0} + I_{(n\ mod\ N=1)}\alpha_{1,0} \quad (4.11) \\ + \ldots + I_{(n\ mod\ N=N-1)}\alpha_{N-1,0} - I_{(n\ mod\ N=0)}, 0], \end{aligned}$$

where $\alpha_{i,j}$ is the binary random variable representing the number of cells arrived to input i and destined to output j in the given time slot, and I is the indicator operator with two

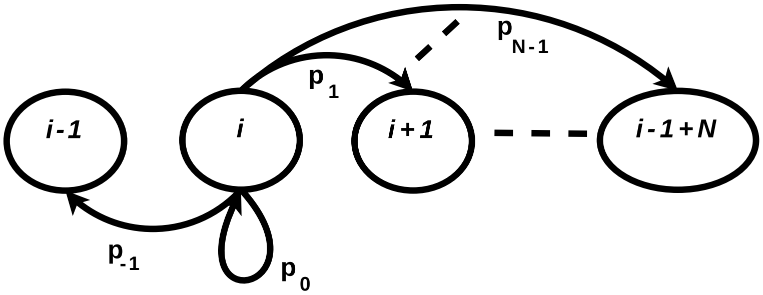


Figure 4.5: Possible transitions of $N \times N$ LB switch when $B > i - 1 + N$

possible values:

$$I_{condition} = \begin{cases} 1, & if condition is true; \\ 0, & otherwise. \end{cases}$$

(4.11) represents the following two main cases:

$$\begin{aligned} &\text{if } n \bmod N = 0, \text{ then} \\ &\qquad vq_{0,0}(n+1) = max[vq_{0,0}(n) + \alpha_{0,0} - 1, 0], \\ &\text{if } n \bmod N > 0, \text{ then} \\ &\qquad vq_{0,0}(n+1) = vq_{0,0}(n) + \alpha_{i,0}. \end{aligned}$$

According to (4.11) an N time slots long interval of $vq_{0,0}(n)$, when $n \bmod N = 0$, is characterized by

$$vq_{0,0}(n+N) = max[vq_{0,0}(n) + \sum_{i=0}^{N-1} \alpha_{i,0} - 1, 0]. \tag{4.12}$$

Indeed $vq_{0,0}(n)$ is non-decreasing during the first $N-1$ time slots and non-increasing during the last time slot.

The process can have N possible transitions from state to the others(state represents number of cells in the buffer). Transition state probabilities could be represented by the following notations:

$$p_i = Pr\left(\sum_{i=0}^{N-1} \alpha_{i,0} - 1 = i\right), \quad \forall i \in \{-1, 0, 1, \ldots, N-1\},$$

where $a_{i,0} = Pr(\alpha_{i,0} = 1)$.

In case of inhomogeneous traffic matrix we have:

$$
\begin{aligned}
p_{-1} &= \prod_i (1-a_{i,0}),\\
p_0 &= \frac{1}{1!}\sum_i \left(a_{i,0} \prod_{j,j\neq i} (1-a_{j,0}) \right) = p_{-1}\frac{1}{1!}\sum_i \frac{a_{i,0}}{1-a_{i,0}},\\
p_1 &= \frac{1}{2!}\sum_i \left(a_{i,0} \sum_{j,j\neq i} \left(a_{j,0} \prod_{k,k\neq i,j} (1-a_{k,0}) \right)\right) =\\
&= \frac{1}{2!}p_{-1}\sum_i \left(\frac{a_{i,0}}{1-a_{i,0}} \sum_{j,j\neq i} \frac{a_{j,0}}{1-a_{j,0}} \right),\\
p_2 &= \frac{1}{3!}p_{-1}\sum_i \left(\frac{a_{i,0}}{1-a_{i,0}} \sum_{j,j\neq i} \left(\frac{a_{j,0}}{1-a_{j,0}} \sum_{k,k\neq i,j} \frac{a_{k,0}}{1-a_{k,0}} \right)\right),\\
&\dots\\
p_{N-1} &= \frac{1}{N!}\prod_i a_{i,0},
\end{aligned}
$$

where the limits of the summations and products are 0 and $N-1$.

In case of a homogeneous traffic matrix, i.e., when $a_{i,0} = a \ \forall i \in \{-1, 0, 1, \dots, N-1\}$, these transition probabilities simplify to

$$
\begin{aligned}
p_{-1} &= (1-a)^N,\\
p_0 &= \binom{N}{1} a^1 (1-a)^{N-1},\\
p_1 &= \binom{N}{2} a^2 (1-a)^{N-2},\\
&\dots\\
p_i &= \binom{N}{i+1} a^{i+1} (1-a)^{N-i-1}.
\end{aligned}
$$

Equation (4.12) represents a DTMC, whose transition structure is depicted in Figure 4.5 (only for the case when $B > i-1+N$). Figure 4.5 does not take into account cell loss. On the contrary, we present the transition graph for $N = 3$ and $B = 5$, where cell loss appears in the final states (Figure 4.6).

The transition probability matrix of the process is presented in Figure 4.7. The transition probability matrix is composed of $N+1$ non-zero diagonals. As the finite buffer case is considered, the dimensions of the transition probability matrix ($(B+1)\times(B+1)$) are directly related to buffer size B. The last column of matrix P is composed of irregular elements as the structure of transitions is changing when the buffer gets full (Figure 4.6).

Let V_i denote the steady state probability that i cells are present in the VOQ and V the row vector composed by the stationary state probabilities, i.e., $V = \{V_0, V_1, \dots, V_B\}$. The stationary distribution of the DTMC can be obtained as the solution of the linear system ([41])

$$V\ P = V,\ VH = 1, \tag{4.13}$$

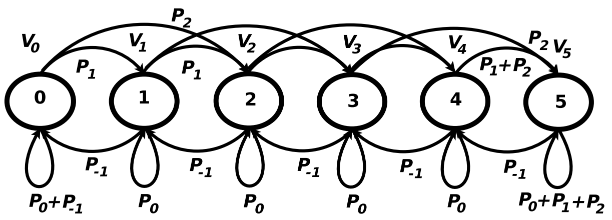


Figure 4.6: The transition graph of LB switch with $N = 3$ and $B = 5$.

$$P = \begin{pmatrix} p_0 + p_{-1} & p_1 & p_2 & \dots & p_{N-1} & 0 & \dots & 0 \\ p_{-1} & p_0 & p_1 & \dots & p_{N-2} & 0 & \dots & 0 \\ 0 & p_{-1} & p_0 & \dots & p_{N-3} & 0 & \dots & 0 \\ 0 & 0 & p_{-1} & \dots & p_{N-4} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \dots & \vdots & & \\ 0 & 0 & \dots & 0 & \dots & p_{-1} & p_0 & \sum_{i=1}^{N-1} p_i \\ 0 & 0 & \dots & 0 & \dots & 0 & p_{-1} & \sum_{i=0}^{N-1} p_i \end{pmatrix}$$

Figure 4.7: The transition probability matrix Q

where H is the column vector whose elements are equal to one.

In contrast with the case when $N = 2$, the state transition graph of the $N \times N$ switch (when $N > 2$) does not exhibit a simple birth-death structure and, consequently, the stationary probability vector does not have a simple closed form solution.

In order to compute the loss probability we create column vector L, whose ith element, L_i, defines the mean number of lost cells in a cycle when the buffer occupancy is i at the beginning of the cycle. In state B the mean number of lost cells is $\sum_{j=1}^{N-1} jp_j$. In general,

$$L_i = \begin{cases} 0 & \text{if } i \leq B - N + 1 \\ \displaystyle\sum_{j=1}^{i-B+N-1} jp_{B-i+j} & \text{if } i > B - N + 1 \end{cases} \tag{4.14}$$

From which the structure of vector L is as follows.

$$L = \begin{bmatrix} 0 \\ \vdots \\ p_{N-1} \\ 2p_{N-1} + p_{N-2} \\ 3p_{N-1} + 2p_{N-2} + p_{N-3} \\ \vdots \\ \sum_{j=1}^{N-1} jp_j \end{bmatrix}, \tag{4.15}$$

Having vector V and vector L the loss probability is obtained as

$$P(loss) = V\ L. \tag{4.16}$$

The simulation and analytical results for the cell loss inside the N x N LB switch are presented in section 4.5.

4.5 Computational Results and Summary

In this section, we present a computational study of a single-stage buffering, LB switch of size $N \geq 2$ under various finite buffer sizes $B \geq N$ and traffic loads. In particular we consider both (1) admissible, and (2) overloading inadmissible [28] incoming traffic conditions.

There are two objectives of the computational study. First, we aim to compare the accuracy of analysis and simulations results under admissibility assumption. Second, we calculate by means of simulations the cell loss probabilities under inadmissible traffic.

Following the work in [18], [40], we omit the cell mis-sequencing issues from the computational study.

4.5.1 Numerical and simulation model

Next we define the arrival rate matrices A used for traffic scenarios(represented also in [56]) as follows:

Uniform i.i.d: $a_{i,j} = \rho/N$,

Unbalanced i.i.d:

$$a_{i,j} = \begin{cases} \rho(W + \frac{1-W}{N}), & if i = j, \\ \rho\frac{(1-W)}{N}, & otherwise. \end{cases}$$

where W is an unbalanced probability. When $W = 0$ the traffic is uniform, but in case when $W = 1$ the system will have "uniform" hotspot.

Hotspot to single output: The system has a hotspot to the output K ("many-to-one" hotspot) if

$$a_{i,j} = \begin{cases} \rho(W + \frac{1-W}{N}), & if j = K, i = \{0 \ldots N-1\}, \\ \rho\frac{(1-W)}{N}, & otherwise. \end{cases}$$

We also compared the behavior of the scheme under admissible uniform i.i.d, unbalanced i.i.d and "many-to-one" hotspot traffic matrices. Finally, we proceed with simulations for uniform inadmissible traffic (Figure 4.9).

4.5.2 Results and interpretations

In our initial experiments all the arrival rate matrix elements were set to $\frac{\rho}{N}$ (N x N LB switch, Figures 4.8 and 4.9). By carefully examining our results we found out that the values of cell loss probability for admissible uniform i.i.d traffic and "many-to-one" hotspot model are similar. If to remember that our analysis is based on the cells occupancy of central stage $VOQs$ with respect to some output, it becoming obvious that in case of uniform i.i.d traffic and considered "many-to-one" hotspot model, the loading of corresponding column in the arrival rate matrix will be similar. However, the complete match in the results will be valid only for the case of admissible traffic matrix. Correspondingly, for inadmissible case the assumption expressed in equation (2) will not be true anymore, as the loadings of specific columns will be different (like the probabilities $p_{-1}, p_0, \ldots etc.$).

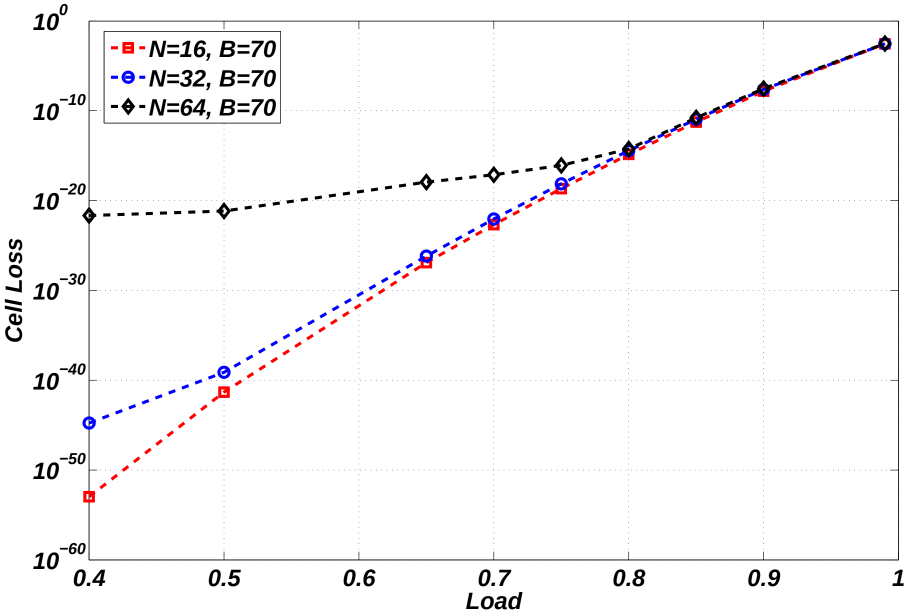


Figure 4.8: Cell loss versus load ($\rho = 0.4 \ldots 0.99$) for N x N LB switch, buffer=70 cells, admissible uniform i.i.d traffic.

The simulations related to the N x N switch loss probability under inadmissible traffic model was depicted in Figure 4.9. We consider the load to be in range of $0.9 \ldots 1.99$. As each stage of the switch obey non-work conserving policy, the amount of cells sent to the specific output (under overload) will remain in the central stage buffers during undefined period of time. In other words arrival service rate of the system is becoming greater than departure service rate (Figure 4.9 - twice doubling the system load will lead to 50% internal loss probability). As a result finite buffers will experience fast queue build up process and congestion inside.

If to consider this situation in context of variable length packets the throughput degra-

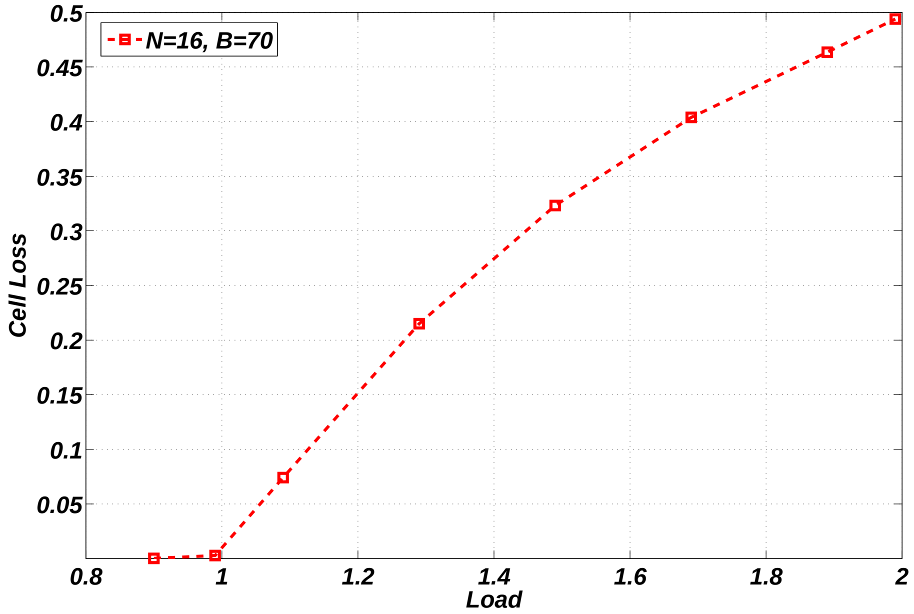


Figure 4.9: System loss versus load for N x N LB switch, overall load range $\rho = 0.9 \ldots 1.99$, buffer = 70 cells, inadmissible traffic.

dation might be even much higher. We suspect the packet loss probability can have a multiplicative effect related to cell loss probability. In this case, the reassembly unit have to continuously exclude the remaining cells of the "broken" packet from the system. Note, that here we assuming the case when there is no feedback between the stages is implemented, so inputs do not have any information about occupancy of the central stage queues and traffic arriving to all the other inputs.

Figure 4.10 shows dependence of internal cell loss regarding to a buffer size under the maximum load. As it is expected, the loss probability is intensively decreasing with increase of buffer size, and press towards zero (and throughput will reach 100% [17], [40]) in ideal infinity buffer size case.

Graph 4.11 shows dependence of central stage buffers cell loss probability versus size of the switch under unchanged buffer size. In this case LB switch presents behavior similar to Input Buffer switch under uniform traffic. As the switch size is increasing, the probability that additional cells will arrive to the specific queue during a time cycle is also increasing. This fact is important to take into account while adding new line cards to scale up system performance. While increasing operating performance one can also increase the internal cell loss probability if the appropriate buffering is not used.

For the final experiment (Figure 4.12), we choose uniform i.i.d ($W = 0.2$), unbalanced i.i.d ($W = 0.5$) and 'uniform' hotspot traffic models ($W = 0.8$) with overall load in range $\rho = 0.4 \ldots 0.99$ and fixed buffering of 20 cells. With increase of unbalanced probability W, the diagonal values of arrival rate matrix are considerably higher than all other elements,

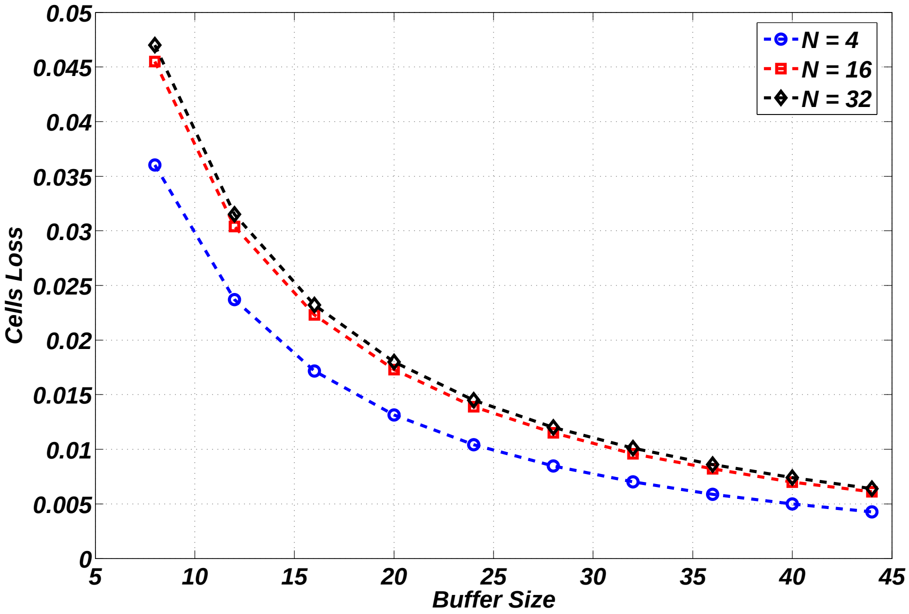


Figure 4.10: Cell loss versus buffer size, overall load 0.99, admissible uniform i.i.d traffic.

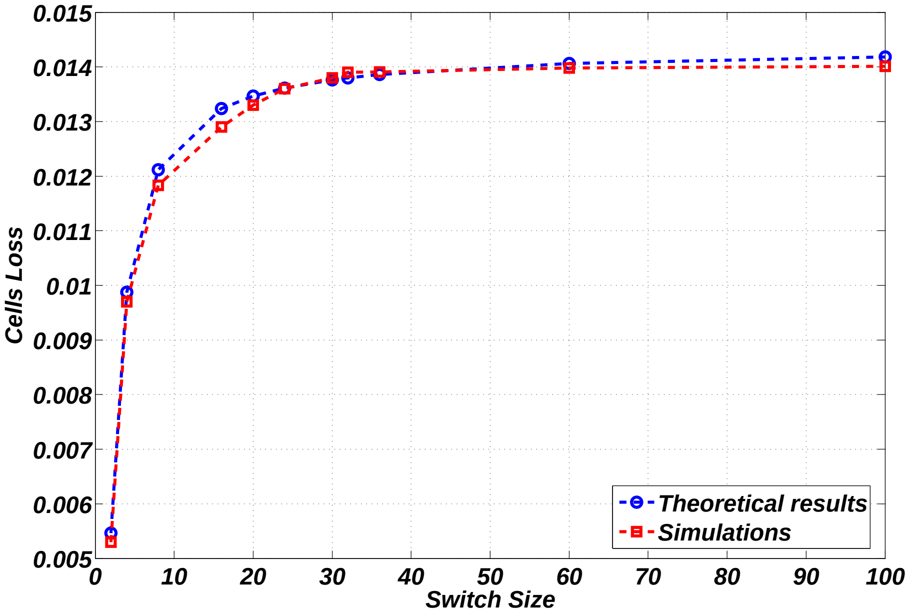


Figure 4.11: Cell loss versus switch size, overall load 0.99, buffer size = 25 cells, admissible uniform i.i.d traffic.

i.e each input has different hotspot. This implies that each input is sending high portion of traffic to the specific output while transmitting negligible amount to all the others. However, even in this case, input traffic remains perfectly balanced between the outputs because of round-robin cell spreading in the first stage crossbar. As a result, the loss probability results for ”uniform” hotspot are appearing to be the smallest among other admissible traffic matrices (Figure 4.12). Each of the graphs was compared also with

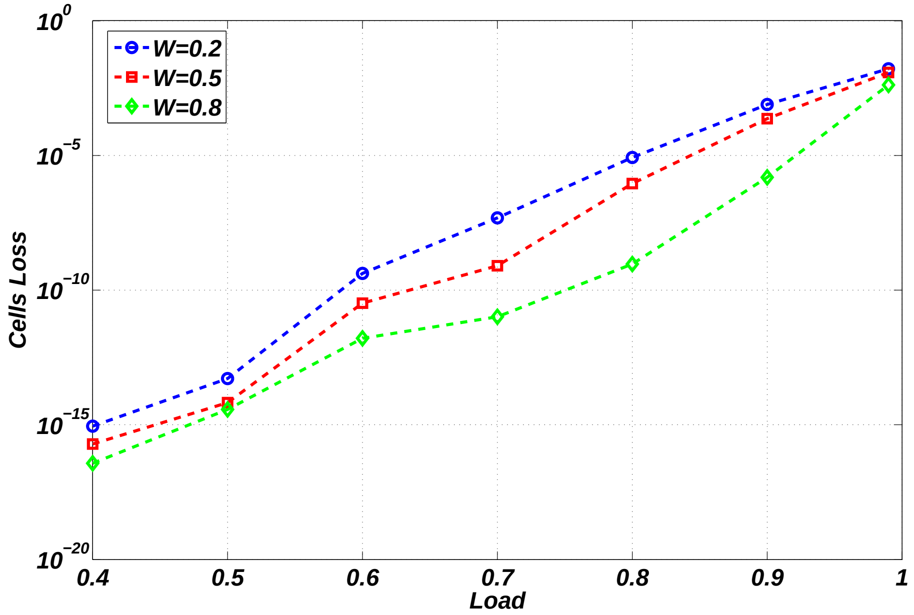


Figure 4.12: Cell loss versus load, 16 x 16 LB switch, buffer size = 20 cells, admissible unbalanced i.i.d traffic, for different W.

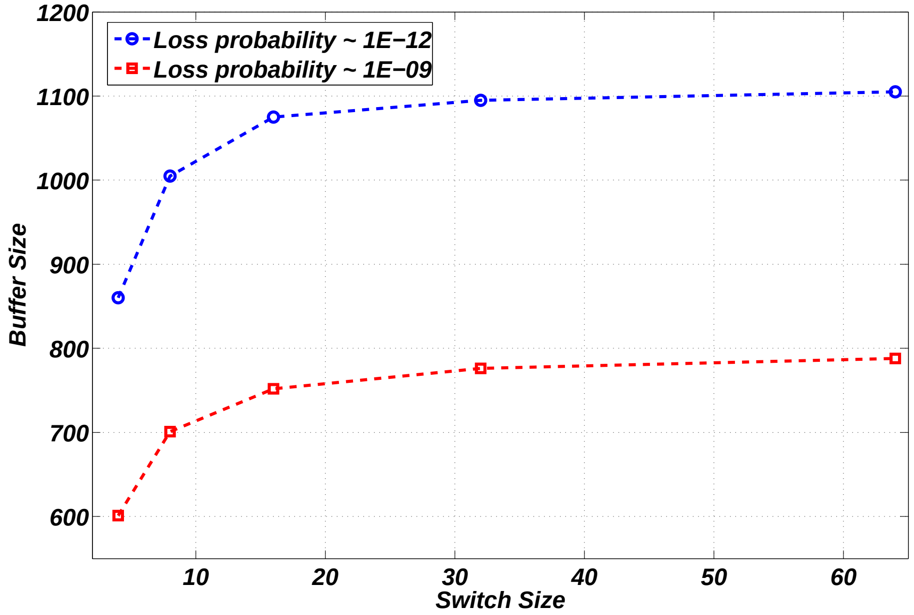


Figure 4.13: Buffer size versus switch size with loss probability $1E-12$ and $1E-09$, for overall load of 0.99 (admissible uniform i.i.d traffic).

simulation results, the spread in the results was presented in the Figure 4.11.

The results demonstrate that the loss probability can be small for admissible input traffic. However, as shown in Figure 4.13, when the admissible traffic is near full load, say 0.99, there can be significant loss if we would like to maintain the packet loss probability with similar values optical fiber transmission, say $1E-12$ and $1E-09$. This may be significant for streaming media (primarily video) applications that are expected to occupy more than 90% of the Internet traffic and will be transmitted using the unreliable UDP (user datagram protocol). Consequently, in the case of multi-hop networks the traffic may become inadmissible, even for a short period of time, will require large number of buffers in order to maintain cell loss in an acceptable level. This may become even worse when variable size packets are transmitted, since a single cell loss will cause a whole packet (of multiple cells) loss. The details of such traffic scenarios will be studied in the next chapter.

Chapter 5

The LBS with Variable Size Packets

Due to a large variety of Internet applications, real IP network traffic is composed of the packets with variable size. Packets arriving to the switching node are commonly stored in the input buffers while the packet header is processed. As soon as the egress port is evaluated the packet is forwarded to the output network. It is well known that the scheduling algorithm which interconnects input and output ports is much simpler in the systems which are segmenting arriving variable size packets into the equal size data cells. As most of the switching systems are operating on the time slot basis, the decision-making algorithm should perform ingress-egress port interconnection during a finite time. Consequently the fixed size cells allow to perform scheduling at a fixed time interval.

In Chapter 3 we have introduced a novel issue of a possible packet drop in the LB switch operating with the fixed size data cells. The detailed investigation on this issue was performed in Chapter 4 [69] where the mathematical analysis for internal cell loss evaluation was proposed.

In contrast to the previous research [69] this chapter presents analysis of the internal LBS packet loss probability while considering the fact that variable size packets are sent through the system. The main difference lies in the assumption that each cell of a packet is not independent from the others. Thus the previously mentioned problem of a single cell drop of a given packet will result in a drop of the whole packet itself.In the following we show that the amount of obtained packet loss can be considerably higher that the corresponding cell loss of that specific system. In order to present mathematical model a set of new assumptions was made. In particular, we assumed the input traffic to have Markovian behavior to be able to use numerically efficient algorithms in order to solve Markov chains. For our system this means geometrically distributed packet and idle period lengths, making easy to capture the mean of these distributions. Real internet traffic shows different packet size distributions [63] and one can fit more parameters using other, more complex Markovian structures like discrete Phase Type (DPH) distributions or discrete Markovian arrival processes (DMAPs). The number of fitted parameters can be increased at an arbitrary level, but it would greatly increase the complexity of the model as well and that would also hide the main contribution of our approach.

Moreover, as it will be shown in our analytical results, the packet loss probability of a VOQ strongly depends on the specific traversing path of the traffic inside the switch, which is an interesting phenomenon described in Section 5.2.1 for the interconnection pattern applied. We analyze the least preferred path among the possibilities.

In the following chapter the analytical part is divided into three main sections. The first section 5.2 presents the mathematical model for evaluation of the LB switch cell and packet loss probabilities. The model is giving a complete characterization of input processes which appear at ingress ports. In particular, an input arrival process is described by a Discrete Time Markov Chain which took into account all the possible packet types arrivals(with a specific output) and an idle interpacket period. Unfortunately, with the enlargement of the switch size the size of the Discrete Time Markov Chains, used for characterization, is also expanding. The overall algorithm complexity results in $O(N^N)$, and does not allow to perform evaluation for large LBS systems. As the LB switch is a reason of chose when the switch (or Central Stage buffer) size is large, we present two approximation mathematical models for the packet loss evaluation. The first approach is derived with the assumption of inhomogeneous traffic at inputs, as it was proposed in the previous model. This means that each input can have packets with various sizes and idle periods. In spite of this fact, the input arrival process was characterized by a DTMC with only two states, which allow us to reduce overall complexity of the mathematical model to the $O(2^N)$. The proposed algorithm allowed to enlarge evaluation capabilities in comparison with the first model, while the complexity remain exponential. In order to scale up the evaluation space of our algorithm, we assumed that all input arrival processes are identical and as the evaluation in all the models was proposed for a specific internal path, the input arrival processes of the whole system were characterized by $(N+1)$ states in a DTMC. This assumption allowed to perform evaluation of the packet/cell loss analysis for large switch sizes.

The rest of the chapter is organized as follows. In Section 5.1 we present the main operation principles of the considered LB switch architecture. Next, the detailed analysis of a switch with arbitrary number (N) of ports in Section 5.2 will be given. Since the analysis presented in Section 5.2 has high computational complexity we show the approximated model for packet loss evaluation in Section 5.3. As a next step, the linear complexity model for evaluation of performance characteristics of the switch will be presented in Section 5.4. Finally, in Section 5.5 we will present computation results and conclusions for all the models presented in this chapter.

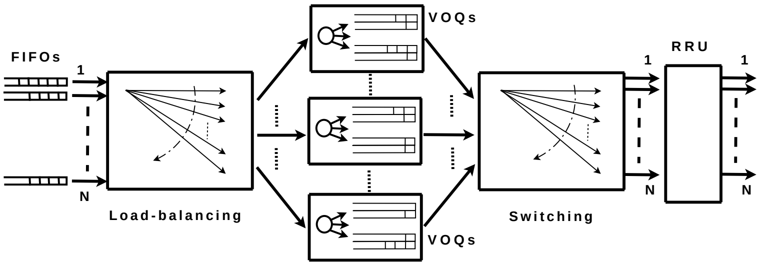


Figure 5.1: The considered LB switch

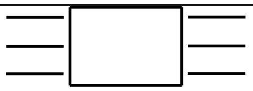
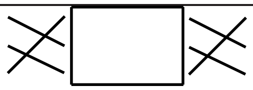
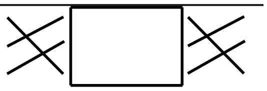
5.1 The Considered Load-Balancing Switching Architecture

Throughout this chapter we assume that the multi-stage buffering LB switch is equipped with First-In-First-Out (FIFO) buffers at the inputs and re-sequencing and reassembly units (RRU) at the outputs (see the illustration in Fig. 5.1). The implementation of RRU is not discussed in this chapter, but it can be used as one among the proposed in research, like in [66]. As a standard LB switch, the considered system has two crossbar switches placed in between of the central stage buffers. The central stage buffers are designed as a set of N units with N VOQs in each, where one VOQ is used for a specific output. The crossbar switches are running through a predefined round-robin interconnection pattern. Finally, we assume that the system does not use any additional information exchange links between the switch stages, e.g. each stage is operating independently.

The finite FIFO buffers at the inputs (Figure 5.1) are large enough to store as much packets as the loss probability remains under a predefined threshold. All the packet drops at the input stage are not taken into account at the central stage buffers. Nevertheless, if a packet is dropped, it is assumed that this packet is removed completely from the system and can be retransmitted by a network protocol in the future. Such kind of packet loss usually does not create any system instability which can result in buffer wastage and long end-to-end delays. The main disadvantage of the considered LB switch is a possible cell drop at the central stage buffers. Due to the fact that the system does not perform any control on the amount of traffic sent from the input stage to the central stage, and the traffic tends to be uniformly balanced between each set of VOQs, a cell loss is possible at some point. In this case, the system is not able to reset all the remaining cells of the “broken” packet from the central stage buffers where these cells were already independently distributed. These can lead to the following consequences. Firstly, the central stage buffers capacity is continuously wasted by the transmission of cells of a ”broken” packets. Secondly, the re-sequencing and reassembly units will not be able to reconstruct these packets correctly.

In our analysis, each variable size packet arrives with a variable rate, which is always less than service rate of cells inside the switch – the switch is not overloaded. After this the system can have an idle period (measured in time slots and modelled using geometric

Table 5.1: The three possible settings of 3×3 LB switch

$t \mod N$	0	1	2
switch state			

distribution). Similarly to presented in Chapter 4.1 a cell transmission rate inside the switch is fixed. Moreover, it is assumed that within a time slot duration, the VOQs are firstly connected to outputs and then inputs are interconnected with the VOQs. This order of interconnections inhibit cells from traversing the switch in a single time slot.

The destination outputs of the packets are chosen uniformly among all the available outputs (this is given in **T** parameter in our analysis). The analysis is done without any respect to cells mis-sequencing inside the switch and packets reassembly. The main goal of the presented analytical models are to show the amount of packet/cell loss experienced by a single *central stage VOQ*.

5.2 The General Case Packet Loss Analysis

The following section contains the stochastic model in order to evaluate the internal packet loss probability of the LB switch. Throughout the chapter we will use the notation $n \times m$ to denote a switch with n input and m output ports or simply refer them as a switch of size N if there are both N input and output ports. However, we first perform the packet/cell loss analysis for the 3×3 LB switch which is less complicated than $N \times N$ case and has its all important properties. Due to the fact that the incoming traffic is uniformly distributed among the outputs and that the stages are interconnected using round-robin policy (5.1) we assume that all the central stage buffer queues (VOQs) show similar behavior. However, as it will be shown in the next section, the packet loss probability of a VOQ is also depends on the transmission inputs and destination outputs. For our analysis we choose a specific central stage buffer queue - VOQ_{00},, which can have packet arrivals from all three inputs.

The crossbar interconnections between input i, cental stage VOQ_{kj}, and output j obey the rules

$$\begin{aligned} k &= i + t \mod N \\ j &= k + t \mod N \end{aligned} \tag{5.1}$$

respectively. Due to these, the interconnection pattern of the switch has a periodic behavior with length N. In case of size 3 the possible interconnection settings are summarized in Table 5.1.

First, in Section 5.2.1, we will show our observations on the different behavior of the packet loss while considering different traffic paths. As the packets are segmented into fix-sized cells the arrival process to a VOQ can be described by a discrete time Markov chain (DTMC) on the cell level (see Section 5.2.2). Extending the DTMC with two absorbing states we are able to model the system on the packet level (Section 5.2.3) . Having the packet level model we give its solution in Section 5.2.4.

5.2.1 Properties of the different paths

One of the most important findings of our analysis is that there are differences between the loss probabilities of paths traversing the switch. Here path means the triple, $\{i, j, k\}$, containing the ordinal number of the input, the output and the VOQ respectively.

Using the interconnection pattern policy given in (5.1) the time difference between the service of the VOQ and the arrival to it can be expressed as

$$\Delta t = (2k - i - j) \mod N. \tag{5.2}$$

Δt also expresses the number of inputs that have the right to send a packet to VOQ$_{kj}$ before observed input i during the same time period.

A particular VOQ is served once in a time period. During this time period all the inputs have a right to send cells to the VOQ (in a particular order determined by (5.2)). In case of “almost full” buffer the higher the Δt value is the higher the probability that there are enough inputs that can fill up the buffer, i.e., make the cell of the observed input to be lost. According to this observation we introduce the notation type-Δt for paths with value Δt.

This implies the difference between different paths as, in case of a given VOQ, it depends on the ordinal number of both the input and the output. The possible Δt values are $0, 1, \ldots, N-1$ time slots. There are two important consequences of this observation.

Differences in cell loss Depending on the value of $\mathit{\Delta t}$ there can be N values of cell loss probabilities – $p_{cl} \quad l \in [0, N]$, as well as N values of the packet loss probabilities – $p_{p_l} \quad l \in [0, N]$. The following set of inequalities holds for the packet/cell loss probabilities

$$\begin{aligned} p_{c0} &= 0 \leq p_{c1} \leq \ldots \leq p_{cN-1}, \\ p_{p_0} &= 0 \leq p_{p_1} \leq \ldots \leq p_{p_{N-1}}, \end{aligned} \tag{5.3}$$

which is explained by the fact that the higher the $\mathit{\Delta t}$ value is, higher the loss probability can be.

In a time period, $\mathit{\Delta t}$ inputs have the right to send a cell to the observed VOQ before the observed input. Consequently, the higher the $\mathit{\Delta t}$ value the larger the chance that there are enough inputs sending cells before the observed input to fill up the queue.

Differences in other performance measures Together with a packet and cell loss, the queues with larger Δt, by default, will suffer larger transmission delays.

The interconnection patterns of the input and output crossbars basically determine the fairness of service in the input – output pairs. Assuming that the crossbars are set to provide a symmetric (fair) chance for all the input – output pairs, we admit that the cells of an input – output pair (distributed in the central stage VOQs) will suffer different loss at different VOQs. This way the loss between an input – output pair is dominated by the VOQ where it has the maximal Δt value. To the best of our knowledge this property has not been reported yet.

Due to this phenomenon it is not relevant which configuration is analyzed. In the following we focus on the analysis of the path with the maximal Δt value and present the behavior of the other packet loss cases in Section 5.5.

5.2.2 The cell level model

In the following we describe the model of the 3 × 3 switch. We have chosen to model VOQ_{00} – the first sub-queue of the first set of VOQs – or, more specifically, the path input $1 \rightarrow VOQ_{00} \rightarrow$ output 0. If one substitutes the ordinal number of these ports and virtual queue into (5.2) it will result in $2 \cdot 0 - 1 - 0 \mod N = 2$, which the highest loss probability values corresponds to.

VOQ_{00} is "fed" by three input processes with geometric distributed packet lengths. Each input can have packets destined to any output.

First we model the operating mechanism on the cell level by building the appropriate DTMCs for the ith input (see Fig. 5.2). Each DTMC has four states denoted as follows

ij the states responsible for cell-arrivals from input i to output j and

$i\ id$ the state responsible for the idle period of input i.

The state transitions describe the beginning of either a new packet, of an idle period or the continuation of the incomplete packet. The graph of the DTMC modeling the ith input is given in Fig. 5.2 and the transition probability matrix of the ith input is given in (5.4), i.e. for example the substitution of $i = 1$ will result in the graph together with the state transition probability matrix of the second, observed, input. The probabilities appearing in the DTMC according to the state transition probabilities have the following meaning:

- $p_{ij}q_it_{ij}$ is the probability that a packet from the ith input to the jth output arrives in the actual time slot,
- $1 - p_{ij}$ is the probability that the packet from the ith input to the jth output is still in progress,

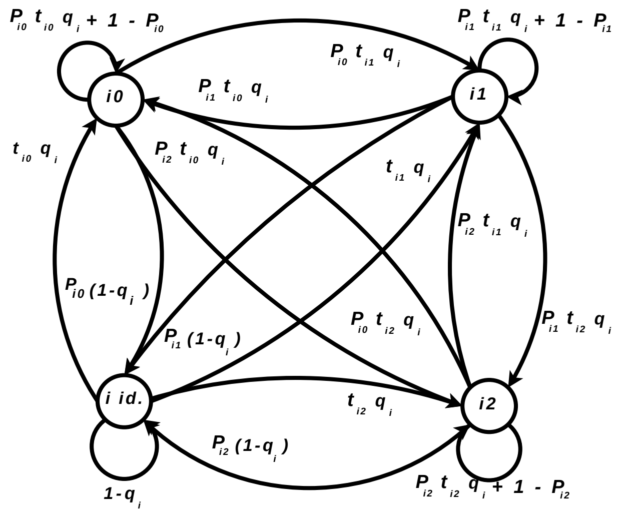


Figure 5.2: The DTMC modeling input i

$$\mathbf{P}_i = \begin{pmatrix} (1-p_{i0}) + p_{i0}q_it_{i0} & p_{i0}q_it_{i1} & p_{i0}q_it_{i2} & p_{i0}(1-q_i) \\ p_{i1}q_it_{i0} & (1-p_{i1}) + p_{i1}q_it_{i1} & p_{i1}q_it_{i2} & p_{i1}(1-q_i) \\ p_{i2}q_it_{i0} & p_{i2}q_it_{i1} & (1-p_{i2}) + p_{i2}q_it_{i2} & p_{i2}(1-q_i) \\ q_it_{i0} & q_it_{i1} & q_it_{i2} & 1-q_i \end{pmatrix} \tag{5.4}$$

- $p_{ij}(1-q_i)$ is the probability that the packet from input i to output j is ended and the ith input changes to idle,
- input i remains in idle state with probability $1-q_i$ in the actual time slot and
- $t_{ij}q_i$ is the probability that the next packet will be sent from input i to output j after the idle.

Hereinafter $\mathbf{T} = (\, t_{ij} \,)$ denotes the probabilities that a packet arrives from input i to output j. $\mathbf{P} = (\, p_{ij} \,)$ are the parameters of the geometric distributed packet length arrivals from input i to output j. $\mathbf{q} = (\, q_i \,)$ is the vector containing the parameters of the geometric distributed idle period of input i.

Let us split the state transition probability matrix of the ith input into two terms

$$\mathbf{P}_i = \mathbf{A}_i + \mathbf{K}_i = \begin{pmatrix} \mathbf{p}_i^1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{p}_i^2 \\ \mathbf{p}_i^3 \\ \mathbf{p}_i^4 \end{pmatrix} \tag{5.5}$$

where p_i^l denotes the lth row vector of matrix $\mathbf{P}_i$. The first term includes state transitions responsible for a cell arrival to output 0 – its first row equals to the first row of matrix $\mathbf{P}_i$ and it is 0 otherwise. The second term includes the other cases, which has zeros in the first row and it is the same as $\mathbf{P}_i$ otherwise.

$$\begin{aligned}\boldsymbol{\mathcal{P}} &= \mathbf{P}_0^3 \otimes \mathbf{P}_1^3 \otimes \mathbf{P}_2^3 = (\mathbf{A}_0 + \mathbf{K}_0)\,\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\,(\mathbf{A}_1 + \mathbf{K}_1) \otimes \mathbf{P}_2\,(\mathbf{A}_2 + \mathbf{K}_2)\,\mathbf{P}_2 = \\ &= \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{1 arrival}} + \\ &+ \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{2 arrivals}} + \\ &+ \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{3 arrivals}} + \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{no arrivals}} = \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{B}\end{aligned} \tag{5.8}$$

The system behavior during three consecutive time slots is described by a DTMC embedded right before the service of the VOQ at every $t \mod N = 0$th time instance. During this period, each input is modeled by the third power of its state transition probability matrix. The joint behavior of the three inputs in the period is described by the Kronecker product of the third power of the state transition probability matrices of each input. It is

$$\boldsymbol{\mathcal{P}} = \mathbf{P}_0^3 \otimes \mathbf{P}_1^3 \otimes \mathbf{P}_2^3, \tag{5.6}$$

which is the phase process of a quasi birth-deathlike (QBD-like) process describing the queue length of the observed VOQ.

In addition to (5.6)

$$\boldsymbol{\mathcal{P}} = \mathbf{B} + \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2 \tag{5.7}$$

also holds for the phase process of the same QBD-like. Here $\mathbf{B}$ is the backward, $\mathbf{L}$ is the local and $\mathbf{F}_k$s are the set of forward level transition matrices (in our 3×3 case $k = 1, 2$).

In the next step we substitute (5.5) into (5.6), expand it, identify the terms corresponding to 0, 1, 2 and 3 cell arrivals to VOQ_{00} and we match its subexpressions to the terms of (5.7). All these are given in (5.8).

Going into details, one factor of the third powers in (5.6) is substituted by (5.5). Namely, in case of input 0 it is the first factor since it can send a cell to the observed central stage queue during the first time slot of the cycle (according to Table 5.1). In case of input 1(2), the third(second) factor is substituted. Once the substitution is done and the expansion is executed, the terms of the equation are collected based on “the number of $\mathbf{A}$s appearing in it”. This basically is equal to the number of cell arrivals to the observed VOQ.

After all manipulations we obtain (5.8) which is also indicating the meaning of all the terms. If there is only one cell served at the beginning of a period, this will result in a local level transition, since the queue is also served once in a cycle (e.g. one arrived and one served). In the similar manner all the other possible arrivals are assigned to the level transitions.

Finally it is possible to derive the irregular levels of the QBD-like process. In the first irregular level, when the central stage queue is empty, the DTMC can have $0, 1, 2$ and 3 level transitions according to $\mathbf{B}, \mathbf{L}, \mathbf{F}_1,$ and $\mathbf{F}_2$ respectively. In case of a full buffer the level process remains in the bth level instead of level transitions forward. According to this the forward level transition matrix in the level before the last one is $\mathbf{F}_1' = \mathbf{F}_1 + \mathbf{F}_2$ and the local state transition in the last level is $\mathbf{L}' = \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2$.

Then the state transition probability matrix of the QBD-like process on the block level is

$$\mathbf{P} = \begin{pmatrix} \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots & 0 \\ \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 \\ 0 & 0 & \dots & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1' \\ 0 & 0 & 0 & \dots & 0 & \mathbf{B} & \mathbf{L}' \end{pmatrix}. \tag{5.9}$$

Its steady state solution is the solution of the linear equation system

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi} \qquad\qquad \mathbf{P}\mathbf{h} = \mathbf{h}, \tag{5.10}$$

where $\mathbf{h}$ is the appropriate size column vector of ones.

5.2.3 The packet level model

We model the system on the packet level by a DTMC describing the life-cycle of a packet. For that reason we introduce two absorbing states indicating successful packet transmission and packet loss. A novel DTMC is based on the DTMC obtained from the previous section, while taking into account the absorbing states. Such a DTMC is described by its initial distribution and novel state transition probability matrix. In this section we present into details all these properties.

First, lets introduce the meaning of the absorbing states. Namely, the states correspond to two possible endings of the packet transmission: either the packet is lost or transmitted successfully. The new transient DTMC will describe the life cycle of the packet.

The Markov model of the system in Figure 5.3 consists of three main parts: first one is the revised QBD-like model of the virtual output queue and the other two are absorbing states appended to the QBD-like part. The absorbing state ST corresponds to the successful packet transmission in the observed path of the LB switch and CL to the first cell loss e.g. to the packet loss.

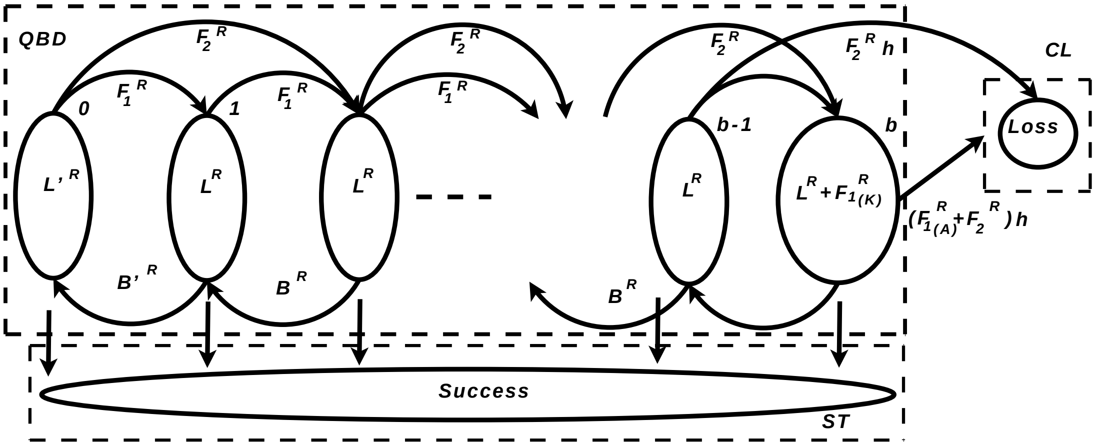


Figure 5.3: The transient DTMC for packet level model

$$\mathbf{P}_1^{\mathcal{R}} = \begin{pmatrix} (1-p_{10}) & 0 & 0 & 0 \\ p_{11}q_1t_{10} & (1-p_{11})+p_{11}q_1t_{11} & p_{11}q_1t_{12} & p_{11}\,(1-q_1) \\ p_{12}q_1t_{10} & p_{12}q_1t_{11} & (1-p_{12})+p_{12}q_1t_{12} & p_{12}\,(1-q_1) \\ q_1t_{10} & q_1t_{11} & q_1t_{12} & 1-q_1 \end{pmatrix}. \tag{5.11}$$

Modifications to the QBD-like process

In the following we describe the revised QBD-like model of VOQ_{00} when a packet is present in the system. The revision covers the determination of the state transition probabilities to the two absorbing states. First we remove state transition probabilities from $\mathbf{P}_1$ according to the successful packet transmission. Later on, these probabilities will be added as state transitions to the absorbing state ST in Section 5.2.3. In practice that means that the QBD-like model will be determined based on the revised DTMC description of input 1 given in Fig. 5.4 and in (5.11). Lets introduce the notation superscript $*^{\mathcal{R}}$ which will denote the properties of the transient DTMC introduced in this section.

The DTMCs of the other two inputs remain the same as in Figure 5.2 and in (5.4) since the observed path contains only input 1. The determination of the state transition probabilities to absorbing state CL will be given later in Section 5.2.3.

Similar to Section 5.2.2 we consider the switch operation during three consecutive time slots. Now we split $\mathbf{P}_1^{\mathcal{R}}$ into the two similar terms as in (5.5)

$$\mathbf{P}_1^{\mathcal{R}} = \mathbf{A}_1^{\mathcal{R}} + \mathbf{K}_1^{\mathcal{R}}. \tag{5.12}$$

Similar to the cell level QBD-like model we calculate the state transition probability matrix of the phase process in two different ways and do comparison between these two expressions in (5.13).

The first irregular level will be similar to that in (5.9). However the last level is composed by two cases. The first case is taken into account if there are no arrivals from

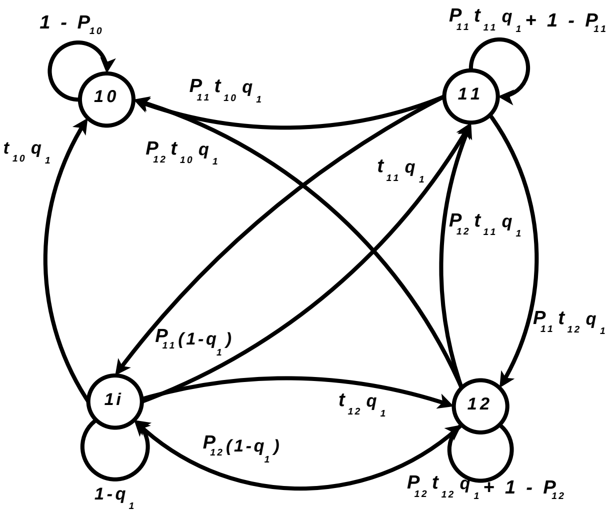


Figure 5.4: The revised DTMC model of input 1

$$
\begin{aligned}
\boldsymbol{\mathcal{P}} &= \mathbf{P}_0^3 \otimes \mathbf{P}_1^{\mathcal{R}^3} \otimes \mathbf{P}_2^3 = (\mathbf{A}_0 + \mathbf{K}_0)\,\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\Big(\mathbf{A}_1^{\mathcal{R}} + \mathbf{K}_1^{\mathcal{R}}\Big) \otimes \mathbf{P}_2\,(\mathbf{A}_2 + \mathbf{K}_2)\,\mathbf{P}_2 = \\
&= \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{K}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{K}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{1 arrival}} + \\
&+ \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{K}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{2 arrivals}} + \\
&+ \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{3 arrivals}} + \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{K}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{no arrivals}} = \mathbf{L}^{\mathcal{R}} + \mathbf{F}_1^{\mathcal{R}} + \mathbf{F}_2^{\mathcal{R}} + \mathbf{B}^{\mathcal{R}}
\end{aligned}
\tag{5.13}
$$

input 1 e.g. the system is remaining on the same level. Contrary to that a cell loss is admitted. According to these two cases $\mathbf{F}_1^{\mathcal{R}}$ is split into two terms

$$
\begin{aligned}
\mathbf{F}_1^{\mathcal{R}} = &\left(\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2\right) + \\
&+ \left(\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{K}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2\right) = \mathbf{F}_1^{\mathcal{R}(\mathcal{A})} + \mathbf{F}_1^{\mathcal{R}(\mathcal{K})}.
\end{aligned}
\tag{5.14}
$$

The first term stands for a cell arrival and the other for zero arrivals. Using this $\mathbf{L}^{\mathcal{R}'} =$

$\mathbf{L}^{\mathcal{R}} + \mathbf{F}_1^{\mathcal{R}(\mathcal{K})}$ and the state transition probability matrix of the QBD-like part we obtain

$$\hat{\mathbf{P}}^{\mathcal{R}} = \left(\begin{array}{ccccccc} \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} & 0 & \dots & 0 \\ \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} & 0 & \dots & 0 \\ \multicolumn{7}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ 0 & \dots & 0 & \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} \\ 0 & 0 & \dots & 0 & \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} \\ 0 & 0 & 0 & \dots & 0 & \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}\prime} \end{array} \right). \tag{5.15}$$

The packet loss

There can be cell loss (or equivalently packet loss) in the system in two cases

- either if the queue length is $b-1$ at the beginning of the cycle and there are three arrivals to VOQ_{00}
- or if the queue is full and there is arrival from input 1 to VOQ_{00}.

Appending the absorbing state CL to the QBD-like part and collecting the state transition probabilities to CL according to the two above cases we can build up the transpose of the state transition probability vector to CL as

$$\mathbf{l}^{\mathsf{T}} = \left(0 \quad \dots \quad 0 \quad \left(\mathbf{F}_2^{\mathcal{R}}\mathbf{h}\right)^{\mathsf{T}} \quad \left(\left(\mathbf{F}_1^{\mathcal{R}(\mathcal{A})} + \mathbf{F}_2^{\mathcal{R}}\right)\mathbf{h}\right)^{\mathsf{T}} \right), \tag{5.16}$$

where $\mathbf{h}$ is the appropriate size column vector of ones. Appending state CL to the QBD-like results in state transition probability matrix $\tilde{\mathbf{P}}^{\mathcal{R}} = \left(\begin{array}{c|c} \hat{\mathbf{P}}^{\mathcal{R}} & \mathbf{l} \\ \hline \multicolumn{2}{c}{\mathbf{e}_l^{\mathsf{T}}} \end{array} \right)$, where $\mathbf{e}_l^{\mathsf{T}} = \left(0 \dots 0 \; 1 \right)$ is the transpose of the last unit vector with the appropriate size.

The introduced analytical approach is also appropriate for the analysis of packet loss probabilities characterized by a different type e.g. different values of $\varDelta t$ (see (5.2)). The following set of modifications should be performed in this case. The first is the representation of the irregular levels of $\hat{\mathbf{P}}^{\mathcal{R}}$ in (5.15).

The second change is the modification of the state transition probability vector while transiting to CL ($\mathbf{l}$). It is represented as

- $\varDelta t = 0$ no state transitions to CL and
- $\varDelta t = 1$ state transitions to CL only from the last level.

The other differences for other types of packet loss on a path will be given in Section 5.2.1.

The cell loss

In order to calculate the cell loss through the path one should create $\mathbf{F}_1^{(\mathcal{A})}$ similarly to $\mathbf{F}_1^{\mathcal{R}(\mathcal{A})}$ by rearranging the term for $\mathbf{F}_1$ from (5.8) as

$$\mathbf{F}_1 = \Big(\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2\Big) + \\ + \Big(\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2\Big) = \mathbf{F}_1^{(\mathcal{A})} + \mathbf{F}_1^{(\mathcal{K})}.$$

It gives two terms, the first stands for the case when there is arrival from input 1 to VOQ_{00} and the second when there are no arrivals.

Having $\mathbf{F}_1^{(\mathcal{A})}$ and using $\mathbf{F}_2$ from (5.8) the cell loss is

$$p_c = \boldsymbol{\pi}_{b-1}\mathbf{F}_2\mathbf{h} + \boldsymbol{\pi}_b\left(\mathbf{F}_1^{(\mathcal{A})} + \mathbf{F}_2\right)\mathbf{h}, \tag{5.17}$$

where b is the buffer size and $\boldsymbol{\pi}_l$ is the $l+1$st sub-vector – with length $N+1$ – of $\boldsymbol{\pi}$ given in (5.10) and $\mathbf{h}$ is the appropriate size column vector of ones.

The successful packet transmission

The DTMC absorbs in state ST if the last cell of a packet (as well as all the others) is transmitted successfully through a VOQ. The preceding parts of this model (the DTMC) do not contain the state transitions responsible for packet ending (see Fig. 5.4 and (5.11)).In this section we take into account the vector containing the probabilities in order to change a state to ST, it is calculated as

$$\mathbf{s} = \mathbf{h} - \tilde{\mathbf{P}}^{\mathcal{R}}\mathbf{h}. \tag{5.18}$$

While appending state ST to the DTMC we get

$$\mathbf{P}^{\mathcal{R}} = \left(\begin{array}{c|c|c} \hat{\mathbf{P}}^{\mathcal{R}} & \mathbf{l} & \mathbf{s} \\ \hline \multicolumn{2}{c|}{\mathbf{e}_l^{\mathsf{T}}} & 0 \\ \hline \multicolumn{3}{c}{\mathbf{e}_l'^{\mathsf{T}}} \end{array}\right) \tag{5.19}$$

since the state transition probability matrix of the DTMC contains two absorbing states (appearing in Fig. 5.3). Finally, $\mathbf{e}_l'^{\mathsf{T}} = (\,0\ \ldots\ 0\ 1\,)$ is the transpose of the appropriate size vector.

5.2.4 The probabilities of packet loss and successful packet transmission

The packet loss probability is given as the probability of absorbing in state CL

$$p_l = \boldsymbol{\pi}^{\mathcal{N}}\left(\mathbf{I} - \hat{\mathbf{P}}^{\mathcal{R}}\right)^{-1}\mathbf{l} \tag{5.20}$$

and the probability of successful packet transmission is the probability of absorbing in state ST

$$p_s = \boldsymbol{\pi}^{\mathcal{N}} \left(\mathbf{I} - \hat{\mathbf{P}}^{\mathcal{R}}\right)^{-1} \mathbf{s}, \tag{5.21}$$

where $\mathbf{I}$ is the appropriate size identity matrix. $\boldsymbol{\pi}^{\mathcal{N}}$ is the initial probability distribution of the system immediately after a new packet arrival from input 1.

The initial distribution of the system

The system is considered to be in the steady state when a new packet arrives. Then the initial distribution $\boldsymbol{\pi}^{\mathcal{N}}$ of the system is given as the probabilities being in each state right after a new packet arrival. The quantities with superscript $*^{\mathcal{N}}$ are describing the system in this state.

In the $4 \le i \le b-1$st regular levels the states are described like

$$\hat{\boldsymbol{\pi}}_i^{\mathcal{N}} = \boldsymbol{\pi}_{i-2}\mathbf{F}_2^{\mathcal{N}} + \boldsymbol{\pi}_{i-1}\mathbf{F}_1^{\mathcal{N}} + \boldsymbol{\pi}_i\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_{i+1}\mathbf{B}^{\mathcal{N}} \tag{5.22}$$

and the irregular levels are the following

$$\begin{aligned}
\hat{\boldsymbol{\pi}}_0^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{B}^{\mathcal{N}} + \boldsymbol{\pi}_1\mathbf{B}^{\mathcal{N}} && (5.23)\\
\hat{\boldsymbol{\pi}}_1^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_1\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_2\mathbf{B}^{\mathcal{N}} && (5.24)\\
\hat{\boldsymbol{\pi}}_2^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{F}_1^{\mathcal{N}} + \boldsymbol{\pi}_1\mathbf{F}_1^{\mathcal{N}} + \boldsymbol{\pi}_2\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_3\mathbf{B}^{\mathcal{N}} && (5.25)\\
\hat{\boldsymbol{\pi}}_3^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{F}_2^{\mathcal{N}} + \boldsymbol{\pi}_1\mathbf{F}_2^{\mathcal{N}} + \boldsymbol{\pi}_2\mathbf{F}_1^{\mathcal{N}} + \boldsymbol{\pi}_3\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_4\mathbf{B}^{\mathcal{N}} && (5.26)\\
\hat{\boldsymbol{\pi}}_b^{\mathcal{N}} &= \boldsymbol{\pi}_{b-2}\mathbf{F}_2^{\mathcal{N}} + \boldsymbol{\pi}_{b-1}\left(\mathbf{F}_1^{\mathcal{N}} + \mathbf{F}_2^{\mathcal{N}}\right) + \\
&\quad + \boldsymbol{\pi}_b\left(\mathbf{L}^{\mathcal{N}} + \mathbf{F}_1^{\mathcal{N}} + \mathbf{F}_2^{\mathcal{N}}\right), && (5.27)
\end{aligned}$$

where $\boldsymbol{\pi}$ is the steady state solution of the cell level model given in (5.10). $\mathbf{B}^{\mathcal{N}}$, $\mathbf{L}^{\mathcal{N}}$ and $\mathbf{F}_i^{\mathcal{N}}$ are the level transition matrices of a QBD-like model describing the system right after a new packet arrival.

This QBD-like model is built in a similar way to that in Section 5.2.3. The the model of input 1 is containing only state transitions corresponding to a new packet arrival as it is shown in Fig. 5.5 and given as

$$\mathbf{P}_1^{\mathcal{N}} = \begin{pmatrix} p_{10}t_{10}q_1 & 0 & 0 & 0 \\ p_{11}q_1t_{10} & 0 & 0 & 0 \\ p_{12}q_1t_{10} & 0 & 0 & 0 \\ q_1t_{10} & 0 & 0 & 0 \end{pmatrix}. \tag{5.28}$$

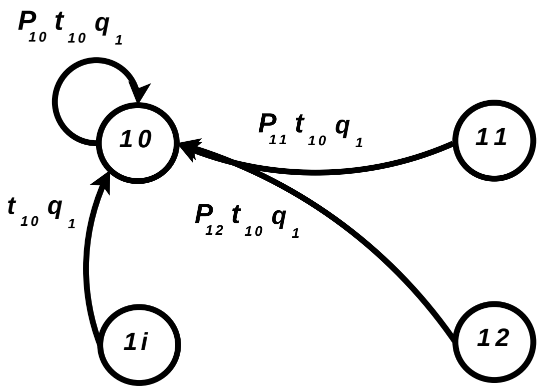


Figure 5.5: The DTMC of input 1; a new packet arrival

Table 5.2: The possible time evolution of input 1 with packet arrival

t mod 3 =			three consecutive time slots
0	1	2	
$+$	$+$	$+$	$\mathbf{P}_1^2\mathbf{P}_1^{\mathcal{N}}$
$-$	$+$	$+$	
$+$	$-$	$+$	
$-$	$-$	$+$	
$+$	$+$	$-$	$\Big(\mathbf{P}_1\mathbf{P}_1^{\mathcal{N}} + \mathbf{P}_1^{\mathcal{N}}\Big(\mathbf{P}_1 - \mathbf{P}_1^{\mathcal{N}}\Big)\Big)\Big(\mathbf{P}_1 - \mathbf{P}_1^{\mathcal{N}}\Big)$
$-$	$+$	$-$	
$+$	$-$	$-$	

The models of input 0 and 2 remain the same and are shown in Fig. 5.2.

$\mathbf{B}^{\mathcal{N}}$, $\mathbf{L}^{\mathcal{N}}$ and $\mathbf{F}_k^{\mathcal{N}}$ $\quad k = 1, 2$ are determined similar to the preceding cases. However, the behavior of input 1 needs some more considerations according to the new packet arrival.

Since the packet can arrive during each of all three time slots the state transition probability matrix of input 1 in a period is reconsidered based on Table 5.2. Its notations are:

- $+$ denotes the arrival of a packet in a time slot,
- the arrival instance(s) in a period positioned on the left hand side,
- the corresponding state transition probability matrix positioned on the right hand side.

We split $\mathbf{P}_1^{\mathcal{N}}$ into two parts

$$\mathbf{P}_1^{\mathcal{N}} = \mathbf{A}_1^{\mathcal{N}} + \mathbf{K}_1^{\mathcal{N}}, \tag{5.29}$$

containing two terms. First stands for the cell arrival and the second for zero arrival.

Based on Table 5.2, using (5.29) and (5.5) for $i = 1$, we give the matrices describing input 1 in a period if there is arrival – $\boldsymbol{\mathcal{A}}_1^{\mathcal{N}}$ – and if there are no arrivals – $\boldsymbol{\mathcal{K}}_1^{\mathcal{N}}$.

$$
\begin{aligned}
\boldsymbol{\mathcal{A}}_1^{\mathcal{N}} &= \mathbf{P}_1^2\mathbf{A}_1^{\mathcal{N}} + \Big(\mathbf{P}_1\mathbf{P}_1^{\mathcal{N}} + \mathbf{P}_1^{\mathcal{N}}\big(\mathbf{P}_1 - \mathbf{P}_1^{\mathcal{N}}\big)\Big)\Big(\mathbf{A}_1 - \mathbf{A}_1^{\mathcal{N}}\Big)\\
\boldsymbol{\mathcal{K}}_1^{\mathcal{N}} &= \mathbf{P}_1^2\mathbf{K}_1^{\mathcal{N}} + \Big(\mathbf{P}_1\mathbf{P}_1^{\mathcal{N}} + \mathbf{P}_1^{\mathcal{N}}\big(\mathbf{P}_1 - \mathbf{P}_1^{\mathcal{N}}\big)\Big)\Big(\mathbf{K}_1 - \mathbf{K}_1^{\mathcal{N}}\Big)
\end{aligned}
\tag{5.30}
$$

$$
\begin{aligned}
\boldsymbol{\mathcal{P}} = \mathbf{P}_0^3 \otimes \Big(\boldsymbol{\mathcal{A}}_1^{\mathcal{N}} + \boldsymbol{\mathcal{K}}_1^{\mathcal{N}}\Big) \otimes \mathbf{P}_2^3 = (\mathbf{A}_0 + \mathbf{K}_0)\,\mathbf{P}_0^2 \otimes \Big(\boldsymbol{\mathcal{A}}_1^{\mathcal{N}} + \boldsymbol{\mathcal{K}}_1^{\mathcal{N}}\Big) \otimes \mathbf{P}_2\,(\mathbf{A}_2 + \mathbf{K}_2)\,\mathbf{P}_2 = \\
= \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{1 arrival}} + \\
+ \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{2 arrivals}} + \\
+ \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{3 arrivals}} + \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{no arrivals}} = \mathbf{L}^{\mathcal{N}} + \mathbf{F}_1^{\mathcal{N}} + \mathbf{F}_2^{\mathcal{N}} + \mathbf{B}^{\mathcal{N}}
\end{aligned}
\tag{5.31}
$$

Now the state transition probability matrix of the phase process can be expressed in two different ways using (5.30) and (5.5) (for $i = 0, 2$). The comparison of these terms is resulting in the level transition matrices of this QBD-like process. It is given in (5.31).

Similarly to (5.8), (5.31) makes mapping with $\mathbf{B}^{\mathcal{N}}$, $\mathbf{L}^{\mathcal{N}}$, $\mathbf{F}_1^{\mathcal{N}}$ and $\mathbf{F}_2^{\mathcal{N}}$ matrices. Substituting them into the expressions (5.22) through (5.27) and normalizing them by $c = \sum_{i=0}^{b} \hat{\boldsymbol{\pi}}_i^{\mathcal{N}}\mathbf{h}$ we obtain

$$
\boldsymbol{\pi}_i^{\mathcal{N}} = \frac{1}{c}\hat{\boldsymbol{\pi}}_i^{\mathcal{N}}, \quad 0 \le i \le b, \tag{5.32}
$$

the individual parts of the initial distribution of the transient DTMC in Fig. 5.3 are modeling the system on the packet level. Combining the parts together we get $\boldsymbol{\pi}^{\mathcal{N}}$ which is the initial distribution.

Finally, substituting (5.15), (5.16), (5.18) and $\boldsymbol{\pi}^{\mathcal{N}}$ into (5.20) and (5.21) we get the packet loss probability and the probability of successful packet transmission.

The model presented in this Section describes a path with the highest loss probability according to Section 5.2.1.

5.2.5 Analysis of $N \times N$ load-balancing switch

The analysis of the $N \times N$ switch can be done in an analogous way to the 3×3 case. Therefore, we present the basic steps to create the model for packet loss evaluation.

Step1. Based on the chosen path create the model of the switch in N time slots long time period for the *cell level* analogously as it is described in the Section 5.2.2 for 3×3 case;

Step2. build up the transient DTMC describing the system for the *packet level,* similarly to the procedure described in Section 5.2.3;

Step3. based on the considered path of the cells determine the possible way of *cell/packet loss*, similarly to the derivations in Section 5.2.4 and determine the *initial probability* of the transient DTMC, as it is done in Section 5.2.4; and

Step4. solve the transient DTMC.

The above steps give the outline of the algorithm and based on Section 5.2 all the steps are well defined for its detailed program-automated implementation.

Unfortunately, even after proper description of the various steps of the algorithm in the general case (for arbitrary N) the state space increases exponentially with the size of the switch (and results in $O(N^N)$ complexity). This can lead to insolvable DTMCs even with the usage of the various sophisticated tools and numerical methods. In order to reduce the complexity of our model we present the approximated model for packet loss evaluation in the next section.

5.3 The O(2^N) Complexity Packet Loss Analysis

The followign section presents the approximated model of a VOQ packet loss probability analysis inside the Lb switch. Similarly to Section 5.2 we introduce all the important properties of the novel model on the example of the 3×3 LB switch. Compared to the exact analysis in [4] the approximation is related to the way in which we model the input processes, i.e., the arrival process to the considered VOQ. In particular, each input process is described using two state space ON/OFF model. With this assumption, the state space of the model can be significantly reduced compared to the exact model of [4]. Indeed the ON/OFF based model of the LB switch differs from the complete characterization in the DTMCs describing the input processes.

5.3.1 Input model

In this section we will introduce the approximate – ON/OFF – input model of path $\{1, 0, 0\}$ of the 3×3 switch.

The ON/OFF model of the first input is derived from its complete characterization depicted in Figure 5.6 using the notations introduced for the input processes in Section 5.1. According to the geometric assumptions for the packet length and idle period length this is a DTMC having four states, 1 *id* corresponds to the idle period, and the other three states corresponds to packet arrival from input 1 to either output 0 (state 10) or output 1 (state 11) or output 2 (state 12). The exact state transition probability matrix describing the behavior of input 1 is

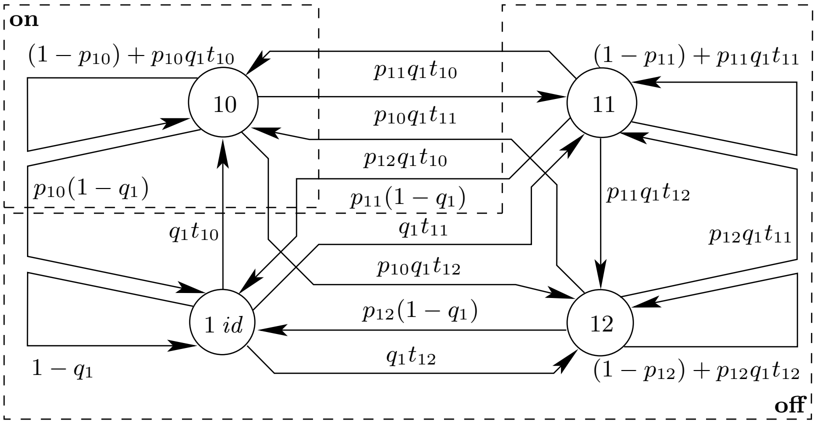


Figure 5.6: The DTMC which fully characterizes input 1 of 3 × 3 LB switch

$$\mathbf{P}_1^{\mathcal{C}} = \begin{pmatrix} (1-p_{10})+p_{10}q_1t_{10} & p_{10}q_1t_{11} & p_{10}q_1t_{12} & p_{10}\,(1-q_1) \\ p_{11}q_1t_{10} & (1-p_{11})+p_{11}q_1t_{11} & p_{11}q_1t_{12} & p_{11}\,(1-q_1) \\ p_{12}q_1t_{10} & p_{12}q_1t_{11} & (1-p_{12})+p_{12}q_1t_{12} & p_{12}\,(1-q_1) \\ q_1t_{10} & q_1t_{11} & q_1t_{12} & 1-q_1 \end{pmatrix}. \tag{5.33}$$

In terms of path $\{1,0,0\}$ the states of the DTMC modeling input 1 can be divided into two subsets

on this is a one-element subset containing state 10 in which there are cell arrivals from input 1 to output 0 and

off the other states in which there is no arrival from input 1 to output 0

which is also indicated in Figure 5.6. Using this division we create the two state ON/OFF model of the input processes. Hereinafter lowercase bold **on** and **off** denotes these two subsets and uppercase ON and OFF the two states of the newly derived DTMC model of the inputs.

OFF properties

The OFF state is used to approximate the set of **off** states. Its properties are determined based on the absorbing time of a *discrete phase type distribution* given in Figure 5.7 with transient states identical to the **off** states and absorbing state given as the **on** state. Its initial distribution then given as the renormalization of the zeroth row of $\mathbf{P}_1^{\mathcal{C}}$ in (5.33) without its zeroth element

$$\boldsymbol{\beta}_1 = \begin{pmatrix} \frac{q_1t_{11}}{q_1t_{11}+q_1t_{12}+(1-q_1)} & \frac{q_1t_{12}}{q_1t_{11}+q_1t_{12}+(1-q_1)} & \frac{1-q_1}{q_1t_{11}+q_1t_{12}+(1-q_1)} \end{pmatrix}. \tag{5.34}$$

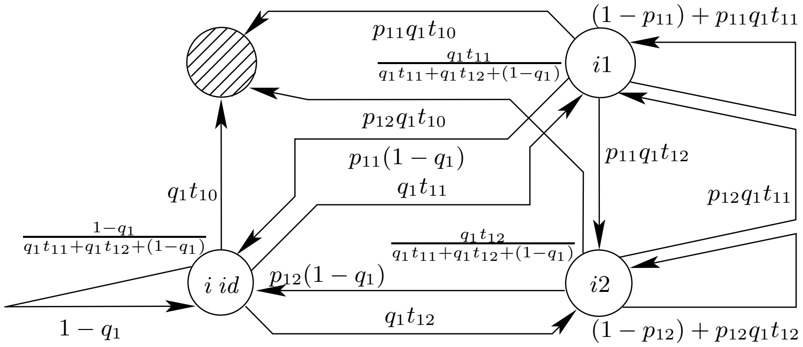

$(1-p_{11})+p_{11}q_1t_{11}$
$p_{11}q_1t_{10}$
$\frac{q_1t_{11}}{q_1t_{11}+q_1t_{12}+(1-q_1)}$
$i1$
$p_{12}q_1t_{10}$
$p_{11}(1-q_1)$
$p_{11}q_1t_{12}$
q_1t_{10}
q_1t_{11}
$p_{12}q_1t_{11}$
$\frac{1-q_1}{q_1t_{11}+q_1t_{12}+(1-q_1)}$
$\frac{q_1t_{12}}{q_1t_{11}+q_1t_{12}+(1-q_1)}$
$i\ id$
$p_{12}(1-q_1)$
$i2$
$1-q_1$
q_1t_{12}
$(1-p_{12})+p_{12}q_1t_{12}$

Figure 5.7: The DPH substitution of **off** states for a pair with input 1 - output 0

$\mathbf{B}_1$, the transition probability matrix of the transient states, is the $N \times N$ matrix given as $\mathbf{P}_1^{\mathcal{C}}$ without its zeroth row and zeroth column

$$\mathbf{B}_1 = \begin{pmatrix} (1-p_{11})+p_{11}q_1t_{11} & p_{11}q_1t_{12} & p_{11}\left(1-q_1\right) \\ p_{12}q_1t_{11} & (1-p_{12})+p_{12}q_1t_{12} & p_{12}\left(1-q_1\right) \\ q_1t_{11} & q_1t_{12} & 1-q_1 \end{pmatrix}. \qquad (5.35)$$

The mean absorbing time of this DPH is

$$\mu_1 = \boldsymbol{\beta}_1 \left(\mathbf{I} - \mathbf{B}_1\right)^{-1} \mathbf{h}, \qquad (5.36)$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{h}$ is the column vector of ones of appropriate size.

We set the sojourn probability of the state OFF to $1 - \frac{1}{\mu_1}$ which sets the mean sojourn time to μ_1. Then the state transition probability from OFF to ON is $\frac{1}{\mu_1}$.

ON properties

In case of ON the sojourn probability remain the same as in the complete characterization, i.e. in case of output 0 the upper left element of $\mathbf{P}_1^{\mathcal{C}}$ in (5.33). The state transition probability from ON to OFF is the summation of the remaining elements of the zeroth row of $\mathbf{P}_1^{\mathcal{C}}$ which is 1 minus the sojourn probability.

Summation of the ON/OFF DTMC

Here we summarize all the properties of the ON/OFF DTMC by giving its graph for the general path $\{i, j, k\}$ in Figure 5.8 together with its state transition probability matrix

$$\mathbf{P}_i = \begin{pmatrix} \left(\mathbf{P}_i^{\mathcal{C}}\right)_{jj} & 1 - \left(\mathbf{P}_i^{\mathcal{C}}\right)_{jj} \\ \frac{1}{\mu_i} & 1 - \frac{1}{\mu_i} \end{pmatrix} = \begin{pmatrix} (1-p_{ij})+p_{ij}q_it_{ij} & p_{ij}-p_{ij}q_it_{ij} \\ \frac{1}{\mu_i} & 1-\frac{1}{\mu_i} \end{pmatrix}, \qquad (5.37)$$

where $\left(*\right)_{ij}$ denotes the ijth element of a matrix.

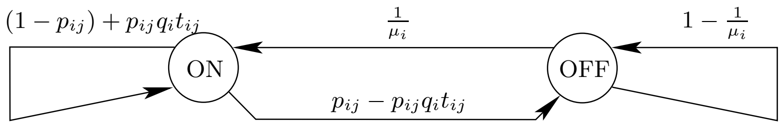


Figure 5.8: The ON/OFF DTMC describing a pair with input i - output j

5.3.2 The cell level model

Up to now we have introduced the differences between the full model of [4] and the ON/OFF model of the input processes. From now on we recall the remaining part of building the model of the VOQ using the ON/OFF model of each input. Here we keep on with building the model of the VOQ of path $\{1,0,0\}$.

First of all we give the cell level model of VOQ_{00} which is a quasi birth-deathlike (QBD-like) DTMC where the level represents the queue length and the phase is the combined state $\left(0,1,\ldots,2^N-1\right)$ of the inputs.

According to the periodic operation of the switch mentioned in Section 5.1 the time unit of the QBD-like model is N time slots – the time period of the operation of the switch.

The DTMC is given in Figure 5.8 and together with (5.37) gives the behavior of the input process in a single time slot. As the next step we raise all of these DTMCs to the Nth $=$ 3rd power to have the model of the input processes in a time period.

Then the joint behavior of the input processes – for all inputs $(i=0,1,2)$ – gives the phase process of the QBD-like model which is the following Kronecker product:

$$\boldsymbol{\mathcal{P}}=\mathbf{P}_0^3\otimes\mathbf{P}_1^3\otimes\mathbf{P}_2^3. \tag{5.38}$$

The number of arrivals to the observed VOQ is determined as the sum of the arrivals from each input, but each input can also transmit a cell into the VOQ in its dedicated time slot. The interconnection process is determined by the pattern given in (5.1), i.e. input 0 sends cell to VOQ_{00} during the 1st time slot of a time period, input 1 sends during the 3rd time slot of a period and, finally, input 2 sends it during the 2nd time slot of a time period. Please note that the ordinal number of the dedicated time slot is equal to $\Delta t+1$ for each input i in any path. According to this we replace the 1st, the 3rd and the 2nd factor of the powers of $\mathbf{P}_0^3$, $\mathbf{P}_1^3$ and $\mathbf{P}_2^3$ respectively in (5.38) to

$$\mathbf{P}_i=\mathbf{A}_i+\mathbf{K}_i\quad\forall i\in[0,N-1], \tag{5.39}$$

in which the first term corresponds to arrival from input i and the second term corresponds to the case when there is no arrival from input i. The substitution is then

$$\boldsymbol{\mathcal{P}}=\mathbf{P}_0^3\otimes\mathbf{P}_1^3\otimes\mathbf{P}_2^3=\left(\mathbf{A}_0+\mathbf{K}_0\right)\mathbf{P}_0^2\otimes\mathbf{P}_1^2\left(\mathbf{A}_1+\mathbf{K}_1\right)\otimes\mathbf{P}_2\left(\mathbf{A}_2+\mathbf{K}_2\right)\mathbf{P}_2 \tag{5.40}$$

based on the Δt values of the inputs calculated as given in (5.2). Expanding this expression and collecting the terms according to $0, 1, 2$ and 3 arrivals we get

$$\begin{aligned}\boldsymbol{\mathcal{P}} = {} & \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{no arrivals} - \mathbf{B}} + \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{1 arrival} - \mathbf{L}} + \\ & + \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{1 arrival} - \mathbf{L}} + \\ & + \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{K}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{2 arrivals} - \mathbf{F}_1} + \\ & + \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{2 arrivals} - \mathbf{F}_1} + \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^2\mathbf{A}_1 \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{3 arrivals} - \mathbf{F}_2} = \\ & = \mathbf{B} + \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2,\end{aligned} \tag{5.41}$$

where we have also indicated the level transition with the decomposition, $\boldsymbol{\mathcal{P}} = \mathbf{B} + \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2$, of such a QBD-like model.

Correspondingly, the state transition probability matrix has the following QBD-like structure

$$\mathbf{P} = \begin{pmatrix} \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots \\ \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 \\ \dots & 0 & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1' \\ \dots & 0 & 0 & 0 & \mathbf{B} & \mathbf{L}' \end{pmatrix}, \tag{5.42}$$

where $\mathbf{F}_1' = \mathbf{F}_1 + \mathbf{F}_2$ and $\mathbf{L}' = \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2$.

Creation process of this kind of QBD-like DTMCs for $N = 3$ is given in Algorithm 1.

Algorithm 1 Building the QBD-like model of a VOQ

```
Require: $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2$ from (5.37)
Ensure: $\mathbf{P}$ the QBD-like model similar to (5.42)
for $i = 0$ to 2 do
    compute $\mathbf{A}_i, \mathbf{K}_i$ as given in (5.39)
    calculate $\Delta t$ for the $i$th input as given in (5.2)
    replace the $(\Delta t + 1)$st factor of $\mathbf{P}_i^3$ in (5.38) with $\mathbf{A}_i + \mathbf{K}_i$ as given in (5.40)
end for
expand the resulting expression for $\boldsymbol{\mathcal{P}}$ and
identify the level transition matrices $\mathbf{B}, \mathbf{L}, \mathbf{F}_1, \mathbf{F}_2$ as given in (5.41)
build $\mathbf{P}$ as in (5.42)
return $\mathbf{P}$
```

The steady state solution of this QBD-like model is the solution of the linear equation

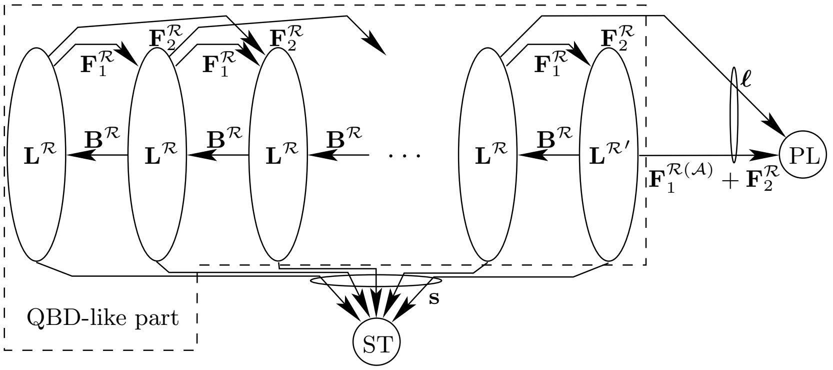

$\mathbf{F}_2^{\mathcal{R}}$
$\mathbf{F}_1^{\mathcal{R}}$
$\mathbf{L}^{\mathcal{R}}$
$\mathbf{B}^{\mathcal{R}}$
$\mathbf{L}^{\mathcal{R}\prime}$
$\cdots$
$\boldsymbol{\ell}$
PL
$\mathbf{F}_1^{\mathcal{R}(\mathcal{A})} + \mathbf{F}_2^{\mathcal{R}}$
$\mathbf{s}$
ST
QBD-like part

Figure 5.9: The transient DTMC which characterizes a VOQ during a life cycle of a packet

system

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}, \qquad\qquad \boldsymbol{\pi}\mathbf{h} = 1. \tag{5.43}$$

5.3.3 The packet level model

With the geometric assumption for the packet length, given in Section 5.1, the life cycle of a packet in the observed path can be modeled by a transient DTMC in which the two absorbing states corresponds to the two possible ending of a packet transmission – the successful transmission (ST) of the packet or its lost (PL), as given in Figure 5.9. In this section we present this transient DTMC with its state transition probability matrix and initial distribution.

The state transition probability matrix of the transient part

The transient DTMC is mainly built in the same way as the QBD-like model of the VOQ on the cell level in Section 5.3.2. The exceptions are

- the state transitions responsible for packet completion in the observed path are removed (its DTMC is given in Figure 5.10(a)) and
- the cell losses in case of “nearly” full buffer are considered.

The removal of the state transitions is explained while introducing the absorbing state ST. Indeed, the transient DTMC moves to state ST when the transmission of a packet is completed. Then, according to these modifications the state transition probability matrix of the modified DTMC of input 1, with such state transitions removed (see Figure 5.10(a)), is

$$\mathbf{P}_1^{\mathcal{R}} = \begin{pmatrix} 1 - p_{10} & 0 \\ 1 - \frac{1}{\mu_i} & \frac{1}{\mu_i} \end{pmatrix}, \tag{5.44}$$

where superscript $\mathcal{R}$ refers to the DTMC with absorbing states PL and ST, in Figure 5.9. The DTMC of the other two inputs remains as in (5.37).

The state transition probability matrix of the QBD-like part of the DTMC in Figure 5.9 is $\mathbf{P}^{\mathcal{R}}$. It is determined by Algorithm 1 with input parameters $\mathbf{P}_0, \mathbf{P}_1^{\mathcal{R}}, \mathbf{P}_2$ but with an exception in line 8.

Having evaluated the level transition matrices $\left(\mathbf{B}^{\mathcal{R}}, \mathbf{L}^{\mathcal{R}}, \mathbf{F}_1^{\mathcal{R}}, \mathbf{F}_2^{\mathcal{R}}\right)$ the structure of the QBD-like part and state transition vector (to state PL) is the following

$$\mathbf{P}^{\mathcal{R}} = \begin{pmatrix} \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} & 0 & \dots \\ \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} \\ \dots & 0 & 0 & \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} \\ \dots & 0 & 0 & 0 & \mathbf{B}^{\mathcal{R}} & \mathbf{L}^{\mathcal{R}\prime} \end{pmatrix}, \qquad \boldsymbol{\ell} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{F}_2^{\mathcal{R}}\mathbf{h} \\ \left(\mathbf{F}_1^{\mathcal{R}(\mathcal{A})} + \mathbf{F}_2^{\mathcal{R}}\right)\mathbf{h} \end{pmatrix}, \tag{5.45}$$

where $\mathbf{L}^{\mathcal{R}\prime} = \mathbf{L}^{\mathcal{R}} + \mathbf{F}_1^{\mathcal{R}(\mathcal{K})}$. Here, forward level transition matrix $\left(\mathbf{F}_1^{\mathcal{R}}\right)$ is decomposed into two terms and both of them are corresponding to two cells arrivals. In the first case one of the cells arrives from input 1

$$\mathbf{F}_1^{\mathcal{R}(\mathcal{A})} = \mathbf{K}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{A}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2$$

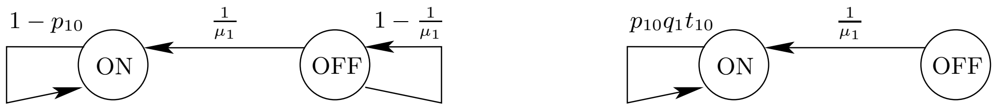
and in the second case none of them is arriving from input 1

$$\mathbf{F}_1^{\mathcal{R}(\mathcal{K})} = \mathbf{A}_0\mathbf{P}_0^2 \otimes \mathbf{P}_1^{\mathcal{R}^2}\mathbf{K}_1^{\mathcal{R}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2.$$

According to this observations, the cell loss and accordingly packet loss is possible in the observed path $\{1, 0, 0\}$ if at the beginning of a time period either

- one free position in VOQ$_{00}$ is available and there are arrivals from all three inputs $\left(\mathbf{F}_2^{\mathcal{R}}\right)$ or
- the buffer is full and there are cell arrivals either
  - from all the three inputs $\left(\mathbf{F}_2^{\mathcal{R}}\right)$ or
  - there are two arrivals from input 1 $\left(\mathbf{F}_1^{\mathcal{R}(\mathcal{A})}\right)$.

Accordingly, if the buffer tends to be full at the beginning of a time period and there are two new arrivals (however none of them is from input 1), the DTMC will remain in the last level $\left(\mathbf{F}_1^{\mathcal{R}(\mathcal{K})}\right)$.



(a) Packet completion

(b) New packet arrival

Figure 5.10: The modified graphs of the ON/OFF DTMC describing input 1

Finally according to Figure 5.10(a) and (5.44) and using the notations of Figure 5.9 and (5.45) the state transition probability vector to the absorbing state ST is the following

$$\mathbf{s}=\mathbf{h}-\left(\mathbf{P}^{\mathcal{R}}\mathbf{h}-\boldsymbol{\ell}\right). \tag{5.46}$$

The initial distribution of the transient DTMC

The initial distribution of $\mathbf{P}^{\mathcal{R}}$ in (5.45) is determined as the state of the system right after an incoming customer arrival. In this section we determine the probability distribution of the system at a time instance when a new packet arrives.

A packet enters input 1 according to the state transitions depicted in Figure 5.10(b). Correspondingly, its state transition probability matrix is

$$\mathbf{P}_1^{\mathcal{N}}=\begin{pmatrix} p_{10}q_1t_{10} & 0 \\ \frac{1}{\mu_1} & 0 \end{pmatrix}, \tag{5.47}$$

where superscript $\mathcal{N}$ refers to the DTMC with new packet arrival.

Lets build a QBD-like model while reusing Algorithm 1 with input parameters $\mathbf{P}_0, \mathbf{P}_1^{\mathcal{N}}, \mathbf{P}_2$ and an exception in line 4 which also affects lines 6 and 7. Instead of replacing the third factor of $\mathbf{P}_1^{\mathcal{N}^3}$ (knowing that $\Delta t=2$ for $i=1$) we represent the state transition probability matrix of input 1 for a whole time period as

$$\mathbf{P}_1^3-\left(\mathbf{P}_1-\mathbf{P}_1^{\mathcal{N}}\right)^3. \tag{5.48}$$

It expresses the behavior of input 1 at an event of a new packet arrival during a three time slot period. According to Algorithm 1 we expand (5.48) while simplifying and replacing the third factors of all the terms as

$$\begin{aligned}\mathbf{P}_1^3-\left(\mathbf{P}_1-\mathbf{P}_1^{\mathcal{N}}\right)^3 &= \underbrace{\mathbf{P}_1^2\mathbf{A}_1^{\mathcal{N}}+\left(\mathbf{P}_1\mathbf{P}_1^{\mathcal{N}}+\mathbf{P}_1^{\mathcal{N}}\left(\mathbf{P}_1-\mathbf{P}_1^{\mathcal{N}}\right)\right)\left(\mathbf{A}_1-\mathbf{A}_1^{\mathcal{N}}\right)}_{\boldsymbol{\mathcal{A}}_1^{\mathcal{N}}}+ \\ &+\underbrace{\mathbf{P}_1^2\mathbf{K}_1^{\mathcal{N}}+\left(\mathbf{P}_1\mathbf{P}_1^{\mathcal{N}}+\mathbf{P}_1^{\mathcal{N}}\left(\mathbf{P}_1-\mathbf{P}_1^{\mathcal{N}}\right)\right)\left(\mathbf{K}_1-\mathbf{K}_1^{\mathcal{N}}\right)}_{\boldsymbol{\mathcal{K}}_1^{\mathcal{N}}}=\boldsymbol{\mathcal{A}}_1^{\mathcal{N}}+\boldsymbol{\mathcal{K}}_1^{\mathcal{N}},\end{aligned} \tag{5.49}$$

where we have also indicated two terms according to a cell arrival $\left(\boldsymbol{\mathcal{A}}_1^{\mathcal{N}}\right)$ into VOQ$_{00}$ and no cell arrivals $\left(\boldsymbol{\mathcal{K}}_1^{\mathcal{N}}\right)$ in a time period. These two matrices are used to replace the whole middle operand of (5.38) in line 6 of Algorithm 1 as

$$\begin{aligned}
\boldsymbol{\mathcal{P}}^{\mathcal{N}} &= (\mathbf{A}_0+\mathbf{K}_0)\,\mathbf{P}_0^2 \otimes \left(\boldsymbol{\mathcal{A}}_1^{\mathcal{N}}+\boldsymbol{\mathcal{K}}_1^{\mathcal{N}}\right) \otimes \mathbf{P}_2\,(\mathbf{A}_2+\mathbf{K}_2)\,\mathbf{P}_2 = \\
&= \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{no arrivals} - \mathbf{B}^{\mathcal{N}}} + \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{1 arrival} - \mathbf{L}^{\mathcal{N}}} + \\
&+ \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2 + \mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{1 arrival} - \mathbf{L}^{\mathcal{N}}} + \\
&+ \underbrace{\mathbf{K}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2 + \mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{K}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{2 arrivals} - \mathbf{F}_1^{\mathcal{N}}} + \\
&+ \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{K}_2\mathbf{P}_2}_{\text{2 arrivals} - \mathbf{F}_1^{\mathcal{N}}} + \underbrace{\mathbf{A}_0\mathbf{P}_0^2 \otimes \boldsymbol{\mathcal{A}}_1^{\mathcal{N}} \otimes \mathbf{P}_2\mathbf{A}_2\mathbf{P}_2}_{\text{3 arrivals} - \mathbf{F}_2^{\mathcal{N}}} = \\
&= \mathbf{B}^{\mathcal{N}} + \mathbf{L}^{\mathcal{N}} + \mathbf{F}_1^{\mathcal{N}} + \mathbf{F}_2^{\mathcal{N}}.
\end{aligned} \tag{5.50}$$

The level transition matrices used in line 8 of Algorithm 1 for state transition probability matrix $\left(\mathbf{P}^{\mathcal{N}}\right)$ construction are defined in the same way as in (5.42).

Using (5.43) and $\mathbf{P}^{\mathcal{N}}$ the initial distribution of the DTMC in Figure 5.9 is the following

$$\boldsymbol{\pi}^{\mathcal{N}} = \frac{\boldsymbol{\pi}\mathbf{P}^{\mathcal{N}}}{\boldsymbol{\pi}\mathbf{P}^{\mathcal{N}}\mathbf{h}}. \tag{5.51}$$

The packet loss of the system

Using (5.45), (5.46) and (5.51) the packet loss probability (p_ℓ) is defined as the probability of absorbing in state PL and the probability of successful packet transmission (p_s) is given as an absorption in state ST

$$p_\ell = \boldsymbol{\pi}^{\mathcal{N}}\left(\mathbf{I}-\mathbf{P}^{\mathcal{R}}\right)^{-1}\boldsymbol{\ell} \qquad\qquad p_s = \boldsymbol{\pi}^{\mathcal{N}}\left(\mathbf{I}-\mathbf{P}^{\mathcal{R}}\right)^{-1}\mathbf{s} = 1-p_\ell. \tag{5.52}$$

In this section, while introducing the diversity of the representation and analysis of input arrival process we have reduced the overall computational complexity of the algorithm from $O(N^N)$ to $O(2^N)$. Unfortunately the model is still not suitable for the packet loss probability evaluation when N is large. To cope with complex representation and construction of the model, we introduce a novel algorithm with the consideration of a homogeneous input traffic. This model will be represented in the next section.

5.4 The Linear Complexity Packet Loss Analysis

In the following model we assume that parameters of packet length and interpacket periods are the same for all inputs according to the identical input process assumption. This makes us possible to introduce a compact approximate model with the linear complexity for the LB switch. The packets arriving at an arbitrary input are spread uniformly between the outputs, i.e., the probability of sending a packet to a particular output is given as

$$\hat{t} = \frac{1}{N}. \tag{5.53}$$

According to the Markovian assumption the packet length (X) distribution (in cells) of the arrival process is geometric distributed with probability mass function (PMF)

$$\Pr\left(X = i\right) = \hat{p}\left(1 - \hat{p}\right)^{i-1} \quad i = 1, 2, \ldots \tag{5.54}$$

The length of the idle periods between packets (Y) are also geometric distributed (in time slots) with PMF

$$\Pr\left(Y = i\right) = \hat{q}\left(1 - \hat{q}\right)^{i} \quad i = 0, 1, \ldots \tag{5.55}$$

It is shown in Section 5.2 [4] and Section 5.3 [2] that the cell loss probability and accordingly the packet loss probability depend on the path through which it is evaluated. Similarly the sections mentioned above, for out representation we model the queue VOQ$_{00}$ as a part of path $\{1, 0, 0\}$ in the 3×3 LB switch.

5.4.1 The model of the input processes

The parameters of the identical input process are

$\hat{p}$ the parameter of the geometric distributed packet length (5.54) in cells,

$\hat{q}$ the parameter of the geometric distributed idle period length (5.55) in time slots and

$\hat{t} = \frac{1}{N}$ the probability of choosing a specific output for a given packet (5.53).

Based on the geometric assumption we can build the DTMC model, fully characterizing any of the identical inputs, with state transition probability matrix

$$\mathbf{P}^{C} = \begin{pmatrix} (1-\hat{p}) + \hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\left(1-\hat{q}\right) \\ \hat{p}\hat{q}\hat{t} & (1-\hat{p}) + \hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\left(1-\hat{q}\right) \\ \hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & (1-\hat{p}) + \hat{p}\hat{q}\hat{t} & \hat{p}\left(1-\hat{q}\right) \\ \hat{q}\hat{t} & \hat{q}\hat{t} & \hat{q}\hat{t} & 1-\hat{q} \end{pmatrix} \tag{5.56}$$

and the graph given in Figure 5.11, where the state identifiers are the following

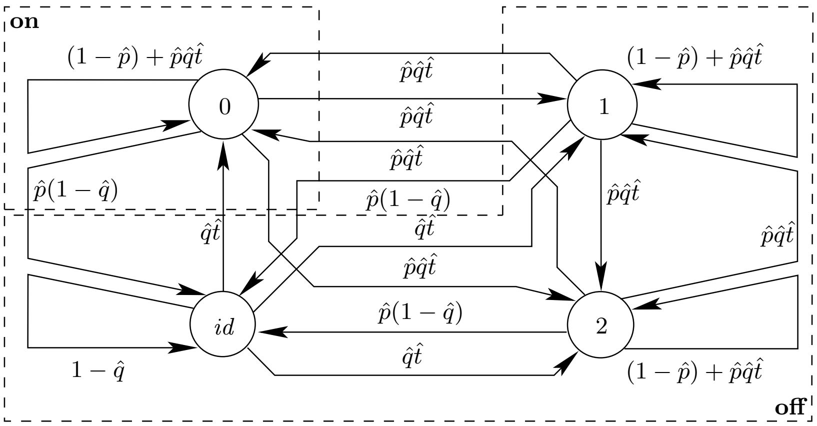


Figure 5.11: The DTMC which characterizes every input of 3 × 3 LB switch

j corresponds to cell arrival from the input to output $j \quad j = 0, 1, 2$

id corresponds to the idle period of the input.

According to the observed output, i.e., output 0, the states of the DTMC in Figure 5.11 are divided into two subsets, denoted as **on** and **off** type, respectively. Their meaning are

on is the state which represents a cell arrival from the observed input to output 0 and all

off are the states which assume no cell arrivals from the observed input to output 0.

In the following we introduce the approximating two state ON/OFF model of the general input mainly as replacing the set **off** with a single state OFF. Hereinafter uppercase ON and OFF denote the states of the approximating two state description of the input process.

The ON properties

State ON replaces the element of a subset (e.g. **on**) with sojourn probability $(1-\hat{p})+\hat{p}\hat{q}\hat{t}$. Accordingly, the state transition probability from ON to OFF states is represented as 1 minus the sojourn probability $\hat{p}-\hat{p}\hat{q}\hat{t}$.

The OFF properties

OFF state replaces the set of **off** states by approximating their sojourn time with the absorbing time of a DPH distribution. For output 0 the transient states of the DPH are the following (as depicted in Figure 5.12).

Based on $\mathbf{P}^{\mathcal{C}}$, given in (5.56), we denote the initial distribution ($\boldsymbol{\beta}$) and the state transition probability matrix ($\mathbf{B}$) of the DPH. The initial distribution is the state probability

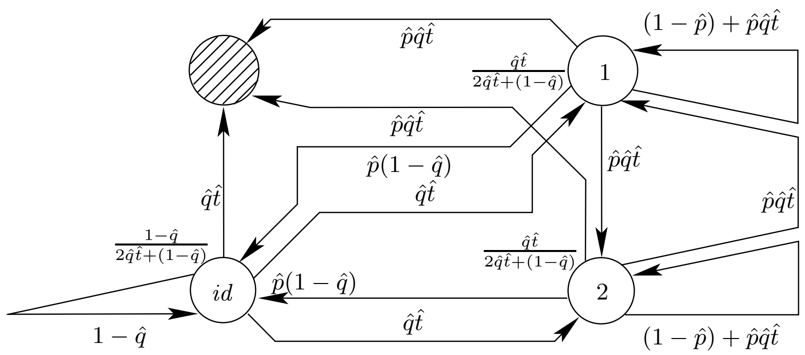

$\hat{p}\hat{q}\hat{t}$
$(1-\hat{p})+\hat{p}\hat{q}\hat{t}$
$\frac{\hat{q}\hat{t}}{2\hat{q}\hat{t}+(1-\hat{q})}$
1
$\hat{p}\hat{q}\hat{t}$
$\hat{p}\hat{q}\hat{t}$
$\hat{p}(1-\hat{q})$
$\hat{q}\hat{t}$
$\hat{q}\hat{t}$
$\hat{p}\hat{q}\hat{t}$
$\frac{1-\hat{q}}{2\hat{q}\hat{t}+(1-\hat{q})}$
$\frac{\hat{q}\hat{t}}{2\hat{q}\hat{t}+(1-\hat{q})}$
id
$\hat{p}(1-\hat{q})$
2
$1-\hat{q}$
$\hat{q}\hat{t}$
$(1-\hat{p})+\hat{p}\hat{q}\hat{t}$

Figure 5.12: The DPH substitution of **off** states in terms of output 0

right after entering state **off** from state **on**. It is obtained as the renormalization of the zeroth row of $\mathbf{P}^{\mathcal{C}}$ without its zeroth element

$$\boldsymbol{\beta} = \begin{pmatrix} \frac{\hat{q}\hat{t}}{2\hat{q}\hat{t}+(1-\hat{q})} & \frac{\hat{q}\hat{t}}{2\hat{q}\hat{t}+(1-\hat{q})} & \frac{1-\hat{q}}{2\hat{q}\hat{t}+(1-\hat{q})} \end{pmatrix},$$

which is also indicated in Figure 5.12. The 3×3 sized state transition probability matrix of all the **off** states is obtained from $\mathbf{P}^{\mathcal{C}}$ by cutting the zeroth row and the zeroth column

$$\mathbf{B} = \begin{pmatrix} (1-\hat{p})+\hat{p}\hat{q}\hat{t} & \hat{p}\hat{q}\hat{t} & \hat{p}\,(1-\hat{q}) \\ \hat{p}\hat{q}\hat{t} & (1-\hat{p})+\hat{p}\hat{q}\hat{t} & \hat{p}\,(1-\hat{q}) \\ \hat{q}\hat{t} & \hat{q}\hat{t} & 1-\hat{q} \end{pmatrix}.$$

The mean absorbing time of this DPH is then

$$\mu = \boldsymbol{\beta}\,(\mathbf{I}-\mathbf{B})^{-1}\,\mathbf{h}, \tag{5.57}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{h}$ is the column vector of ones of the appropriate size. According to the structure of (5.56) μ is the same for all the output-input pairs – indeed the input processes are identical. Consequently, the sojourn probability of state OFF is $1-\frac{1}{\mu}$. The state transition probability from OFF to ON is $\frac{1}{\mu}$ which sets the mean sojourn time in state OFF equals to μ. The state transition probability matrix of the two state DTMC describing the ON/OFF input process is

$$\mathbf{P} = \begin{pmatrix} (1-\hat{p})+\hat{p}\hat{q}\hat{t} & \hat{p}-\hat{p}\hat{q}\hat{t} \\ \frac{1}{\mu} & 1-\frac{1}{\mu} \end{pmatrix} = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}, \tag{5.58}$$

where we also introduced a simplified notation with p and q. The graph of the ON/OFF DTMC using the simplified notation is given in Figure 5.13 which is the same as in all the other inputs according to the identical input process assumption.

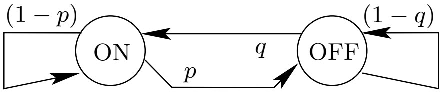


Figure 5.13: ON/OFF model of the input process with a simplified notation

5.4.2 Aggregate input model

We describe the combined behavior of the N inputs by a DTMC of $N+1$ states representing a number of inputs in ON. Using the considerations from Section 5.4.1 and especially (5.58), the ijth element of the state transition probability matrix of such a DTMC describing N inputs after 1 time slots is the following

$$\left(\boldsymbol{\mathcal{P}}_{N,1}(p,q)\right)_{ij} = \sum_{k=\max(0,j-i)}^{\min(i,N-j)} \binom{i}{k} p^k \left(1-p\right)^{i-k} \binom{N-i}{j-i+k} q^{j-i+k} \left(1-q\right)^{N-j-k}. \quad (5.59)$$

The presented probabilities are also depending on the parameters of (5.58) – e.g. p, q. The first binomial factor of (5.59) represents that out of i ON sources k moves to OFF and the second factor represents that out of $N-i$ OFF sources $j-i+k$ moves to ON, $i, j \in [0, N-1]$. (5.59) also introduces the notation $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ hereinafter denoting the state of N inputs during M time slots with each input modeled by an ON/OFF DTMC with parameters p and q given in (5.58). As an example, the state of N inputs after M time slots is

$$\boldsymbol{\mathcal{P}}_{N,M}(p,q) = \boldsymbol{\mathcal{P}}_{N,1}^{M}(p,q). \quad (5.60)$$

Using the above method one can describe the behavior of any number of inputs within any number of time slots.

Based on $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ we decompose the matrix of the arrival process as

$$\underbrace{\mathbf{B} = \begin{pmatrix} \mathbf{p}^0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{\text{0 arrivals}} \quad \underbrace{\mathbf{L} = \begin{pmatrix} 0 \\ \mathbf{p}^1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}}_{\text{1 arrival}} \quad \underbrace{\mathbf{F}_1 = \begin{pmatrix} 0 \\ 0 \\ \mathbf{p}^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{\text{2 arrivals}} \quad \ldots \quad \underbrace{\mathbf{F}_{N-1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \mathbf{p}^N \end{pmatrix}}_{\text{N arrivals}}, \quad (5.61)$$

where $\mathbf{p}^i$ denotes the ith row vector of $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$.

The arrival-based decomposition of the $N \times N$ switch in M time slots, is formalized in Algorithm 2.

Algorithm 2 Arrival based decomposition of the input process

```
Require: $N, M, \mathbf{P}$ from (5.58)
Ensure: $\mathbf{B}, \mathbf{L}, \mathbf{F}_1, \dots, \mathbf{F}_{N-1}$ the arrival based decomposition
1: determine $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ similar to (5.60) using $\mathbf{P}$
2: decompose $\boldsymbol{\mathcal{P}}_{N,M}(p,q)$ as in (5.61)
3: return $\mathbf{B}, \mathbf{L}, \mathbf{F}_1, \dots, \mathbf{F}_{N-1}$
```

5.4.3 The cell level model of the 3×3 switch

In order to calculate the packet loss on the path $\{1, 0, 0\}$, first we derive the model describing VOQ$_{00}$ behavior on the cell level. It is given as a quasi birth-death like (QBD-like) structure whose levels represent the current queue length and phases describe the current state of the input process.

Since the phase process of the QBD-like model is considered to be the combined state of all the inputs, while using the arrival-based decomposition we represent the level transition matrices used to build the QBD-like structure. $\mathbf{B}, \mathbf{L}, \mathbf{F}_1, \mathbf{F}_2$ are determined by Algorithm 2 with input parameters $N = 3$, according to the number of inputs, $M = 3$ the number of time slots in a time period and $\mathbf{P}$ (from (5.58)).

A level transition backward is described according to matrix $\mathbf{B}$ since there is only one cell is served during a time period. In the similar way the local and forward transitions are described ($\mathbf{L}$, $\mathbf{F}_1$, $\mathbf{F}_2$).

The state transition probability matrix of the QBD-like model is

$$\mathbb{P} = \begin{pmatrix} \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots \\ \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 \\ \dots & 0 & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}'_1 \\ \dots & 0 & 0 & 0 & \mathbf{B} & \mathbf{L}' \end{pmatrix}, \tag{5.62}$$

where $\mathbf{F}'_1 = \mathbf{F}_1 + \mathbf{F}_2$ and $\mathbf{L}' = \mathbf{L} + \mathbf{F}_1 + \mathbf{F}_2$.

The steady state solution of this QBD-like model is the solution of the linear system of equations

$$\boldsymbol{\pi}\mathbb{P} = \boldsymbol{\pi}, \qquad \boldsymbol{\pi}\mathbf{h} = 1. \tag{5.63}$$

5.4.4 The packet level model

With the geometric assumption for the packet length, given at the beginning of this chapter, and, using the notations of the previous chapters, the life cycle of a packet in the observed path can be modeled by a transient DTMC in which the two additional absorbing states should be added. The first state corresponds to the packet loss (PL) and the other corresponds to the successful packet transmission (ST). The transient DTMC with two absorbing states is given in Figure 5.9.

The transient part of the DTMC

Basically during the life cycle of a packet VOQ_{00} is modeled by a quasi birth like (QB-like) structure. Its level represents the queue length and its phase process is the combined state of all 3 inputs. In this case there is one important difference compared to the model given in the previous section. Input 1 is certainly stays in ON state, due to the path assumption, which implies that there is no backward level transition.

The other two inputs behave in the “normal” manner, i.e., their corresponding level transition matrices are determined by Algorithm 2 with input parameters $N = 2$, $M = 3$ and $\mathbf{P}$ in (5.58). According to these considerations the state transition probability matrix of the QB-like structure is built using the blocks

$$\mathbf{L}^{\mathcal{R}} = (1-p)^3\,\mathbf{B}, \qquad \mathbf{F}_1^{\mathcal{R}} = (1-p)^3\,\mathbf{L} \qquad \text{and} \qquad \mathbf{F}_2^{\mathcal{R}} = (1-p)^3\,\mathbf{F}. \tag{5.64}$$

Superscript $\mathcal{R}$ denotes quantities describing this transient DTMC of Figure 5.9. (5.64) describes the joint behavior of input 1 ($(1-p)^3$ is the probability that input 1 remains in ON) and the other two inputs (given by matrices $\mathbf{B}, \mathbf{L}, \mathbf{F}$).

Finally using (5.64) the state transition probability matrix of the transient part and the state transition probability vector to state PL are represented as

$$\mathbb{P}^{\mathcal{R}} = \begin{pmatrix} \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} \\ \dots & 0 & 0 & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} \\ \dots & 0 & 0 & 0 & \mathbf{L}^{\mathcal{R}} \end{pmatrix}, \qquad \boldsymbol{\ell} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{F}_2^{\mathcal{R}}\mathbf{h} \\ \left(\mathbf{F}_1^{\mathcal{R}} + \mathbf{F}_2^{\mathcal{R}}\right)\mathbf{h} \end{pmatrix}. \tag{5.65}$$

Since input 1 is in ON state, the packet loss is possible if at the beginning of the time period there is either

- one free position in the VOQ and there are three arrivals $\left(\mathbf{F}_2^{\mathcal{R}}\mathbf{h}\right)$ or
- no free positions in the buffer and there are either
  - two arrivals $\left(\mathbf{F}_1^{\mathcal{R}}\mathbf{h}\right)$ or
  - three arrivals $\left(\mathbf{F}_2^{\mathcal{R}}\mathbf{h}\right)$.

Using $\mathbb{P}^{\mathcal{R}}\mathbf{h} + \boldsymbol{\ell} + \mathbf{s} = \mathbf{h}$ the state transition probability vector to state ST is

$$\mathbf{s} = \mathbf{h} - \left(\mathbb{P}^{\mathcal{R}}\mathbf{h} + \boldsymbol{\ell}\right). \tag{5.66}$$

The initial distribution of the transient DTMC

The initial distribution of $\mathbb{P}^{\mathcal{R}}$ in (5.65) is determined as the state of the system right after an arrival of a packet.

In following we represent the joint probability of arriving a new packet at input 1 and the "normal" behavior of the other two inputs. Using the notations introduced in (5.58) the first probability is $1-(1-q)^3$ and latter one is determined as the output of Algorithm 2 with input parameters $N = 2, M = 3, \mathbf{P}$. If $\tilde{q} = 1 - q$ then their joint behavior is described by the matrices

$$\hat{\mathbf{B}}^{\mathcal{N}} = \left(1-\tilde{q}^3\right)\mathbf{B}, \qquad \hat{\mathbf{L}}^{\mathcal{N}} = \left(1-\tilde{q}^3\right)\mathbf{L} \qquad \text{and} \qquad \hat{\mathbf{F}}^{\mathcal{N}} = \left(1-\tilde{q}^3\right)\mathbf{F}. \tag{5.67}$$

There are 4 blocks of size $\boldsymbol{\pi}$ in (5.63) which describe all 3 inputs. According to this there is a row of zeros appended to every level transition matrices in (5.67) as

$$\mathbf{B}^{\mathcal{N}} = \begin{pmatrix} \hat{\mathbf{B}}^{\mathcal{N}} \\ 0 \end{pmatrix} \qquad \mathbf{L}^{\mathcal{N}} = \begin{pmatrix} \hat{\mathbf{L}}^{\mathcal{N}} \\ 0 \end{pmatrix} \qquad \mathbf{F}^{\mathcal{N}} = \begin{pmatrix} \hat{\mathbf{F}}^{\mathcal{N}} \\ 0 \end{pmatrix}. \tag{5.68}$$

The last row expresses that in case of a new packet arrival there cannot be all the $N = 3$ inputs in ON state. It is explained by the fact that in our model there is no corresponding cell arrival to state change from OFF to ON, i.e., in case of new packet arrival there is no cell arrival from the observed input.

Starting from the steady state of the cell level model (5.63) and using the level transitions according to new packet arrival (5.68) the blocks of the initial distribution of the transient DTMC are given in Figure 5.9 are

$$\begin{aligned}
\hat{\boldsymbol{\pi}}_0^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{B}^{\mathcal{N}} + \boldsymbol{\pi}_1\mathbf{B}^{\mathcal{N}} \\
\hat{\boldsymbol{\pi}}_1^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_1\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_2\mathbf{B}^{\mathcal{N}} \\
\hat{\boldsymbol{\pi}}_2^{\mathcal{N}} &= \boldsymbol{\pi}_0\mathbf{F}^{\mathcal{N}} + \boldsymbol{\pi}_1\mathbf{F}^{\mathcal{N}} + \boldsymbol{\pi}_2\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_3\mathbf{B}^{\mathcal{N}} \\
\hat{\boldsymbol{\pi}}_i^{\mathcal{N}} &= \boldsymbol{\pi}_{i-1}\mathbf{F}^{\mathcal{N}} + \boldsymbol{\pi}_i\mathbf{L}^{\mathcal{N}} + \boldsymbol{\pi}_{i+1}\mathbf{B}^{\mathcal{N}} \quad 3 \le i \le b-1 \\
\hat{\boldsymbol{\pi}}_b^{\mathcal{N}} &= \boldsymbol{\pi}_{b-1}\mathbf{F}^{\mathcal{N}} + \boldsymbol{\pi}_b\left(\mathbf{L}^{\mathcal{N}} + \mathbf{F}^{\mathcal{N}}\right).
\end{aligned}$$

$\hat{\boldsymbol{\pi}}^{\mathcal{N}}$ is normalized as

$$\boldsymbol{\pi}^{\mathcal{N}} = \frac{\hat{\boldsymbol{\pi}}^{\mathcal{N}}}{\hat{\boldsymbol{\pi}}^{\mathcal{N}}\mathbf{h}} \tag{5.69}$$

resulting in the initial distribution of the packet level model in Figure 5.9.

The packet loss of the system

Using (5.65), (5.66) and (5.69) the packet loss probability of the system and the probability of successful packet transmission on the given path are calculated as absorbing in state

PL and ST, respectively, i.e.,

$$p_\ell = \boldsymbol{\pi}^{\mathcal{N}} \left(\mathbf{I} - \mathbb{P}^{\mathcal{R}}\right)^{-1} \boldsymbol{\ell}, \qquad p_s = \boldsymbol{\pi}^{\mathcal{N}} \left(\mathbf{I} - \mathbb{P}^{\mathcal{R}}\right)^{-1} \mathbf{s} = 1 - p_\ell. \tag{5.70}$$

Estimation for the packet waiting time

We estimate the mean packet waiting time with the mean cell waiting time. The mean cell waiting time equals to the mean system time of the cells entering the queue minus the cell service time. Since the service of the VOQ is deterministic the system time of a cell in the VOQ is $N = 3$ time slots times the queue length right after the cell arrival given that the cell is not dropped (denoted as $\tilde{\boldsymbol{\pi}}'$).

$\tilde{\boldsymbol{\pi}}'$ can be determined by the equation system

$$\begin{aligned}
\tilde{\boldsymbol{\pi}}'_1 &= \boldsymbol{\pi}_1 \left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1 + \mathbf{L}\right) + \boldsymbol{\pi}_0 \left(\frac{2}{3}\mathbf{F}_2 + \mathbf{F}_1 + \mathbf{L}\right) \\
\tilde{\boldsymbol{\pi}}'_i &= \boldsymbol{\pi}_{i-2}\frac{1}{3}\mathbf{F}_2 + \boldsymbol{\pi}_{i-1} \left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1\right) + \boldsymbol{\pi}_i \left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1 + \mathbf{L}\right) \qquad i \in [2, b-2] \\
\tilde{\boldsymbol{\pi}}'_{b-1} &= \boldsymbol{\pi}_{b-3}\frac{1}{3}\mathbf{F}_2 + \boldsymbol{\pi}_{b-2} \left(\frac{1}{3}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1\right) + \boldsymbol{\pi}_{b-1} \left(\frac{1}{2}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1 + \mathbf{L}\right) \\
\tilde{\boldsymbol{\pi}}'_b &= \boldsymbol{\pi}_{b-2}\frac{1}{3}\mathbf{F}_2 + \boldsymbol{\pi}_{b-1} \left(\frac{1}{2}\mathbf{F}_2 + \frac{1}{2}\mathbf{F}_1\right) + \boldsymbol{\pi}_b \left(\mathbf{F}_2 + \mathbf{F}_1 + \mathbf{L}\right).
\end{aligned}$$

$\tilde{\boldsymbol{\pi}}'$ is normalized as $\tilde{\boldsymbol{\pi}} = \frac{\tilde{\boldsymbol{\pi}}'}{\tilde{\boldsymbol{\pi}}'\mathbf{h}}$ resulting in the queue length distribution right after a cell arrival given that the cell enters the queue.

5.4.5 On the solution of large QBD-like DTMCs

Building the cell level DTMC as in (5.62) and solving it as in (5.63) results in the solution of a linear equation system of size $(b+1)(N+1)$ which can lead to inaccurate numerical results.

As a fast and numerically efficient solution of this we apply the folding algorithm, e.g., in [70], based solution of (5.63). The algorithm is prepared to block tri-diagonal matrices, hence we make the following repartition (5.62)

$$\mathbb{P} = \left(\begin{array}{cc|cc|cc|cc}
\mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & & & & \\
\mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & & 0 & & \\
\hline
0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & & \\
0 & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}_1 & \mathbf{F}_2 & 0 & \\
\hline
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\hline
 & & & & 0 & \mathbf{B} & \mathbf{L} & \mathbf{F}'_1 \\
 & 0 & & & 0 & 0 & \mathbf{B} & \mathbf{L}'
\end{array}\right) = \begin{pmatrix}
\mathbb{L}' & \mathbb{F}' & 0 & \\
\mathbb{B} & \mathbb{L} & \mathbb{F} & 0 \\
\cdots & \cdots & \cdots & \cdots \\
0 & & \mathbb{B} & \mathbb{L}''
\end{pmatrix}, \tag{5.71}$$

where we have enlarged the block size to a new size $(N-1)(N+1)$. The inverse of the enlarged block is calculated in the folding algorithm, by which we are increasing the complexity as well. On the other hand it is possible to enlarge the buffer size b to higher values since the computational complexity of the folding algorithm is $O(\log_2 b)$.

In the followings we give the reduction of the matrix inversion of $\mathbf{I}-\mathbb{P}^{\mathcal{R}}$, in (5.70), to the inversion of its diagonal block – denoted as $\mathbf{V}=\mathbf{I}-\mathbf{L}^{\mathcal{R}}$. Considering the matrix equation

$$\mathbf{x}\left(\mathbf{I}-\mathbb{P}^{\mathcal{R}}\right)=\boldsymbol{\pi}^{\mathcal{N}} \tag{5.72}$$

where the coefficient matrix $\left(\mathbf{I}-\mathbb{P}^{\mathcal{R}}\right)$ has an upper triangular structure, on the block level, we apply the following iterative solution for the matrix equations

$$\begin{aligned} \mathbf{x}_0\mathbf{V}&=\boldsymbol{\pi}_0^{\mathcal{N}} &&\rightarrow \mathbf{x}_0=\boldsymbol{\pi}_0^{\mathcal{N}}\mathbf{V}^{-1}\\ \mathbf{x}_0\mathbf{F}_1+\mathbf{x}_1\mathbf{V}&=\boldsymbol{\pi}_1^{\mathcal{N}} &&\rightarrow \mathbf{x}_1=\left(\boldsymbol{\pi}_1^{\mathcal{N}}-\mathbf{x}_0\mathbf{F}_1\right)\mathbf{V}^{-1}\end{aligned}$$

and all the other blocks for $i=2,\ldots,b$ are

$$\mathbf{x}_{i-2}\mathbf{F}_2+\mathbf{x}_{i-1}\mathbf{F}_1+\mathbf{x}_i\mathbf{V}=\boldsymbol{\pi}_i^{\mathcal{N}} \qquad \rightarrow \mathbf{x}_i=\left(\boldsymbol{\pi}_i^{\mathcal{N}}-\mathbf{x}_{i-1}\mathbf{F}_1-\mathbf{x}_{i-2}\mathbf{F}_2\right)\mathbf{V}^{-1}$$

Rearranging (5.72) results in $\mathbf{x}=\boldsymbol{\pi}^{\mathcal{N}}\left(\mathbf{I}-\mathbb{P}^{\mathcal{R}}\right)^{-1}$ which implies that from (5.70) the packet loss probability (p_ℓ) and the probability of successful packet transmission (p_s) of the observed VOQ can be calculated as

$$p_\ell=\mathbf{x}\boldsymbol{\ell} \qquad\qquad \text{and} \qquad\qquad p_s=\mathbf{x}\mathbf{s}. \tag{5.73}$$

5.5 Experimental Results and Summary

5.5.1 Computation study: general case

In this section we perform different computations to present the numerical results of the previously introduced analysis. In following we mainly present the results related to the packet loss analysis. Although the analysis in the previous sections was presented to path $in1-VOQ_{00}-out0$ only, in this section we present results for a packet loss trough all possible paths. Unfortunately, due to the high complexity of the general case characterization model, the results will be shown only for small-sized switches. We consider that the packet loss occurs in the queue if at least one cell of the packet is dropped. According to the analysis presented in Section 5.2 the arrival input traffic is characterized by three main parameters, which were as matrices t,p,q. Matrix t is responsible for choosing the destination of an arriving variable size packet. As we consider a destination to be randomly chosen between all available outputs, each element of the matrix is equal to $t_{i,j}=1/N$,

Figure	5.14	5.15
N	3	3
b	$6, \dots, 34$	60
p	$\frac{1}{20}$	$\frac{1}{10}, \frac{1}{30}, \frac{1}{50}$
q	$\frac{1}{3}$	$\frac{1}{2}$
t	$\frac{1}{N}$	

Table 5.3: Parameters used for the numerical studies

where N is a size of the switch. According to the matrix p the average packet length is chosen. In our experiments we generate packets with equal average packet length for all inputs. Additionally to the presented analytical results, we used the LB switch simulator, which is capable to calculate packet loss probabilities for the switch with variable size packets and with finite buffers. The simulation and analysis results show good match and are represented in Figure 5.14. The main parameters our modelling scenario were given in Table 5.3.

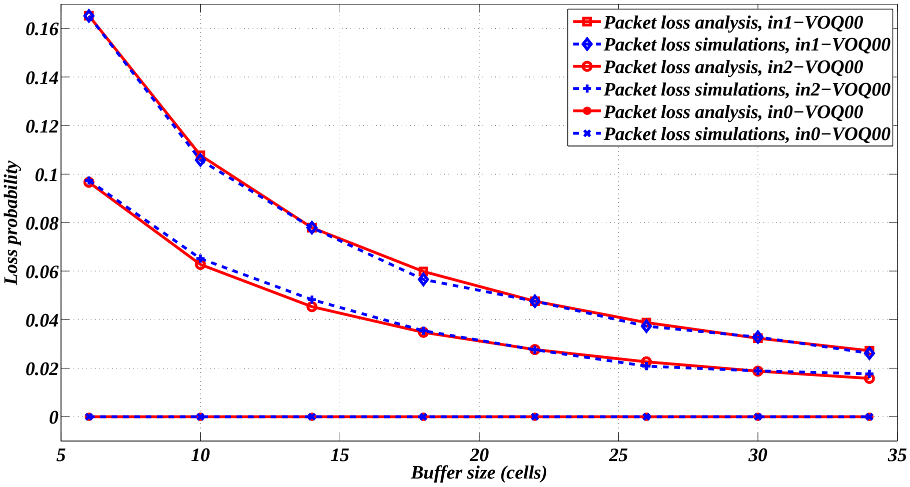


Figure 5.14: Packet loss as a function of buffer size; comparison between analysis and simulations

In Figure 5.14 by means of mathematical model and simulations we show the difference in the amount of packet loss probabilities that experience a single queue depending on the path (sequence i, j, k) considered. In particular in 3×3 LB switch the path in0-VOQ_{00}-out0 will not experience any loss due to the fact that this queue has service and arrival when the crossbar is in "bar" position. Since we assume that the service inside the queue is happening before the arrival, the VOQ_{00} will never experience cell and packet loss. The loss(both packet and cell one) of the queue with path in1-VOQ_{00}-out0 will be greater than the one of path in2-VOQ_{00}-out0. The VOQ_{00} considered for our analysis is served last in a cycle (after arrivals from in0 and in2) and, consequently, has the highest loss probability. Particularly, by the time of a cell arrival from input 1, VOQ_{00} has a possibility to experience a cell drop from input 2, together with a drop of its own cell if the buffer is full. According to the same explanations the in2 regarding to the VOQ_{00} is served first and can experience only a drop of its own cell. The properties of different packet loss presented above give some insight on the problem why it is difficult to calculate the possibility that a cell drop(and further on the packet) will happen due to congestion in the nearest future and to block the packet in the input in advance for the necessary number of time slots. In order to do such kind of calculations each input should have extended information about the occupancy(the occupancy is changing each time slot) of all the $VOQs$ (since packet can have any destination) together with the knowledge about the current traffic in all the inputs (to evaluate with arrival and when overflow the buffer). In spite of the complicated implementation, the no-loss LB switch architecture with centralized management will be presented in Section 6.2.

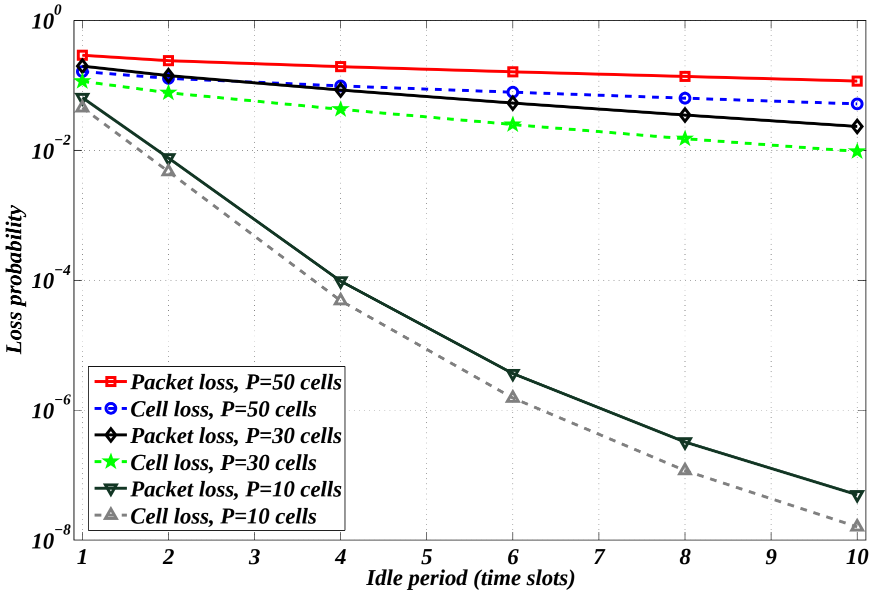


Figure 5.15: Packet/cell loss of a specified path as a function of idle period

In Figure 5.15 we present the dependence of the packet loss for the specified path as a function idle period. As it is shown on the picture the LB switch packet/cell loss probability is not decreasing intensively if the packet length remains larger than the average length of the idle period. The significant drop of the packet/cell loss probability function is observed after the interpacket period is equal or near to the packet length size. This property can be used in the dynamic control of the central stage loss probability while operating at the input stage.

In spite of the high precision which the model with full input traffic characterization can provide, the overall complexity of the mathematical model makes impossible to analyze the architectures with large number of ports and buffers. Therefore, in the next sections we will focus on the demonstration of the results for larger switches obtained from the approximated models.

5.5.2 Computation study: 2^N complexity case

In this section we present the comparative study of the analysis with ON/OFF model and the simulation results using the memoryless (geometric) assumptions and the notations introduced in Section 5.3. We executed two studies with two different sets of parameters given in Table 5.4 representing a set of considered parameters in detail, instead of just the ON and the OFF parameters (the model is derived from the detailed parameters). Although the independent variables are discrete we used continuous plots to improve visibility of Figure 5.16, Figure 5.17.

study 1		study 2	
variable	value	variable	value
N	4	N	3,...,8
p_{ij}	$\frac{1}{20}$ (av. 20 cells)	p_{ij}	$\frac{1}{50}$ (av. 50 cells)
q_i	$\frac{1}{3}$ (av. 2 cells)	q_i	$\frac{1}{6}$ (av. 5 cells)
t_{ij}	$\frac{1}{N}$	t_{ij}	$\frac{1}{N}$
b	$8, \ldots, 40$	b	20

Table 5.4: The main parameters of the computation

Study 1 Figure 5.16 plots the packet loss probability of different types of paths through VOQ$_{00}$ versus the buffer size. The loss of a single queue is decreasing with increase of the buffer size, which is obvious with increase of system capacity. Here the dependence of packet loss on the chosen paths is also shown. The set of parameters of study 1 is given in the left hand side of Table 5.4. The experimental results proof the validity of our assumptions. In particular, in Figure 5.16 we show, that the queue does not experience any loss for the type-0 path $\{0, 0, 0\}$, and, as expected, the higher the d value is the higher the loss probability of the path is. It is also shown in Figure 5.16 that the higher the buffer size (b) is the less the difference between the loss values for types.

Study 2 Due to lower analysis complexity in comparison with [4], the packet loss of a single queue can be evaluated for larger switches – than those ones in [4]. Figure 5.17 plots the packet loss of the queue if the switch size is increasing – up to the solvable highest size of this model. The detailed set of parameters used in Study 2 is shown in the right hand side of Table 5.4. We present packet loss only for those two traffic path ($\{1, 0, 0\}$ and $\{2, 0, 0\}$) which exist for all considered switch sizes. As it is shown on the plot, with the increase of the switch size, the packet loss decreases. As the average packet size and idle period size keeps to be the same, the increase in number of ports increases the number of queues at the central stage and consequently the buffering capacity for the same set of parameters. Correspondingly, the higher is the LB switch buffering capacity the lower packet loss is experienced.

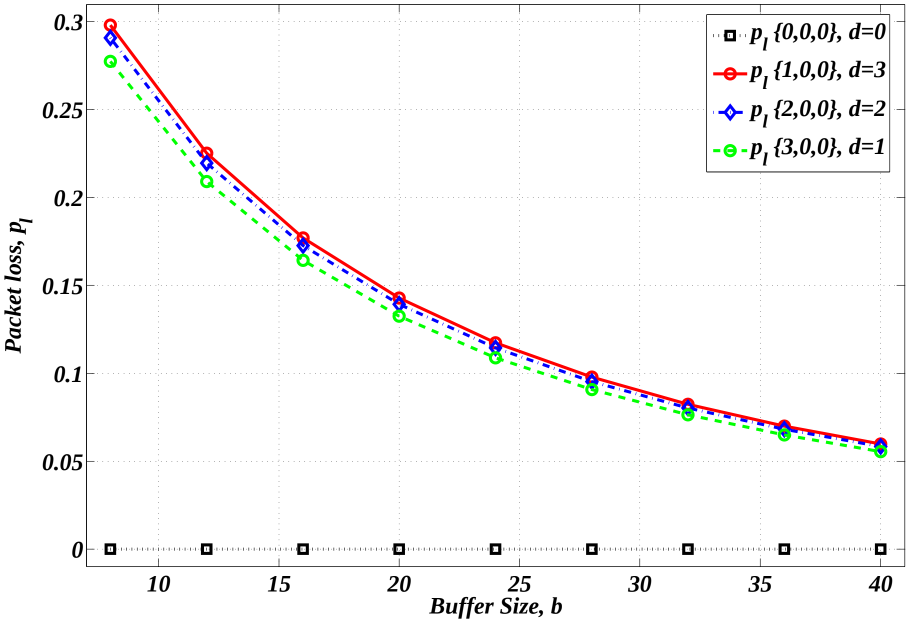


Figure 5.16: Packet loss probability versus buffer size (study 1)

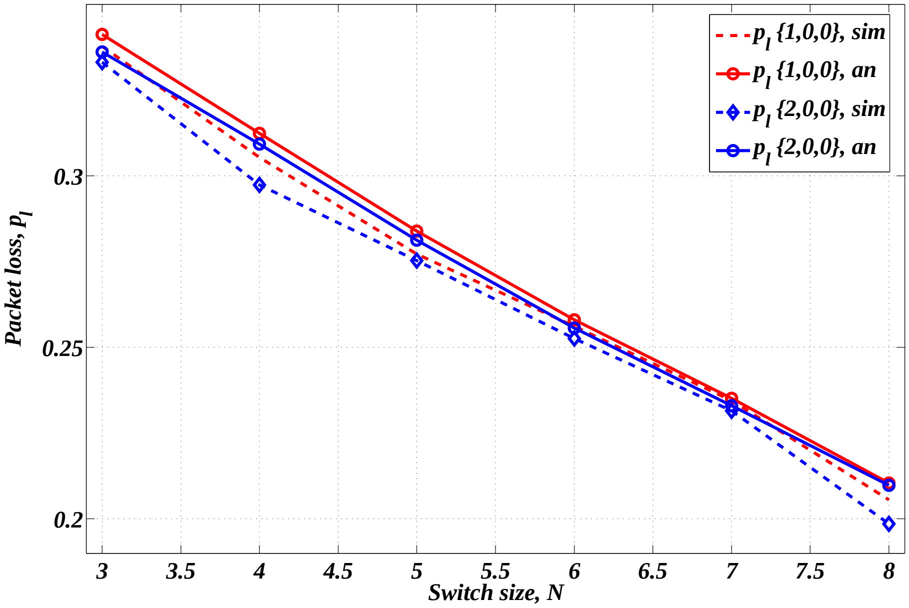


Figure 5.17: Packet loss probability versus switch size (study 2)

5.5.3 Computation study: $N + 1$ complexity case

In contrast to [2,4] (presented also in sections 5.2 and 5.3) where we described extended methodology of packet loss analysis in the LB switch, this section presents optimized solution with linear complexity. The computational study has two parts. The first part shows the behavior of the packet loss and waiting time of the LB switch as a function of buffer length and switch size. The second part examines some extreme cases when central

Figure	5.18	5.19	5.20	5.21	5.22	5.23
name	p_ℓ vs. b	p_ℓ vs. N	T vs. b	T vs. N	p_ℓ vs. b	T vs. b
	without folding algorithm				with folding	
N	4	$4,\dots,32$	4	$3,\dots,33$	3	
b	$8,\dots,40$	36	$8,\dots,40$	127	$9,\dots,999$	
$\hat{p}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{20}$	$\frac{1}{50}$		
$\hat{q}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{3}$			
$\hat{t}$	$\frac{1}{N}$					

Table 5.5: Parameters used for the numerical studies

stage buffers are large to show the power of the folding algorithm based solution method presented in Section 5.4.5. For the results of this section we used the parameters given in Table 5.5. In order the comparative analysis, we made the specified measurements also with our LB switch simulation tool.

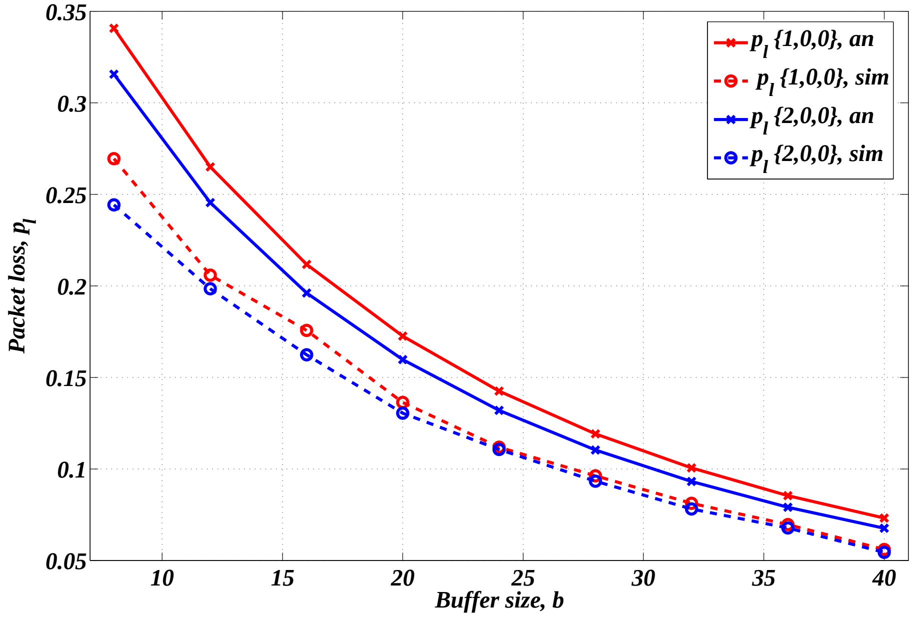


Figure 5.18: Packet loss versus buffer size

Part 1 In [2, 4] we examined the dependence of packet loss at the central stage buffers on the buffer size and switch size. It was found that the packet loss probability strongly depends on the chosen path ($\{i, j, k\}$). Figure 5.18 and 5.19 present similar results using the approximate model introduced in the previous sections.

Figures 5.20 and 5.21 indicate another performance characteristic, the packet waiting time estimator compared to simulation results. The packet waiting time is evaluated considering only the successfully transmitted packets. The packet waiting time is generally increases together with the buffer size (larger interval between cell arrivals and services),

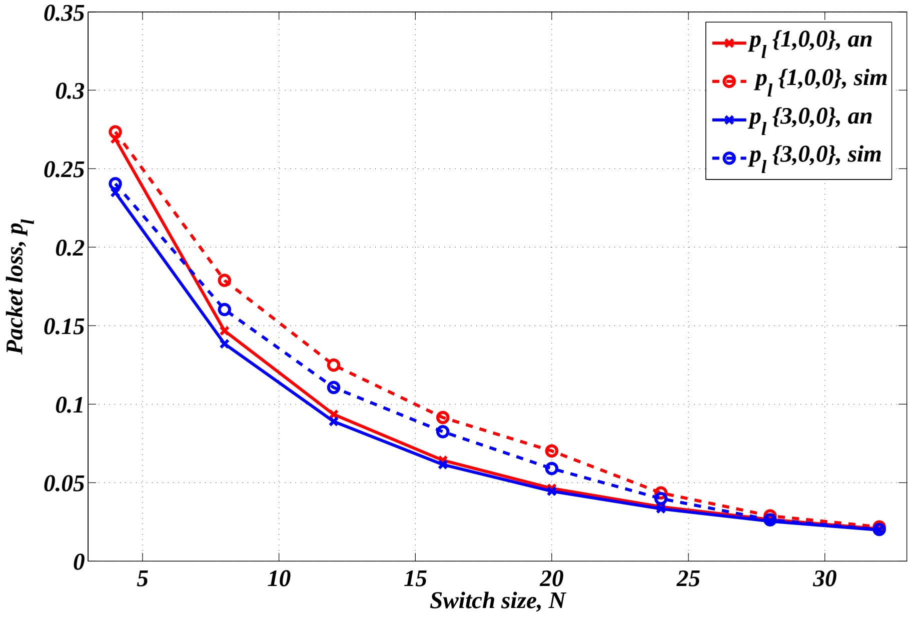


Figure 5.19: Packet loss versus switch size

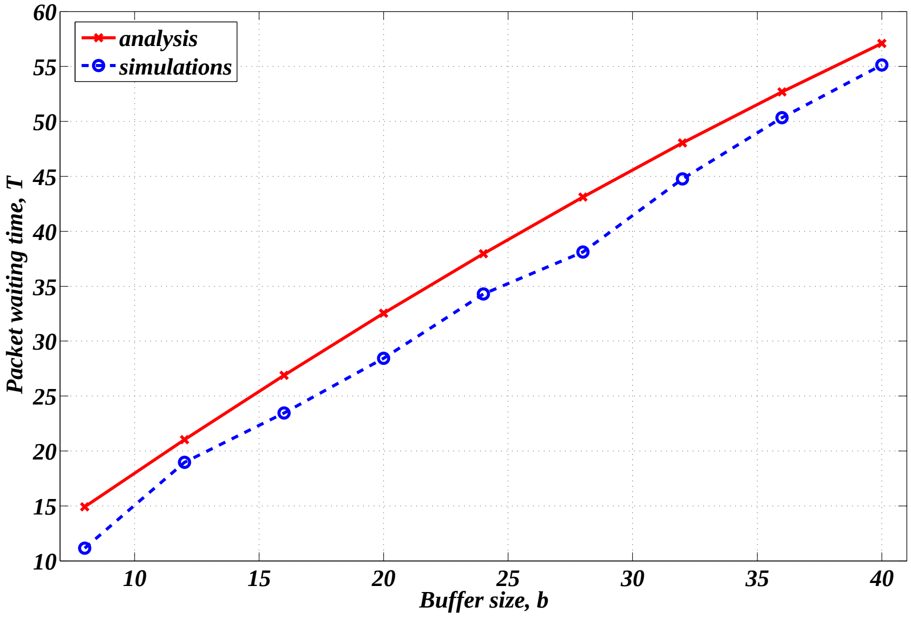


Figure 5.20: Packet waiting time versus buffer size

like in Figure 5.20 and switch size (cells are spread to more queues), like in Figure 5.21.

Part 2 Figure 5.22 and 5.23 shows the applicability of the analytical model for large buffer sizes. According to presented results, we admit that the ratio between the switch size and buffer length of the VOQs is a crucial issue for the expected packet loss and system performance. Unfortunately, the optimal set of parameters (e.g. switch size and

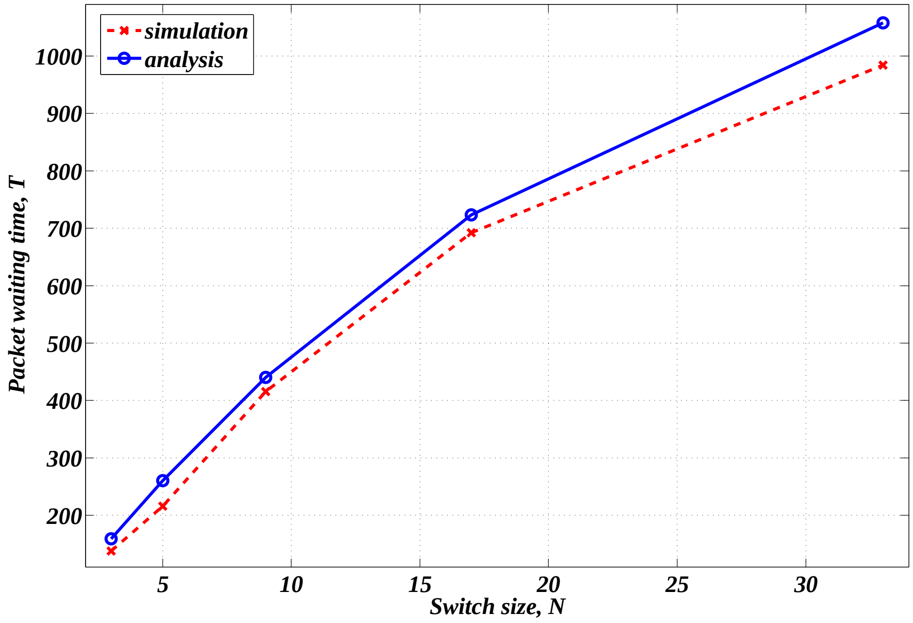


Figure 5.21: Packet waiting time versus switch size

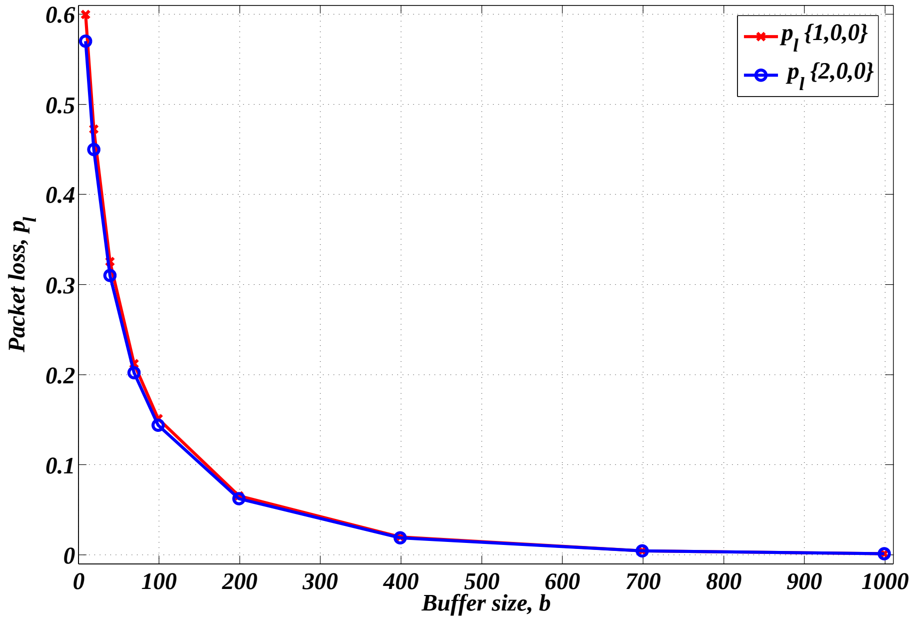


Figure 5.22: Packet loss as a function of buffer size

buffer length) is not constant and should be chosen to the specific needs.

5.5.4 Additional simulation results

The following section describes a set of additionally performed simulation results for the LB switch with finite buffers and variable size packets. For our analysis we assume that the switch is not overloaded and the service rate of the traffic inside the switch is higher

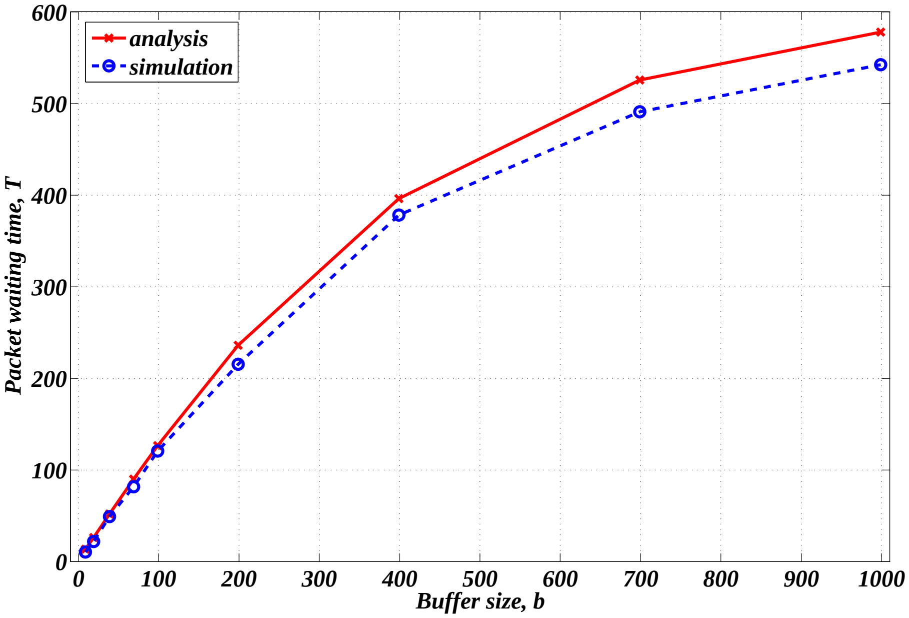


Figure 5.23: Packet waiting time versus buffer size

than the rate of the arriving traffic. To consider the main performance characteristics of this switch only the simulation results were used. The calculation of the cell and packet loss/delay was carried out during the time period T equal to 40000 time slots. Each value on the curve represents an average of 100 measurements. During the T time period packets of variable length are asynchronously sent to each input. In particular, traffic profiles (input packet sizes and interarrival periods) are generated according to certain distributions. Arriving packets are assumed to be distributed uniformly between the set of outputs. The geometric (exponential) distribution, which accounts only one parameter for probability density function (pdf), is used in our analysis. Besides this, we present simulation results utilizing the Pareto distribution (with at least two defined parameters), which is considered to be a traffic profile similar to the statistical behavior of realistic traces. For geometrical distribution value p in Table 5.6 characterizes an average packet size used in the simulations. Parameter q presents an average interarrival period between two consecutive packets.

Simulation results are divided into two main parts. The first part presents simulation results associated with the packet loss analysis. Here we show the differences between the cell loss inside the switch dependent on the various types of the packet loss. The second part is dedicated to another performance characteristic of the switch - packet delays. An average value of packet delay at the central and output stages will be described using the exponential and the Pareto traffic patterns. Please note that the exponential distribution is considered as a continuous counterpart of the geometric distribution.

Part 1. Cell/Packet loss analysis. The total buffering capacity of the central

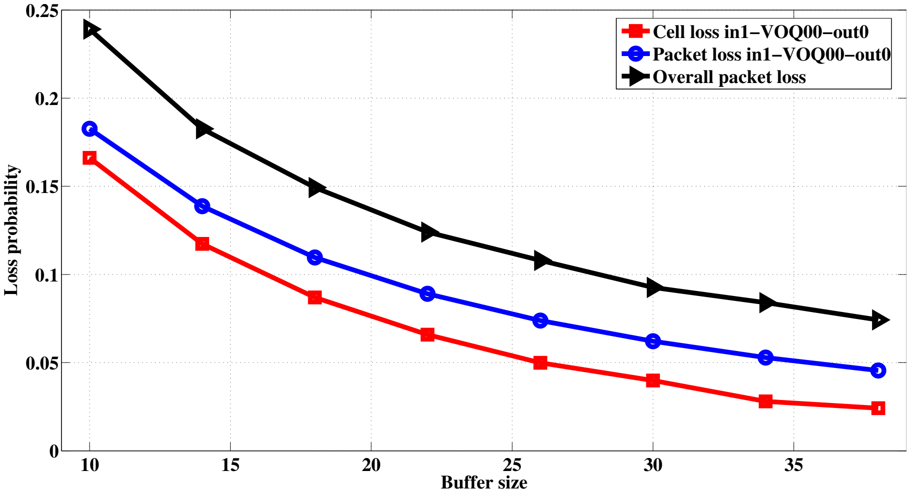


Figure 5.24: Dependence of the overall packet loss, packet loss on a path and cell loss versus buffer size

Fig.	5.24	5.25	5.26
name	p_{pl}, p_{cl} vs. B	p_{pl}, p_{cl} vs. N	p_{path} vs. B
N	8	$4, \dots, 32$	16
B	$10, \dots, 38$	40	$18, \dots, 38$
p(cells)	0.05 (20)	0.025 (40)	0.03(3) (30)
q(cells)	0.5 (2)	0.3(3) (3)	0.5 (2)
$Dist.$	Exp.	Exp.	Exp.

Table 5.6: Parameters used for simulations of Part 1

stage can be expanded due to increase of the number of buffers in central stage VOQs and by enlargement the switch size. The packet loss of the system is expected to decrease under indicated conditions (for simulation parameters see Table.1). Figure 5.24 demonstrates dependence of the packet/cell loss on the amount of buffers at the central stage. The cell/packet loss is decreasing with the increase of the buffer size, as was suggested. Under the conditions, when packet length and packet interarrival periods are exponentially distributed, the overall packet loss of the system appears to be larger than the packet/cell loss in the path in$_1$-VOQ$_{00}$-out$_0$. The cell loss in the path is always lower than corresponding packet loss. This behavior is explained by the fact that a single cell discard can cause a drop of the whole sequence of cells belonging to the same packet.

Alike behavior is observed for dependency of the packet loss probability on the switch size (see Figure 5.25). Enlargement of the physical size of the switch decreases the probability of the overall packet loss, packet and cell loss of the path. Due to the switch size enlargement, the total number of the queues at the central stage grows. However, Figure 5.25 shows that intermediate values of the overall packet loss can be lower than the value of the packet loss inside the path. Interestingly, the overall packet loss in this

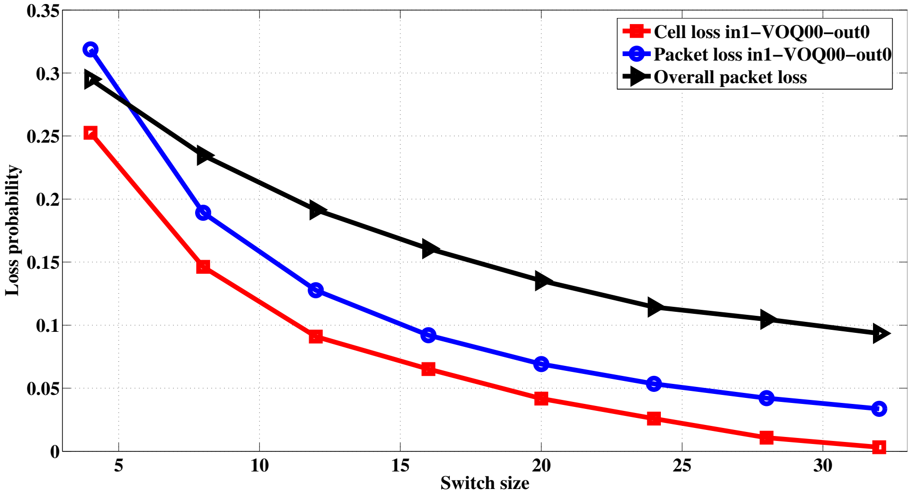


Figure 5.25: Dependence of the overall packet loss, packet loss on a path and cell loss versus switch size

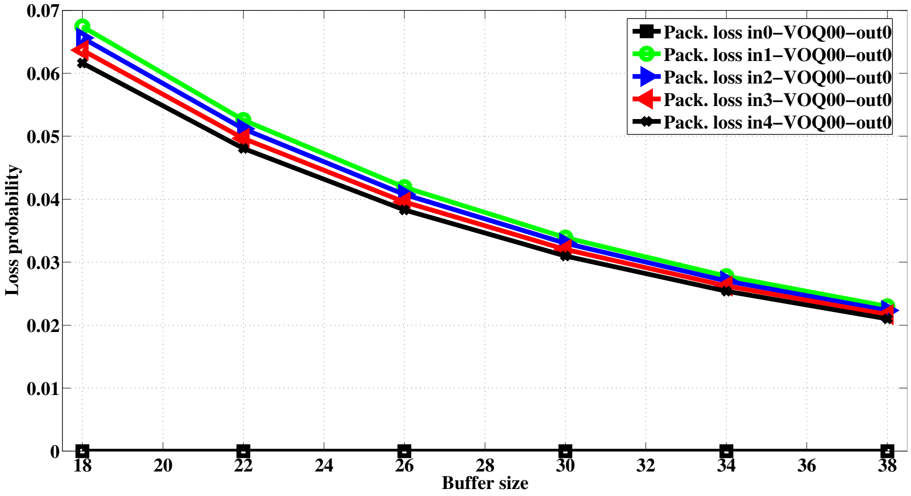


Figure 5.26: Dependence of packet loss inside a path over buffer size

case decreases with the lower slope when compared to the corresponding packet/cell loss of the path.

Figure 5.26 demonstrates changes in the packet loss inside the path (exponential distribution) caused by increasing buffer size of central stage VOQ. The packet loss decreases with increase of the buffer size, but differs for distinct traversing paths, for a certain buffer size. This implies a strong dependency of the packet loss on a traversing path chosen, and can be explained by unique crossbars interconnection pattern for an individual path. This phenomenon was described in [2, 3].

Part 2. Packet delay analysis. As confirmed by simulations (Section 5.5.3) the packet/cell loss can be significantly decreased by expansion of the buffer and switch size, however theoretically this will affect the performance of the switch and result in delays.

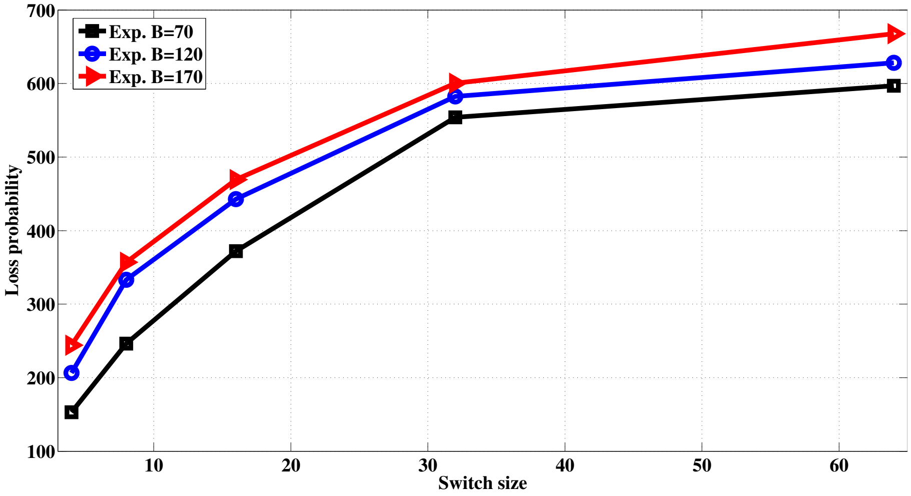


Figure 5.27: Dependence of average packet traversing delay at the second stage versus switch size

To investigate this issue we characterized the packet delay for the LB switch. Main parameters of simulations are shown in Table 5.7. As expected, in Figure 5.27 we observe an increase in packet delay on the central stage with the enlargement of the switch size. The result means that while increasing switch size the number of the central stage VOQs is growing automatically. So each packet in the input stage is making placements of cells to extra queues. For a given switch size stage 2 packet delay also depends on the amount of buffering in each VOQ. The packet delay at stage 2 correlates with the reassembly delay (in time slots) at stage 3 that shown in Figure 5.28. Due to the difference between the arrival of the first cell of considered packet and the last one (stage 2 packet delay), the reassembly delay is expected to elevate with grows of stage 2 packet delay. Based on stage 3 delay results it is possible to make some conclusions on the expected buffer size at the output stage. In the particular case of Figure 5.28 the ratio between output packet delay and stage 2 delay is roughly in the proportion 1/2. Namely this property means that the buffer size of the reassembly unit is expected to be lower than the central stage buffer. Unfortunately it is impossible to evaluate the precise ratio between the central stage buffer amount and output buffer size due to several factors which can influence on the reassembly process. In particular, we name packet loss at the central stage, uniformity of packets distribution between the outputs, arrival traffic matrices and others.

Though the heavy-tailed distributions are important in the real traffic studies they are difficult to analyze. Due to the fact that the pdf of heavy-tailed distributions can be build using two or more parameters, it is quite complex to fit the exponential distribution to the heavy-tailed one. In the following, we use the simplest heavy-tailed distribution –

Fig.	5.27	5.28
name	D_{ST2} vs. B	D_{ST2} vs. D_{ST3}
N	$4, \ldots, 64$	$4, \ldots, 34$
B	$70, 120, 170$	$70,\ 170$
p(cells)	0.02 (50)	0.02 (50)
q(cells)	0.2 (5)	0.2 (5)
$Dist.$	Exp.	Exp.

Table 5.7: Parameters used for simulations of Part 2

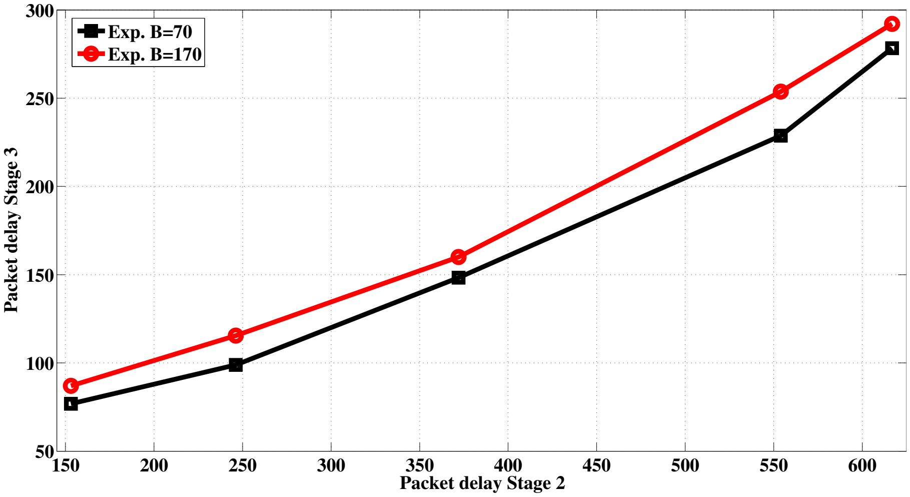


Figure 5.28: Average packet delay at the output stage as a function of central stage packet delay

the Pareto distribution. The pdf of the Pareto distribution is

$$f(x; a, b) = \frac{ab}{x^{a+1}},$$

with the mean value

$$E(x) = \frac{ab}{a-1},\ a > 1,$$

and variance

$$var(x) = \frac{b^2 a}{(a-1)^2(a-2)},\ a > 2.$$

Figure 5.29 shows an average packet delay at stage2 as a function of the central stage buffer size. The average packet length is assumed to be 40 cells and the average packet interarrival period is 4 cells. The traffic patterns with the same mean (40 cells) but different variances (and correspondingly different parameters a and b) are presented in Figure 5.29. As was already described for exponential distribution of the packet length, simulation using Pareto distribution shows increased packet delay upon increasing buffer size (See Figure 5.29). The character of the dependency on the buffer size is the same

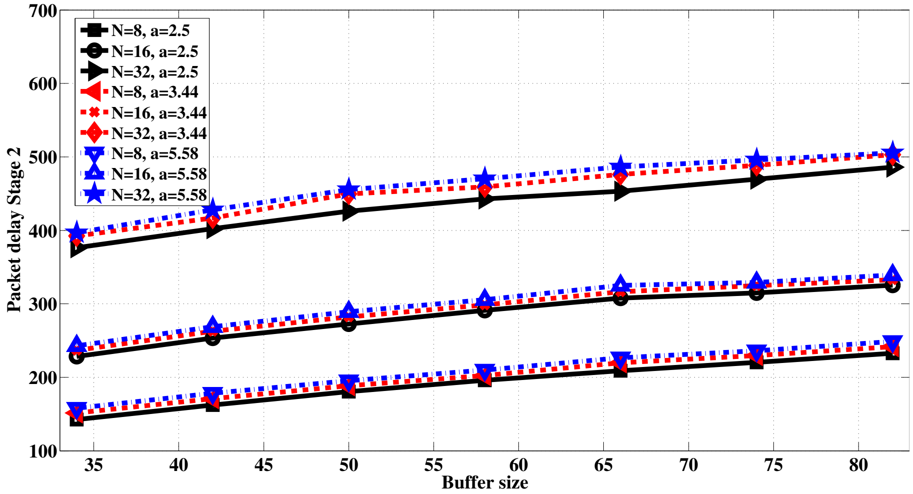


Figure 5.29: Packet delay stage2 versus buffer size, average packet size 40 cells, interarrival period 4 cells

Fig.	5.29(black)	5.29(red)	5.29(blue)
$E(x)$	40	40	40
a	2.5	3.45	5.58
b	24	28.4	32.8
N	8, 16, 32	8, 16, 32	8, 16, 32
B	$34, \ldots, 82$	$34, \ldots, 82$	$34, \ldots, 82$

Table 5.8: Parameters of used Pareto distribution

for all N, but has different absolute values. The results show that even with the same packet size mean, the resulting curves are different for various set of parameters a and b. This implies the fact that none of the presented curves of the Pareto distribution can be compared directly to the exponential distribution.

However, Figure 5.29 shows an interesting tendency - the ”difference” between the pairs of curves (for N=8, N=16 and N=16, N=32) for all assigned a remains the same. In other words the difference between the delay value for a=2.5, B=50, N=16 and N=8 will be the same like the delay value for a=2.5, B=66, N=16 and N=8, and similar to parameters a=3.44, B=50, N=16 and N=8! The property denotes that the difference between the examined curves for N=16 and N=8 same like for N=32 and N=16 remains the same for any set of parameters a and b (but the same mean $E(x)$), where the mean and variance exist ($a > 2$).

The fact that the ”difference between the curves” is independent on the parameters a and b gives the possibility to compare this ”delay grows” parameter under Pareto and exponential traffic patterns. Figure 5.30 shows the amount of extra delay which appear with switch size enlargement (N=8 and N 16) for both distributions. The obtained values are presented in Table 5.9. According to the results, the packet reassembly delay at the

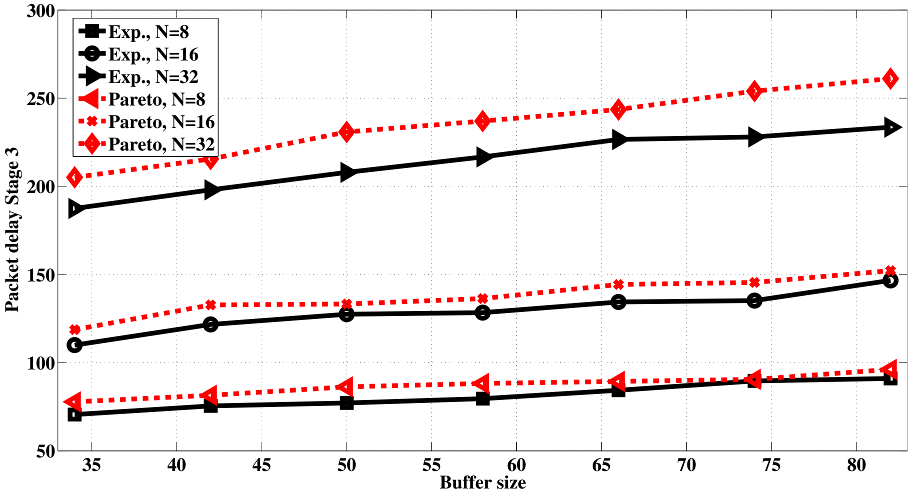


Figure 5.30: Dependence of packet reassembly delay on buffer size

Distr.	Exp.	Pareto
E(x) - pack.	40	40
E(x) - interarriv.	4	4
$N16 - N8$ (stage3)	48	56
$N32 - N16$ (stage3)	85	105
N	8, 16, 32	8, 16, 32
B	$34, \ldots, 82$	$34, \ldots, 82$

Table 5.9: Packet delay grows analyzed in Figure 5.30

output stage is increasing more intensively under Pareto distribution. The property can be caused by the long-range dependency of Pareto-distributed traffic. More precisely, the Pareto distribution can generate with high probability the packet sizes which are in the large range from the mean packet size. Thus such kind of pattern creates larger delays on the central and correspondingly on the output stage.

5.5.5 Summary

According to the analysis and simulation results performed, the packet loss and the packet delay inside the system correlate with the amount of buffering space at central and output stages. In order to achieve high throughput in the system (and minimize packet loss), the largest possible amount of buffers should be implemented at the central and output stages. Please also note, that the internal central stage packet loss inside the LB switch is changing according the traffic traversing path, which applies some restrictions on the maximum amount of buffering used at the output reassembly unit. On the other hand, our measurements show that the increasing buffer size of the system correlates with the growing internal packet delay. As the buffering capacity of the system can be accumulated using both 1) enlargement of queue size – B and 2) enlargement of switch size – N, some optimal ratio between B, N, Packet loss and Delay should be found. Using data obtained and analyzed in this chapter we present several solutions for 1) minimization of the CS packet loss as well as 2) avoidance of the packet loss at the CS VOQs.

Chapter 6

Packet Loss Minimization and Avoidance

The parameters estimation of the LB switching architecture, under a set of extensive assumptions, was given in previous chapters and in [3, 4]. According to the new assumptions the arriving packets of variable size are segmented into a smaller equal size data cells and transmitted through the switch in a cell-by-cell basis. It was also considered that the arrival rate of input packets is less than the switch's internal transmission rate. In other words switch inputs are not overloaded and input packet loss can never happen. The packet/cell loss which appears at the central stage buffers due to the overflow can potentially introduce the problems at an output reassembly unit. Since cells transmission inside the switch is done without any respect of the possible congestion in the next stage, a single cell loss at the central stage buffers will not allow to reassemble this packet at the output stage. Moreover, a set of such incomplete packets will create congestion at the output stage, waste considerable amount of buffering capacity as well as will require some sophisticated algorithms to identify and remove incomplete packets. Finally, as it was shown in [4] the internal packet loss probability inside the switch is strictly depends on the crossbars interconnection pattern and the path (input - central stage - output sequence) chosen for evaluation.

In this chapter we present a novel LB service protocol, which allows to drop arriving packets also at the input stage if these packets will exceed the allowed buffering threshold at the central stage buffers. In such a way, an input stage will drop a whole packet at arrival which reduces the probability of serious problems at the output reassembly unit because of incomplete packets arrival. In the following we also present the mathematical analysis in order to evaluate the joint input/central stage packet loss and present a way to minimize the mentioned packet loss.

Additionally to the mentioned service protocol, this chapter proposes a novel NoLoss switching architecture (Section 6.2) which allows to completely avoid packet loss at the central stage buffers. The operation principles of this architecture are based on the appropriate traffic management presented at the input and central stage buffers. The LB switch architecture is equipped with centralize controller which is capable to obtain occupancy information from the inputs and central stage buffering sets and make decision whether arriving packets will be dropped in the future.

In the following chapter we present both solutions while introducing all the details related to the operation principles and representing results obtained.

6.1 Protocol for Packet Loss Minimization

In this section we introduce a novel service protocol which controls the amount of packet loss at the central stage buffers by allowing the packets to be dropped at the input stage. In particular we introduce an artificial buffering threshold at the central stage buffers in such a way that packets at the input stage are dropped if the central stage buffer occupancy is above the predefined threshold. Although the protocol is implemented in the controller with centralized management it has small computation and communication overheads. The protocol does not guarantee a zero packet loss at the central stage buffers, however it allows to reduce the amount of incomplete packets at the central stage and, thus, control the maximal amount of incomplete packets arriving to the output. Moreover our analysis shows that it is possible to control input and central stage packet loss due to threshold variation in such a way that the joint input-central stage (I-CS) packet loss is the minimal between the mentioned boundaries.

In the following first we present the overall description of the considered LB switching architecture, paying particular attention to the realization of the service protocol by means of centralized controller (Section 6.1.1). Next, in Section 6.1.4 we present mathematical analysis which allows to evaluate the joint I-CS packet loss inside the switch. Section 6.1.9 will present computation study related to the protocol performance for various switch, central stage buffer and packet sizes. This part will also verify the mathematical analysis with developed simulation model. Finally, Section 6.1.10 will conclude the section.

6.1.1 The Implementation

In comparison to the traditional two-stage LB switch presented in [3], the examined architecture includes also a centralized controller (Figure 6.1). The functionality of the LB switch without external management is described in details in [2]. It is assumed that the variable length packets arriving to inputs are stored and segmented into fixed-size data cells at First-In-First-Out (FIFO) queues. The arrival rate of incoming packets is considered to be lower than the service rate of the switch. Therefore, input packet loss is presumed to be zero. The transmission process of cells through the switch is also well reported in [16, 17]. Data cells arriving to outputs are reassembled at re-sequencing and reassembly unit back to packets.

The implementation of the reassembly unit is not specified in this work, but it can be implemented based on the standard scheme described in [66]. In short, two possible implementations of the output re-sequencers can be considered. The first approach performs the insertion of the sequence numbers to each packet (e.g. packet sequence number and cell sequence number) segmented at egress ports of a switch. This solution is able to successfully reconstruct a packet if all cells of this packet reached the destination. If by some reason at least one cell was not received, the remaining members of the packet can stay enqueued for a considerable time. Additionally, a set of timers may control the amount of time spent by a packet in a queue. In case the threshold is overtaken, the incomplete packet is discarded from the queue. The second approach implements timers at output re-sequencing queues of a switch. In particular, the timers are reset to zero as soon as a first cell of a packet arrived to the queue. Irrespective of the fact whether the packet is complete or incomplete it is discarded from the queue as soon as the times expires.

As it was shown in [4, 69] the traditional LB switch with finite buffers can have a cell/packet loss due to the central stage buffers overflow. As a result, the arrival of incomplete packets to the output RRU can introduce large buffering space wastage and enormous delays. To handle these issues, the service protocol was introduced in this section.

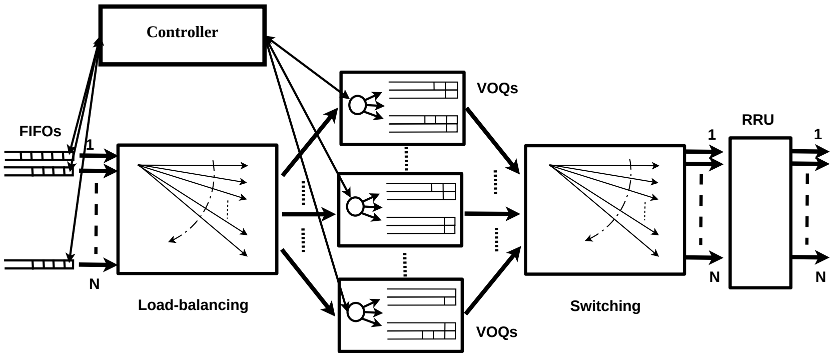


Figure 6.1: The considered LB switch

Since the main point of congestion in the traditional LB switch is found in central stage buffers, the following two values should be observed/controlled in order to minimize/avoid packet loss. In particular, it is necessary to carry the information: 1) about the input packet arrivals during a time slot (the basic time unit of the system) and 2) information about the occupancy of the central stage buffers. Since each stage of a basic two-stage switch is independent from all the other stages of the switch, the most appropriate way of the data collecting is by means of centralized unit. The controller is using detached links for information exchange and is interconnected with both inputs and central stage buffer sets (CSSs). Please note, that in order to maintain distributed control in the system with such a service protocol the switch might have considerably greater communication and computation overheads than that with centralized control (this issue is discussed in section 6.1.3). The service protocol which is implemented in the centralized controller, can set the artificial buffering threshold at the central stage buffers (either statically or dynamically) in order to distribute a packet loss between input and central stages. One of the important considerations is that the service protocol allows to drop packets (of variable size and directed to some specific output) at an input stage in case if the occupancy of at least one VOQ, where the packet is supposed to be distributed, is greater than the defined threshold.

Example of the protocol function. Suppose that a packet composed of Q cells and directed to output k is just arrived to input i. According to a standard operation of the switch, this packet could be distributed (depending on the current crossbar interconnection and packet size) cell-by-cell between the following virtual output queues – $VOQ_{0,k}$, $VOQ_{1,k}$, $VOQ_{2,k}$, ... , $VOQ_{N-1,k}$ at CSSs. However, in the current implementation of the switch, the controller performs congestion detection based on the value of an artificial buffer T, set at all central stage buffer sets. The packet is allowed to be forwarded in case if the occupancy of $VOQ_{0,k}$, $VOQ_{1,k}$, $VOQ_{2,k}$, ... , $VOQ_{N-1,k}$ is less than threshold T. Otherwise, if at least one of these queues has the occupancy greater than T, the packet in an input is dropped. Such kind of comparison is performed for all the packets arriving to inputs during a time slot.

On the definition of the different loss probabilities. Packet loss probability inside the system is proved to be different for different transmission paths (input - central stage buffering set - output sequence) [2]. In particular, *cell loss probability* is considered to be a ratio between a number of cells which were dropped from the observed VOQ versus the total number of cells sent through the VOQ. In contrast, a packet is considered to be *lost* on a VOQ if at least one of its cells is lost in the observed VOQ. If one of the remaining cells of this packet was dropped at some other queue, this packet is not considered to be dropped at this queue, since a packet can be dropped only once. Correspondingly, packet loss probability on a path takes into account the ratio of dropped and successfully transmitted packets refereing to the considered triple – $\{in - i, VOQ - \{j, k\}, out - k\}$.

In this section we introduce the input packet loss on a path in the way similar to the presented packet loss on a path for a VOQ. The *input packet loss* appears only after a specific controller decision to drop an arrived packet. Please note that input packets are removed entirely before the actual transmission is made. An arrived packet is assigned to be dropped at some specific path – in$_i$ - $VOQ_{j,k}$ - out$_k$ – if this packet is currently placed at input i, has as a destination point output k and supposed to be transmitted to j-th CSS according to the current crossbar configuration. In the similar way, packets successfully transmitted through a path are accounted.

Throughout the section we performed evaluation of the joint I-CS packet loss probability, with relation to property of packet loss difference over various traversing paths and to the model presented in [3]. The joint I-CS packet loss through a path is considered to be a ratio between the sum of packets dropped both at the input and central stage buffering set to the total number of packets transmitted through this path.

6.1.2 Information exchange in the controller

In this section we present the controller design and main operation principles used for implementation of the service protocol. The controller which is interconnected with both all inputs and CSSs is using N^2 bidirectional links for information exchange. The management unit is synchronized with the rest of the system, so the information exchange and all the computations are performed within a time slot basis.

The protocol can operate in two different modes. The first mode presumes statical initialization of a buffering threshold at the central stage, so the threshold is initialized before switch operation. In this case the threshold remains similar for the entire operation interval of the switch. The second approach assumes the threshold to be dynamically changing in time, e.g. can be modified at the beginning of any time slot. In this case the joint I-CS packet loss probability will be dynamically changing in correspondence to a threshold set. Section 6.1.9 presents results only for a statically configured threshold operation.

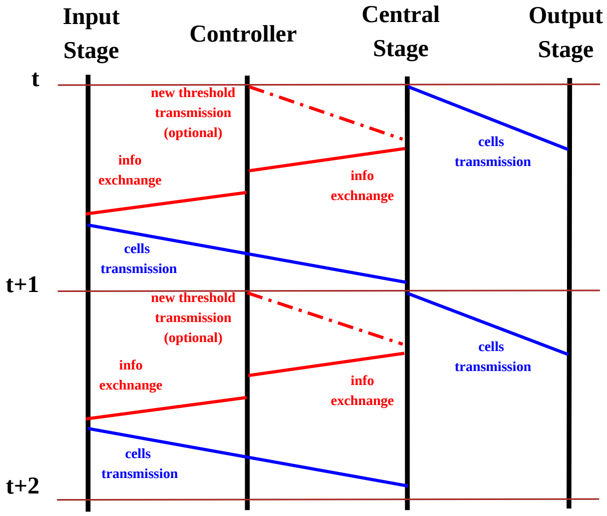


Figure 6.2: Timing diagram

The timing diagram in Figure 6.2 is representing a set of consecutive operations performed in the switch during a time slot. At the beginning of a time slot transmission of cells from central stage buffers to the currently interconnected outputs is done. During this time the controller is capable to set a new threshold value. It is done by transmission of $\log b$ bits of information, where b is the physical length in cells of a VOQ (all VOQ buffers have the same size). As soon as forwarding from the CSSs to the outputs is done, CSSs check current occupancy of the virtual output queues and compare it with the prescribed threshold. Based on the comparison, each CSS creates a vector of N elements, each element of which is keeping one bit of information representing occupancy (congestion) status. If the current VOQ occupancy is greater than the threshold value, the bit is set to 1, otherwise it is set to 0. When occupancy vectors are formed, they are immediately transmitted from CCSs to the controller. The controller forms a decision matrix in a way that each arriving vector of N elements is placed as a column of the matrix. The decision is made on the destination basis in such a way that each row of the matrix is processed. If in a row of N elements at least one 1-bit exists (logical OR is applied), than all packets destined to that output are considered to be blocked for transmission. Otherwise if all elements of a row are 0-bits, transmission to this output is possible. Based on a simple logical OR operation, the final decision vector is created as it is shown in Figure 6.3. Finally, N copies of the decision vector (N bits) are distributed to inputs. Based on the arrived final decision vector and availability of arrived packets, inputs either transmit packets to the next stage or drop them immediately (if a packet is already in transmission, no action is performed).

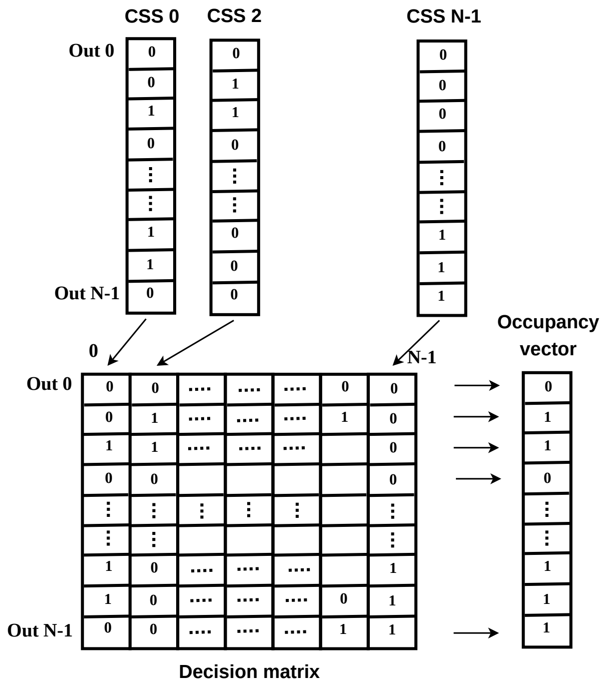


Figure 6.3: Information processing and exchange at the centralized controller

6.1.3 LB switch overheads and scalability

The traditional LB switch presented in [16,40] is considered to be a highly scalable solution in comparison to a crossbar switch, driven by a stable matching algorithm, if relaxed assumptions (central stage buffers are infinite, packets are of the same size and admissible traffic arrivals) are applied. As a result the LB switch was able to provide high throughput and have zero information exchange implemented (each stage made self decisions). In contrast, it was shown in [3, 69] that if more realistic scenario is applied, the traditional switch is not able to provide high throughput due to significant internal packet loss.

To control the amount of packet loss inside the LB switch the following information should be known: 1) input packet arrivals during a time slot and 2) the occupancy of all the VOQs in each CSS of the system. Since each element of the traditional LB switch is independent from the others, it is impossible to evaluate a potential central stage packet loss based only on the existing information and current switch configuration. In order to improve the overall throughput of the switch, the additional information exchange can be implemented between the stages (giving non-zero communication and computation overheads). These modifications, in their turn will make an impact on the scalability properties of the system. Therefore, with regard to more realistic assumptions, *the tradeoff between the system's scalability properties and throughput characteristics exists.*

The protocol used for minimization of packet loss described in this papers can be implemented both 1) using distributed information exchange and 2) centralized information exchange, e.g. as it is currently implemented. Lets compare the overheads which each of the solutions can produce during a time slot in order to motivate the choice of centralized controller.

The distributed scheme. Lets assume that all elements of the LB switch perform independent transmission decisions. In order to set a buffering threshold at all CSSs it is enough to send a request from a single input to all N CSSs. As soon as the threshold set, each CSS is creating an occupancy vector, composed of N bits, for the current time slot and distribute this information between all N inputs. In total, during a time slot, each CSS should send N vectors of N bits, making it N^2 vectors of N bits for the whole system. Based on the information arrived from all CSSs each input (in total system will have N decision matrices) is creating a decision matrix and performing logical comparison of bits. As a result, the transmission decision for the currently arrived packet is performed.

The centralized management. The protocol realized in this section is utilizing a centralized management which introduces extra wiring costs to the system since it utilizes N^2 detached links for information exchange. Considering the fact that only a centralized controller is performing the information exchange, we get $2N$ vectors of N bits as a total communication overhead in the system. Moreover, due to a simple bit-by-bit comparison (logical OR) performed in decision matrix of the controller the communication overhead is

Management	Distributed	Centralized
Communication overhead	N^2 vectors of N bits	N vectors of N bits
Computation overhead	$N * N^2$ bits to compare	N^2 bits to compare
Additional wiring	0 links	N^2 links

Table 6.1: Total system overheads for various implementation schemes

negligible and is constant in time. As a result, in terms of total overheads, the realization of the controller with centralized management is less complicated.

6.1.4 The Packet Loss Analysis

In [3] the authors gave an approximate model of the LB switches with identical input process assumption. The main contribution of that work is the scalable model for central stage packet loss probability of different size packets.

The main contribution of the present section is the modification of the working mechanism of the switch such that the packet loss probability is kept on the minimal level using the protocol given in Section 6.1.1. This improvement is achieved by the modification of the packet acceptance policy at the central stage of the switch. According to the new policy the model of the switch should also be changed. The model description throughout this section highly depends on the notations, the procedures and all the knowledge of [3].

In [3] the model of the switch is given based on the simplified model of the inputs. All the inputs are assumed to be identical and their two state (ON/OFF) discrete time Markov chain model is the basic building block of the transient DTMC model of the life cycle of a tagged packet passing through the switch. Here we dwell on the main differences of the original model without and the new model with the introduction of the buffering threshold.

Using the notations of [3] the differences are

- how the cell level model ($\mathbb{P}$) and
- how the initial distribution $\left(\boldsymbol{\pi}^{\mathcal{N}}\right)$, the state transition probability $\left(\mathbb{P}^{\mathcal{R}}\right)$ and the loss vectors ($\boldsymbol{\ell}$ and $\mathbf{s}$) of the packet level model

are built.

The packet acceptance threshold (t) is defined as the distance in cells evaluated from the beginning of a virtual output queue at the central stage buffers. As a beginning position of the queue we assume the point where transmission of a cell to the *next stage* is performed.

When the queue length of the observed VOQ is above T the arrival processes at the inputs are forced to be OFF. From the mathematical point of view this means that the parameter of the geometric distributed length of the idle period tends to be 0, i.e.,

$$\mathbb{P}^{(\mathrm{th})} = \left(\begin{array}{l} \mathbf{B} \quad \mathbf{L} \quad \mathbf{F}_1 \quad \mathbf{F}_2 \quad 0 \quad \dots\dots\dots\dots\dots\dots\dots\dots \\ \mathbf{B} \quad \mathbf{L} \quad \mathbf{F}_1 \quad \mathbf{F}_2 \quad 0 \quad \dots\dots\dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \dots \quad 0 \quad \mathbf{B} \quad \mathbf{L} \quad \mathbf{F}_1 \quad \mathbf{F}_2 \quad 0 \quad \dots\dots\dots\dots \\ \hline \dots\dots\dots \quad 0 \quad \mathbf{B}^{(\mathrm{th})} \quad \mathbf{L}^{(\mathrm{th})} \quad \mathbf{F}_1^{(\mathrm{th})} \quad \mathbf{F}_2^{(\mathrm{th})} \quad 0 \quad \dots \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots \quad 0 \quad \mathbf{B}^{(\mathrm{th})} \quad \mathbf{L}^{(\mathrm{th})} \quad \mathbf{F}_1^{(\mathrm{th})} \quad \mathbf{F}_2^{(\mathrm{th})} \\ \dots\dots\dots\dots\dots\dots\dots \quad 0 \quad \mathbf{B}^{(\mathrm{th})} \quad \mathbf{L}^{(\mathrm{th})} \quad \mathbf{F}_1^{(\mathrm{th})\prime} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \quad 0 \quad \mathbf{B}^{(\mathrm{th})} \quad \mathbf{L}^{(\mathrm{th})\prime} \end{array}\right) \tag{6.1}$$

$\lim \hat{q} = 0$. From the modeling point of view $\lim \hat{q} = 0$ is equivalent to the drop of the packets present at the inputs. The substitution of $\lim \hat{q} = 0$ results in the modified version of the model. The main differences are given in the followings for the 3×3 switch as in [3].

6.1.5 Differences in the cell level model

The state transition probability matrix of the cell level model changes to (6.1), where superscript $^{(\mathrm{th})}$ emphasizes the changes caused by the threshold. The block level of (6.1) describes the VOQ queue length and accordingly a horizontal line between the Tth and $T+1$st denotes the threshold. The blocks “above” the line are built exactly in the same way as the “regular” blocks in Sections 3.1 and 3.2 in [3]. The changed blocks “below” the line are built with the substitution of infinite idle period ($\lim \hat{q} = 0$).

Accordingly the steady state solution of the cell level model also changes to solution of the linear system of equations

$$\boldsymbol{\pi}^{(\mathrm{th})}\mathbb{P}^{(\mathrm{th})} = \boldsymbol{\pi}^{(\mathrm{th})} \qquad\qquad \boldsymbol{\pi}^{(\mathrm{th})}\mathbf{h} = 1, \tag{6.2}$$

where the notation $\mathbf{h}$ is introduced for the appropriate size column vector of ones.

The probability of dropping a packet at the input is

$$p_i = \sum_{i=B-(N+1)T+1}^{B} \pi_i^{(\mathrm{th})}, \tag{6.3}$$

i.e., the probability that the queue length of the system is above the threshold.

6.1.6 The packet level model

The packet level model is a transient DTMC with two absorbing states modeling the life cycle of the tagged packet. The introduced changes affects the state transition probability matrix of the transient part as well as the absorption vectors. Here again the blocks “above” the threshold are built in the way presented in [3] and the blocks “below” the line are created with the substitution of $\lim \hat{q} = 0$. The state transition probability matrix of the transient part and the probabilities of absorbing in the state responsible for a packet loss are

$$\mathbb{P}^{(\mathrm{th})^{\mathcal{R}}} = \left(\begin{array}{ccccc} \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} & 0 & \ldots\ldots \\ \multicolumn{5}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\ \ldots & 0 & \mathbf{L}^{\mathcal{R}} & \mathbf{F}_1^{\mathcal{R}} & \mathbf{F}_2^{\mathcal{R}} \\ \hline \multicolumn{2}{c}{\ldots\ldots\ldots} & 0 & \mathbf{L}^{(\mathrm{th})^{\mathcal{R}}} & \mathbf{F}_1^{(\mathrm{th})^{\mathcal{R}}} \\ \multicolumn{3}{c}{\ldots\ldots\ldots\ldots\ldots} & 0 & \mathbf{L}^{(\mathrm{th})^{\mathcal{R}}} \end{array}\right) \text{ and} \tag{6.4}$$

$$\boldsymbol{\ell}^{(\mathrm{th})} = \left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline \mathbf{F}_2^{(\mathrm{th})^{\mathcal{R}}}\mathbf{h} \\ \left(\mathbf{F}_1^{(\mathrm{th})^{\mathcal{R}}} + \mathbf{F}_2^{(\mathrm{th})^{\mathcal{R}}}\right)\mathbf{h} \end{array}\right) \tag{6.5}$$

respectively.

6.1.7 The initial distribution of the packet level model

The creation of the initial distribution is based on the steady state solution of the cell level model (6.2) and also on a block matrix. The upper part of the matrix is built up with blocks given in equation (20) in [3] and the lower part is zeroed by the $\lim \hat{q} = 0$ substitution

$$\mathbb{P}^{(\mathrm{th})^{\mathcal{N}}} = \left(\begin{array}{cccccc} \mathbf{L}^{\mathcal{N}} & \mathbf{F}_1^{\mathcal{N}} & \mathbf{F}_2^{\mathcal{N}} & 0 & \multicolumn{2}{c}{\ldots\ldots\ldots} \\ \mathbf{L}^{\mathcal{N}} & \mathbf{F}_1^{\mathcal{N}} & \mathbf{F}_2^{\mathcal{N}} & 0 & \multicolumn{2}{c}{\ldots\ldots\ldots} \\ \multicolumn{6}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\ \ldots\; 0 & \mathbf{L}^{\mathcal{N}} & \mathbf{F}_1^{\mathcal{N}} & \mathbf{F}_2^{\mathcal{N}} & 0 & . \\ \hline 0 & \multicolumn{4}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots} & 0 \end{array}\right). \tag{6.6}$$

Then the unnormalized initial distribution of the transient DTMC modeling the system on the packet level is given as

$$\boldsymbol{\pi}_u^{(\mathrm{th})^{\mathcal{N}}} = \boldsymbol{\pi}^{(\mathrm{th})}\mathbb{P}^{(\mathrm{th})^{\mathcal{N}}}$$

and it is normalized as

$$\boldsymbol{\pi}^{(\mathrm{th})^{\mathcal{N}}} = \frac{\boldsymbol{\pi}_u^{(\mathrm{th})^{\mathcal{N}}}}{\boldsymbol{\pi}_u^{(\mathrm{th})^{\mathcal{N}}}\mathbf{h}}. \tag{6.7}$$

Figure	6.4	6.5	6.6	6.7
name	$p_{\text{I-CS}}(T)$ versus T			
N	$4, \ldots, 12$	$4, \ldots, 18$	$4, \ldots, 40$	4
B	30	20	50	15
T	$1, \ldots, 30$	$1, \ldots, 20$	$0, 50$	$1, \ldots, 15$
$\hat{p}$	$\frac{1}{50}$	$\frac{1}{20}$	$\frac{1}{40}$	$\frac{1}{20}, \ldots, \frac{1}{50}$
$\hat{q}$	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{9}{10}$
$\hat{t}$	$\frac{1}{N}$			

Table 6.2: Parameters used for the numerical studies

6.1.8 The minimal loss probability of the system

Using the initial distribution (6.7), the state transition probability matrix of the transient part and the loss vector (6.5) the loss probability due to the finite central stage buffer capacity is given as the function of the threshold as

$$p_\ell(T) = \boldsymbol{\pi}^{(\mathrm{th})^{\mathcal{N}}} \left(\mathbf{I} - \mathbb{P}^{(\mathrm{th})^{\mathcal{R}}}\right)^{-1} \boldsymbol{\ell}^{(\mathrm{th})}. \tag{6.8}$$

Then using (6.3) and (6.8) the T-dependent joint I-CS loss probability is

$$p_{\text{I-CS}}(T) = p_i + (1 - p_i) p_\ell(T), \tag{6.9}$$

the probability of dropping a packet at the input or if it is not dropped at the input it is dropped at the CS due to buffer overflow.

6.1.9 Experimental results

In this section we study the joint I-CS packet loss of the switch as a function of the CSSs' buffering threshold by the consecutive execution of (6.9) for all $T \in [0, B]$. We also present the results of various simulations that we have performed to verify our theoretical model presented in Section 6.1.4.

In our experiments we assume that all inputs of the switch have identical arrival processes. The length of the variable size packets arriving to the input stage is considered to be geometric distributed (with parameter $\hat{p}$). The packets interarrival periods are also geometric distributed (with parameter $\hat{q}$). The arriving packet at an input is directed to a particular output with probability $\hat{t}$. Please also note that all the evaluations of the joint I-CS packet loss were performed for the input 1 - VOQ$_{00}$ - output 0 traversing path.

The results of the first experiment are presented in Figure 6.4, 6.5 and 6.6. The figures show dependency of the joint I-CS packet loss for various threshold values and switch sizes. The parameters used for packet loss evaluation are listed in Table 6.2. Based the observed results several conclusions can be made. First, while the switch threshold is considered to be around 0, the input packet loss has the main impact on the joint packet loss. Basically

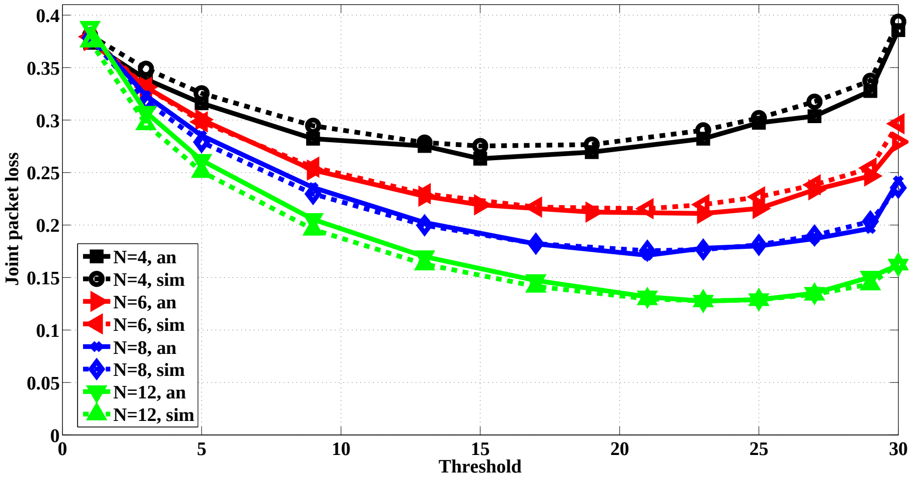


Figure 6.4: Analytical and simulations representation of joint packet loss probability versus threshold

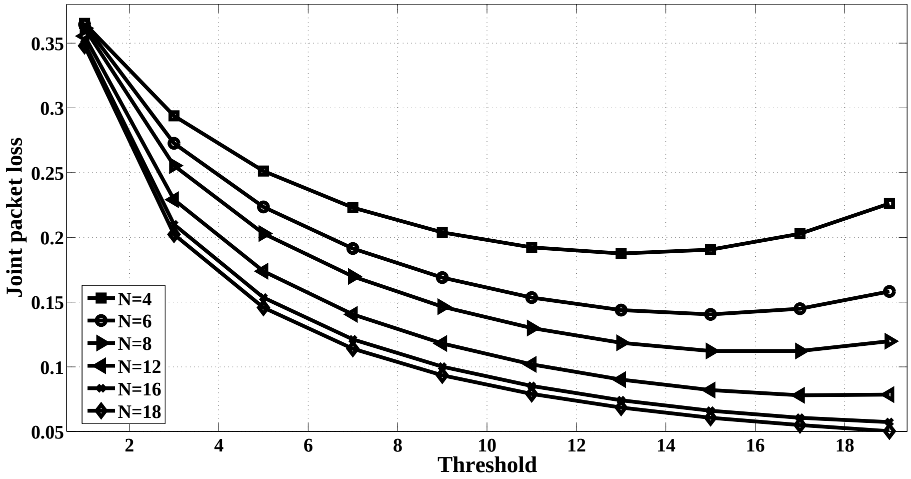


Figure 6.5: Dependence of joint packet loss probability versus threshold for various switch sizes

the protocol is dropping most of the packets arriving to the inputs since none of the central stage buffers is allowed to be used for the packet transmission. Indeed the loss value is almost independent of the switch size (see also curve $T = 0$ in Figure 6.6). On the other hand, when the threshold at a central stage is equal to the physical buffer size b the switch is operating in the traditional way (without protocol support) and the joint packet loss is composed only of the loss obtained due to the central stage buffers congestion. Finally, moving the threshold in between of these two extremes we obtained the minimum joint packet loss probability. Since the results were performed for the different switch sizes it is also possible to see how the packet loss minimum is moving (it is appearing closer to the actual VOQ size – b) when the switch size is increasing.

Figure 6.5 shows one other interesting feature of the protocol. The minimum moves towards b to the upper boundary of the $[0, b]$ interval with the enlargement of the switch.

The explanation of such kind of behavior is the following. The introduction of the threshold aims to reduce the wasted capacity and if the loss probability at the central stage is large the introduction of $T < b$ delivers the goods. The higher the loss probability the lower the T results in the minimal joint I-CS loss. On the other hand in [3, 4] the authors concluded that the growth of the switch size results in larger system capacity and accordingly lower central stage packet loss probability. These two effects causes the point of the minimum to move towards b with the increase of the switch size. From a given point the central stage packet loss probability decreases as low as the introduction of $T < b$ practically reduce the packet acceptance without decreasing the number of pointlessly transmitted number of cells and consequently without decreasing the wasted capacity. Moreover with the reduction of T one introduce a grown input loss resulting in grown joined I-CS loss.

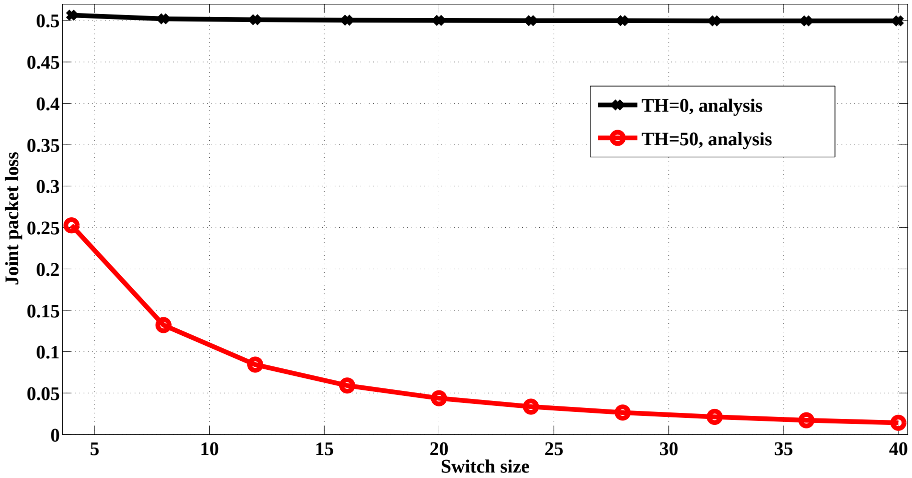


Figure 6.6: Joint packet loss probability versus switch size

In the next experiment in Figure 6.7 we examine the joint I-CS packet loss probability evaluated by means of mathematical model and simulations in response to the various threshold sets. However, in this experiment we focus on the behavior of the system when various types of traffic matrices appear at the inputs. In particular, we modify the average packet sizes which are running through the switch. The exact set of parameters related to the experiment is presented in Table 6.2. According to the obtained results, and also to our expectations, with the growth of the average packet size the joint packet loss of the system also increases. Figure 6.7 reflects to the fact that not only the system capacity plays significant role in the central stage loss probability but the average packet size too. If the average packet size is larger compared to the switch size the CS packet loss probability also increases. Similarly to the previous experiment the higher the CS packet loss probability the lower the T results in the minimal joint I-CS loss.

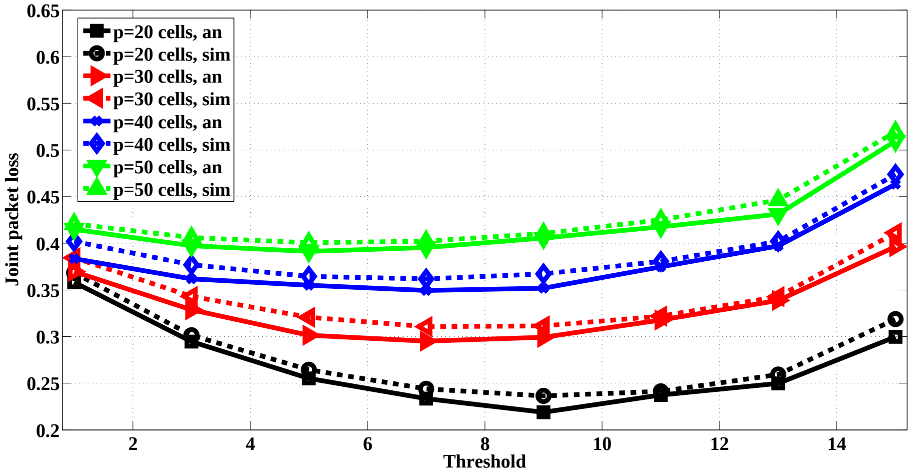


Figure 6.7: Joint packet loss probability as a function of protocol threshold

6.1.10 Summary

In this section we presented the service protocol which allows to calculate and configure the LB switch in order to obtain the minimal packet loss probability for both input and central stage buffers. Using the protocol one can reduce the capacity which is wasted by the LB switch, the computation overheads and the complexity of the equipment used for a reassembly unit implementation. Additionally, while performing the computational studies we have proposed several solutions to minimize the joint packet loss probability. We have also given explanations to two interesting phenomena, e.g. how the switch size and the load of the switch affects the threshold value at which the minimal joint I-CS loss probability is gained. Finally, we showed that the low implementation complexity of the proposed protocol allows the system to scale up for a large number of ports and buffer sizes.

6.2 NoLoss Load-Balancing Switch

In the previous section we introduced a service protocol for the LB switch which allows to minimize internal packet loss at the cental stage buffers. In contrast to the traditional LB switch architecture where all the arriving traffic is traversing through the switch without any knowledge weather it will be internally dropped, the service protocol is capable to perform decision on the amount of packets dropped at the input and central stage buffers. Basically, the management controller acts as a traffic gateway by setting a virtual sliding threshold at the central stage buffers. The proposed algorithm allows to drop the entire packet at the input FIFO buffers and by this reason reduce potential congestion at the output re-sequencing and reassembly units. Unfortunately, even operating with the minimal packet loss, the service protocol can have a non-zero probability to drop a cell at a central stage buffer. All these create a potential risk to have a congestion or a high packet loss at the output RRU buffers due to incomplete packets. Due to impossibility of withdrawal of all the cells of the broken packet, after their distribution to the central stage, one of the possible solutions can be a development of an algorithm capable to predict possible packet loss and reset the whole packet directly at the input.

In this section we show the modified version of the service protocol which is completely avoiding packet loss at the central stage buffers. To control the amount of traffic accessed to the central stage buffers of the switch the centralized controller in between of the input and central stage is installed. The controller algorithm implements a sophisticated information exchange between the central stage buffers and inputs in order to perform the transmission decision for the arriving packet. In particular, within a time slot the novel scheme is collecting information about the current states of input and central stage queues. Based on gathered information the algorithm generates decision whether the packets arrived to ingress ports have a right to be transmitted further through the system. Note that all the arriving packets are kept in the input buffer queues while the transmission decision is made. The decision is formed based on the knowledge of the available space at central stage buffers and the length of the packets arrived to the inputs during this time slot. To avoid VOQs congestion, the controller emulates the normal switch's operation and checks whether these packets can fit to the corresponding VOQs if they would be transmitted. For this purpose, the controller emulates a packet transmission during a time period which corresponds to at least a packet size. For instance, if packet is composed of K cells, the controller will perform the emulation process during the period of K time slots only for this packet. Please also note that if there is a simultaneous arrival of several packets to the same output, the controller will consider the emulation duration in time slots to be equal to the maximum packet length (in cells) for the considered destination. If central stage buffers does not have enough space to place the entire packet inside the buffers, the controller does not allow this packet to be transmitted through the switch. As soon as the controller distributes the decision vector between the inputs the packets are either transmitted or immediately dropped.

The presence of the centralize controller is considerably increasing the system's computational and communication overheads. On the other hand such a switch can guarantee the stability of operation and complete elimination of defects at the output RRU.

The rest of this section is organized as follows. First in section 6.2.1 we present all the details regarding the operational principles of the NoLoss LB switch. Then, the experimental results related to the considered architecture will be given in Section 6.2.2. Finally, the concluding remarks will be presented in Section 6.2.3.

6.2.1 The architecture

The NoLoss (NL) LB switch architecture is represented in Figure 6.8. Similarly to the traditional LB switch [17,18,65] the NL switch is composed of three stages. In between of three stages two interconnection fabrics are placed. Usually the interconnection pattern is realized in crossbar switches, but, due to the fact that the switch runs through the similar interconnection sequence every cycle (5.1), only a limited number of interconnections is used. It is also assumed that each of the two crossbars is synchronized with the rest of the system and only one cell can be transmitted during a time slot. The main novelty introduced in the NL switch is the implementation of the centralize controller which is placed in between of the input and central stages and interconnected with all the stage elements. When the controller is in the off state, the NL switch is operating as a traditional LB switch, otherwise, the NL switch is controlling the amount of traffic sent through the switch and partially drops packets in the inputs.

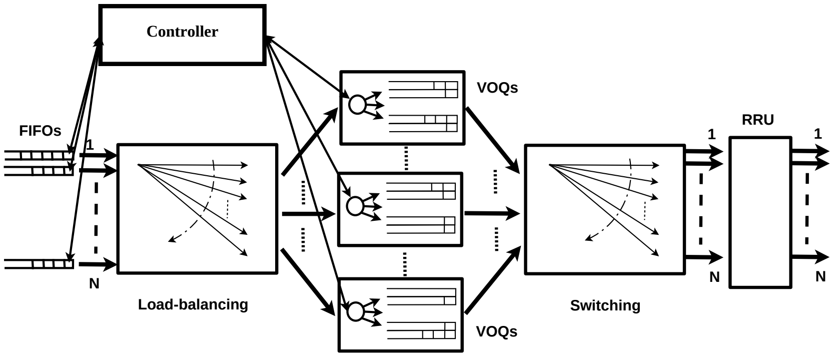


Figure 6.8: The NoLoss LB switching architecture

To get the clear understanding how the prediction algorithm works, lets reopen the main reasons of the central stage packet loss of the traditional switch. In the LB switch, the arriving traffic is immediately spread between the central stage VOQs without any respect whether this queues are congested or not. A cell drop occurs faster if the number of transmitting inputs with the same destination is large. In order to have a notion on a possible packet loss, two main characteristics of the switch should be taken into account for every time slot (as it is already presented in previous section). They are: 1) the occupancy status of all central stage VOQs and 2) the length and destination of each packet arrived to the inputs during a time slot. These parameters can potentially give an information about the available space at the central stage buffers, e.g. about the traffic which can be accepted, and knowledge about the traffic which is going to be send. The comparison between these two measures gives the necessary transmission decision. Indeed, this idea was implemented in the NL switch controller.

To summarize the main operation steps of the NoLoss switch, we provide the following sequence of operations:

Step 1. Information exchange phase (CSSs – Controller)

Step 2. Transmission decision evaluation phase

Step 3. Drop vector distribution (Controller - Inputs)

Step 4. Traffic forwarding phase

Controller implementation

Operation of the controller can be separated into two main steps: 1) an information collecting and 2) a decision making phase. In this section we describe only the first phase.

At the beginning of a time slot, both inputs and central stage buffers are sending information to the controller. In particular, each input is providing the information about packets arrival (if any), the length of each packet and their destination. More precisely, if a packet with length l_i cells has arrived to input i, the input suppose to send immediately $log(l_i)$ bits of information to the controller.

The information about current packet arrivals is updated in two-dimensional matrix A of the controller. At the beginning of a time slot, all the elements of matrix A are reset to zeros. Each element of matrix $A[i, j]$ can store two measures of a packet (positioned at input i and destined to output j) – the packet length (in cells) and its identification number. The elements of the matrix are updated as soon as new packets arrive and the information about packet arrivals is kept in matrix A until a transmission decision is made. The entity of the matrix is reset to zero (a packet receives drop flag) if the packet cannot fit the available space in the central stage buffers. On the other hand, if the packet can be transmitted further the values of the matrix A are transferred to the specific two-dimensional matrix $S[i, j]$ which is called a system status matrix.

The status matrix S has the same structure and functionality as matrix A, since it is keeping the length and id of packets which are in the transmission process. Unlike matrix A, which is reset to zero every time slot, matrix S keeps information about packets in transmission all the time. In particular, each element of matrix S is showing the amount of cells which remains to be transmitted from as input stage buffer to specific VOQs. Please also note that matrix S is updated every time slot (either incremented or decremented), and these updates are related to either the cell-by-cell transmission of packets from an input to the central stage buffers or when a new packet is set for forwarding. Finally, the controller implements a central stage status matrix $C[i, j]$ which represents the occupancy status of all the central stage VOQs.

To simplify the controller operation while collecting central stage buffers occupancy information, it is enough to have knowledge only of *the maximum occupancy* of the VOQs placing cells *with the same destination*. Since cells with the same output are forwarded through the same sequence of VOQs (for instance cells packet form in 1 to out 3 will be transmitted trough $VOQ_{0,3}$,$VOQ_{1,3}$,. . .,$VOQ_{N,3}$) the maximum occupancy value is easy to obtain when matrix C is formed. Indeed, this information is stored in the *maximal occupancy vector (mov)*. Since the main goal of the controller is to know the available capacity at the central stage with regard to each output, the *central stage space vector (cssv)* is formed. Each element is simply $cssv_i = (B - mov_i) * N$, where B is a physical VOQ size and N is the number of ports in the system.

The controller is involving decision making algorithm only when a new packet(s) arrives to an input(s). Otherwise, the controller remains in the steady state, updating only the information about the packets which are currently in transmission (e.g. matrix C).

To collect all the necessary information the controller goes through the following steps:

1. checks matrix A for new packet arrivals,

2. updates matrix C with the current occupancy states of all VOQs, creating mov and $cssv$ vectors,

3. based on the information available in matrices S,A and vector $cssv$ make decision on every input packet,

4. based on the decision obtained, the controller sends bits of the transmission flag to the specific inputs.

Example of 4×4 NL switch information gathering. 4×4 NoLoss LB switch is represented in Figure 6.9. Lets assume that at the beginning of a time slot 4, four packets arrive to switch inputs. In particular, three packets arrive to inputs $0, 1, 2$ and are directed to output 0, the remaining packet placed at input 3 is directed to output 1. Since there are packet arrivals to switch's inputs, the controller should update the corresponding information in matrix A. Since inputs 0 and 2 have packets composed of 5 data cells, input 1 has a packet segmented into 6 cells, and, input 3 has a packet composed of 4 cells, therefore, matrices A and S will look like:

$$A = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}, S = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6.10}$$

In the similar manner, each central stage buffer set (CSS) (composed of N VOQs) is informing the controller about the occupancy of each VOQ. For this purpose a vector of N elements is sent from a CSS to controller, each entity of the vector keeps $logB$ bits of information. Lets assume that the maximum physical size of each VOQ is equal to $B = 10$ cells. In this case, matrix C and vectors mov and $cssv$ are created in a way (and

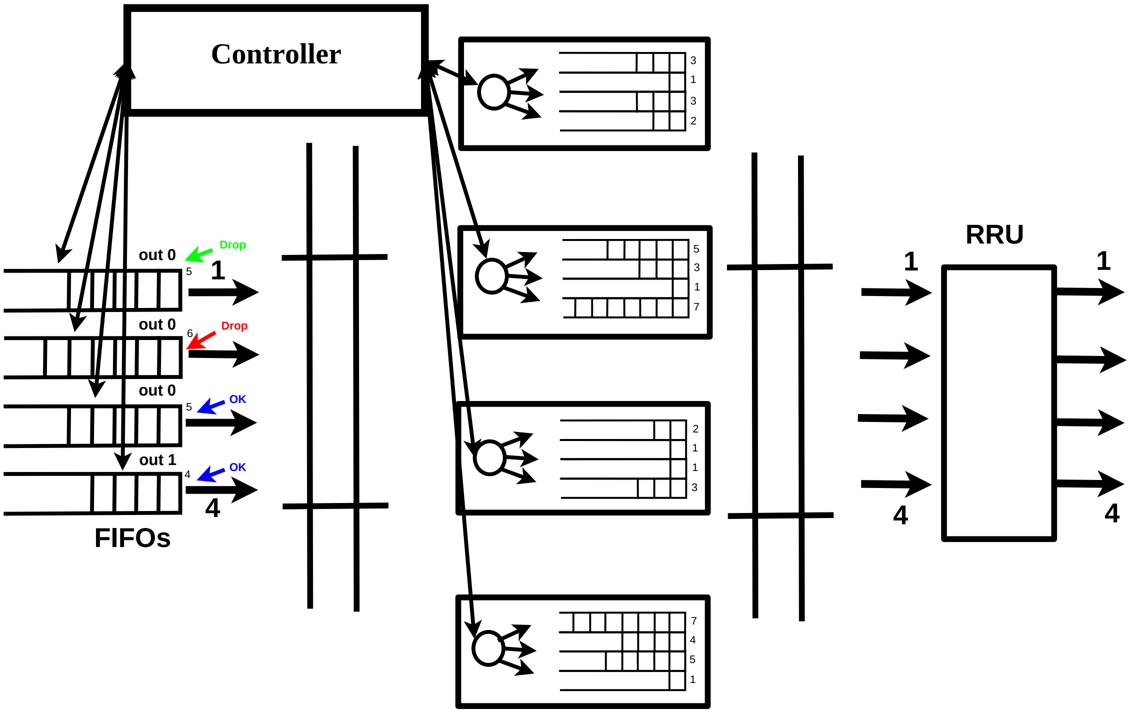


Figure 6.9: Example of 4 × 4 NoLoss LB switch

in correspondence to Figure 6.9):

$$C = \begin{pmatrix} 3 & 1 & 3 & 2 \\ 5 & 3 & 1 & 7 \\ 2 & 1 & 1 & 3 \\ 7 & 4 & 5 & 1 \end{pmatrix}, \quad mov = \begin{pmatrix} 7 & 4 & 5 & 7 \end{pmatrix}, \quad cssv = \begin{pmatrix} 12 & 24 & 20 & 12 \end{pmatrix}. \qquad (6.11)$$

After the current state information from inputs and central stage buffers is transmitted the controller enters into the decision making phase.

Packet loss prediction algorithm

In order to decide which input have a right to forward packets through the switch, the controller should make a decision based on the information obtained form stages and processed by a prediction algorithm. A basic operation idea performed by the algorithm presumes comparison of the amount of input traffic and buffering space available in the central stage.

The analysis is derived on a traffic destination basis (evaluations are done with respect to traffic destination) and explores the knowledge of matrices A,S and C obtained from the previous step. In order give clear understanding how the prediction algorithm works, lets present a simple example referring to Figure 6.9.

Transmission decision. Lets define an amount of traffic which is suppose to be transmitted from all inputs to each output. Based on the example presented in Figure 6.9, inputs 0,1,2 are transmitting packets to output 0 with total amount of 16 cells. Similarly, output 1 suppose to receive 4 cells (from input 1) from the system. The other outputs are not receiving any traffic. Lets define vector itr_i as a vector which keeps amounts of traffic in cells to each output. Based on the occupancy information given in matrix C and, more importantly, in vector $cssv$ it is easy to evaluate the amount of traffic which can be accepted by VOQs with respect to a specific output. For instance, central stage VOQs keeping traffic for output 0 have availability to accept 12 cells. VOQs keeping packets to output 1, in their turn, can accept 24 cells, etc. As soon as this information is gathered, the controller starts the comparative phase.

The algorithm starts from output 0. As it is seen from itr_0 and $cssv_0$ 16 cells cannot be accepted by 12 available buffer places, therefore not all inputs are allowed transmit. Moreover, some extra decision should be made. To simplify prediction algorithm, the controller chose randomly (or in a round-robin manner) an input which suppose to drop a packet. In our example (Figure 6.9) the packet from input 1 was chosen for removal (marked by red color in Figure 6.9). As a consequence a drop flag (which is simply a bit equal to "1", and erasing element $A_{1,0}$ from matrix A) is assigned to the packet. Since the amount of traffic scheduled for transmission to output 0 has changed, the controller start from the beginning a comparative phase for all remaining inputs transmitting to $output0$. Thus, the algorithm compares itr_0 and $cssv_0$, or 10 cells to forward and 12 cells to accept, which at the first sight can be placed at corresponding central stage buffers. However, it can happen that although the amount of cells to be transmitted can fit the number of cells to be accepted, during this time slot, these measures are not be acceptable during posterior time slots (itr_0 can be larger than $cssv_0$!). In order to avoid inconsistences, the algorithm emulates a cell-by-cell transmission of packets during a period of time which will correspond to complete transition of packets to the central stage (unless the conflict is obtained).

During the emulation process, the $cssv$ vector is modified according to a worst case scenario. According to formula (5.1) the occupancy value of VOQs for the considered output increases during $N-1$ time slots and remains on the same level during the last time slot. Due to the round-robin interconnection, all the VOQs are behaving in the similar way. In our example (presented in Figure 6.9), VOQs at central stage will be increasing during 3 consecutive time slots and will remain on the same level during the $4-th$ time slot.

Based on the description above, lets emulate packets transmission with destination to output 0 (presented in Figure 6.9). Going back to the step where itr_0 was compared to $cssv_0$ (10 with 12), the transmission decision was granted for the first time slot. Making emulation of packet transmission for the second time slot, two cells from both inputs are considered to be sent e.g. $cssv_0 = 8$. Due to the worst case scenario one of the

$VOQ_{0,0}$,$VOQ_{1,0}$,$VOQ_{2,0}$,$VOQ_{3,0}$ will increase by 1 so $cssv_0 = 4 * (10 - 8) = 8$, which results in 8 = 8 - transmission is granted. However, during a third "virtual" time slot we obtain that $itr_0 = 6$ and $cssv_0 = 4$ which leads to a contradiction. Therefore, one of the packets going to output 0 cannot be considered for further transmission. As a consequence the controller assigns randomly the drop flag to packet of input 1 (marked by green color in Figure 6.9).

Since there is still one packet destined to output 0, the controller recursively goes through the prediction algorithm represented above once again. Performed evaluations show that packet from input 2 can be forwarded through the NL switch. The estimations performed for traffic with a final destination to output 1 showed that only one packet out of three can be forwarded through the central stage. Similarly, the controller estimates congestion possibility for remaining traffic (e.g. traffic with other destinations). Please also note, that if a new packet arrives during a time slot when other packets are in the process of transmission to the same output, the controller goes through the evaluation procedure described above while involving matrix S, e.g. taking into account traffic which is in transmission. For the simplicity we assume that the traffic in transmission to some specific output is kept in vector $tcur$ and can be easily derived from matrix S.

In short the presented congestion avoidance algorithm is running though the following steps (within a time slot):

As soon as the transmission decision is made, the controller distributes N copies of drop vector to all inputs. Based on the final decision, the actual transmission of packets is performed.

Overheads of the algorithm

Due to the fact that the centralize controller (and not the distributed one) is managing the system's traffic, the algorithm may not be scalable with enlargement of switch size. In this section we present all the overheads involved into data collecting and decision making phases of the controller.

Communication overhead. In order to evaluate the amount of communication overhead, lets trace the sequence of information exchange operations inside the NL switch. Lets refer to Figure 6.10. At the beginning of a time slot inputs start sending information about current packets arrivals (if they have any) to the controller. If there is no arrival, or packet is in the process of transmission, inputs do not send any information to the controller. Assume that input has a new packet of length l_i. In this case the input sends $log_2(l_{max})$ bits of information to the controller, where l_{max} is the maximum packet size which can be present in our system. On the other hand, each CSS sends a vector of N elements to the controller, each element of the vector is represented by $log_2(B)$ bits. The following information is used for transmission decision phase. As soon as a

Algorithm 3 Congestion avoidance algorithm

```
Require: Update $\mathbf{A}, \mathbf{C}$ involving Inputs and CSSs
Require: reset local timer $t$
while ($out < N$) do
    Evaluate $cssv(out)$, $itr(out)$, $tcur(out)$, $tmp(out) = tcur(out) + itr(out)$
    Create $cssv'(out) = cssv(out)$, $itr'(out) = itr(out)$
    Create $tcur'(out) = tcur(out)$, $tmp'(out) = tmp(out)$
    if ($itr(out) \neq 0$) then
        while ($input\ not\ dropped$ and $itr(out) \neq 0$) do
            Update $itr'(out)$, $tcur'(out)$ and $tmp'(out)$
            Emulate $cssv'(out)$
            Compare $tmp'(out)$ and $cssv'(out)$
            Increment $t$,
            if (t = N) then
                $t = 0$
            end if
            if ($tmp'(out) < cssv'(out)$) then
                Chose one of $itr'(out)$ and assign a drop flag
                Update $itr(out)$ and $A$
            else $\{tmp'(out) > cssv'(out)\}$
                Update $tcur(out)$ with $itr(out)$
                Increment $out$
            end if
        end while
    end if
end while
return drop flag vector
```

drop vector is created, it is distributed bit-by-bit for all inputs (number of bits in drop vector corresponds to number of ports N). Every bit informs corresponding input about packet drop or forwarding. To summarize, the total communication overhead inside the system during a time slot is $Nlog_2(l_{max}) + N^2 log_2(B) + N$ bits (please remember that the controller operates only if new packets arrive).

Computational overhead. The decision making phase can be split into two main parts. The first phase is related to forming of matrices A and C which, indeed, takes a constant time to complete. The second phase is associated to evaluation of Algorithm 3. Lets examine this phase more in details while defining the worst case time for Algorithm 3 to converge. During a time slot the controller makes recursive emulation of packets transmission (in cell-by-cell basis) from multiple inputs to a single output. Therefore, the worst case scenario is the case when all switch inputs hold packets to the same destination. If we assume that the time to emulate a single cell transmission is equal to t_{em} (in seconds) than $N!$ checks in total should be done to create a drop vector. As a result, the computational time inside the controller will result in $t_{total} = t_{em} * l_{max} * N!$ seconds. Please also note that the controller's computational time has strong dependence

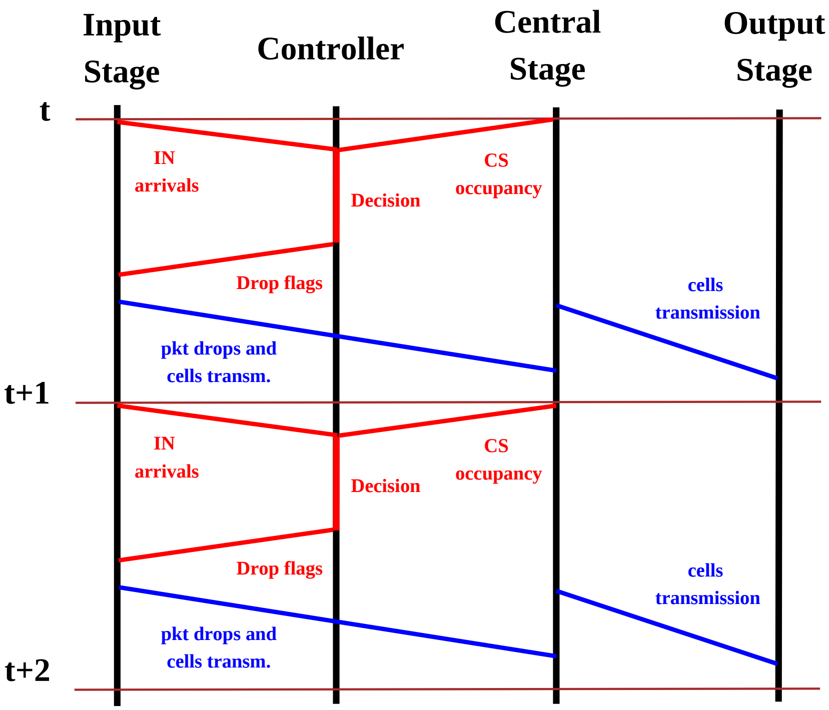


Figure 6.10: Sequence of controller operations during a time slot

on the switch size or maximum packet length grows. Therefore, the time slot duration should be adjusted in compliance with mentioned characteristics increase.

However, if larger packets are arriving to the system, the LB switch makes less requests to the controller to perform management actions (the controller is called only when a new packet arrives). As a result, the overall performance of the system grows as well.

6.2.2 Experimental results

In this section we will present a set of results related to the packet loss probability in the NoLoss LB switch. In order to perform all the evaluations the NL LB switch simulator was written. In comparison to the simulator which was used for numerical studies in previous sections, in our current version the controlling unit with congestion avoidance algorithm described in Section 6.2 was added. To set up experiments a set of restrictive assumptions was used. The same way as presented in previous sections, packet lengths and interpacket arrival periods in the system are geometrically distributed. Although the simulator is capable to support any complex traffic distribution, in particular patterns which are close analogy to the real network distributions [24,25,44,63], we apply geometric distribution to be able to compare the packet loss probabilities with and without external management.

To evaluate the joint packet loss probability inside the system, we analyzed both the amount of packets dropped at the input stage and number of packets discarded at central

Figure	6.11	6.12	6.13
N	16	$4, \ldots, 64$	16
B	$20, \ldots, 80$	$20, 40, 80$	50
p	$\frac{1}{30}, \frac{1}{40}$	$\frac{1}{50}$	$\frac{1}{10}, \frac{1}{20}, \frac{1}{30}, \frac{1}{40}, \frac{1}{50}$
q	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{4}$
t	$\frac{1}{N}$		

Table 6.3: Parameters used for the simulations

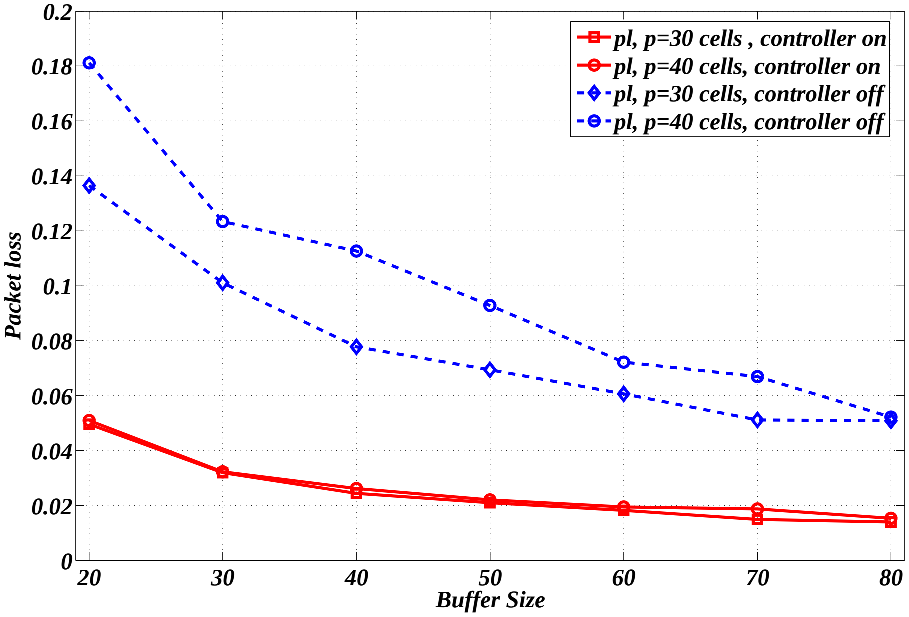


Figure 6.11: Dependence of joint packet loss as function of buffer size

stage buffers. It is noted that when the controller is in ON state, the joint packet loss probability is composed only from input packet loss. Otherwise, if the controller is in OFF state, the system starts experiencing the central stage packet loss as in the traditional set up. Figure 6.11 makes comparison between the packet loss amounts experienced in case when the controller is operating and when it is switched off. The parameters used for simulations are given in Table 6.3. As it is expected, the packet loss degrades with the central stage buffer grows. However, in case when controller is off, the joint packet loss degrades much faster (packets are lost only at central stage buffers). On the other hand, when controller is on, the joint packet loss is composed only of packets dropped at the input stage, and, therefore, it does not have strong dependency on the amount of buffering introduced at the central stage VOQs.

In general, the joint packet loss for controller in ON state is much lower than the corresponding loss in OFF state. The following behavior can be explained in the following way. When the LB switch is operating in the traditional "mode", a cell drop in a packet transferred through the central stage buffers does not trigger the whole packet removal.

In such a way an additional capacity is used by the cells of incomplete packets. On the other hand, the proposed management scheme removes completely ”problematic” packets at inputs which giving extra capacity at central stage buffers for packets which will arrive in the future.

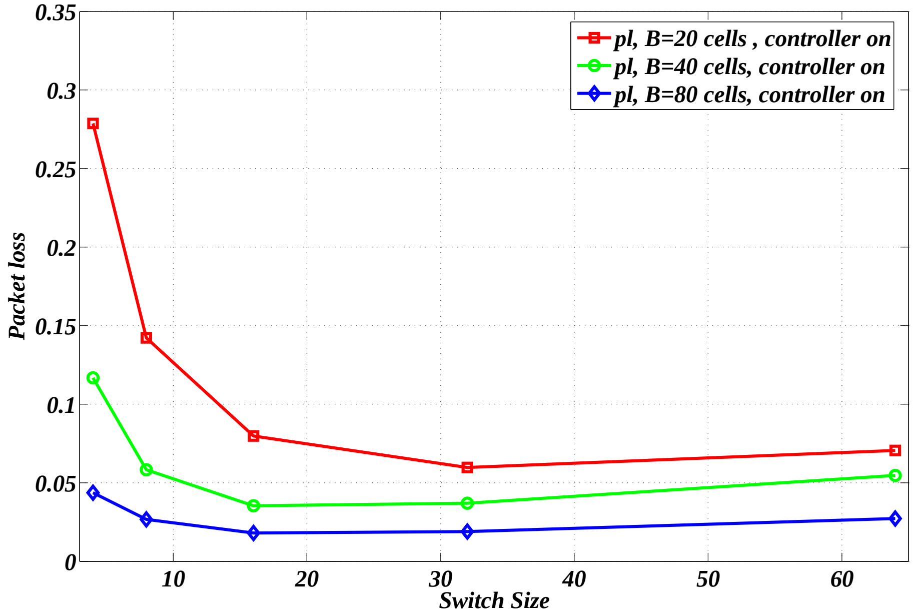


Figure 6.12: Joint packet loss dependence versus switch size

In Section 5.5.3 we showed that the reduction of the packet loss at central stage VOQs with enlargement of the switch size is dictated by the enlargement of the total buffering capacity. However it is not the main reason for such a behavior. The grows of the switch size has an impact on the probability matrix T in a way that the probability of packets transmission from all inputs to the same output (which has strong relation to the packet loss) is getting small.

In a new set of simulations we performed evaluations of the packet loss probability as a function of switch size. Figure 6.12 does not illustrate the constant drop in the packet loss probability as it happens with the traditional switch. Moreover, unexpectedly the joint packet loss probability starts growing after a certain value. The explanation of a such strange behavior is related to the decision making phase and, in particular, to the emulation of VOQs occupancy function. In order to reduce computational space and not to store all the occupancies of central stage VOQs, the management algorithm is taking the maximal occupancy value of VOQs storing cells with the same destination. As it is mentioned, during a decision making phase the controller emulates a cell-by-cell packet transmission while using the worst case scenario of VOQs evolution. With the enlargement of switch size (N) the virtual time cycle (in virtual time slots) is becoming longer. This basically means that for similar parameters of CS buffer size, average packet size and interarrival period, due to larger N every VOQ will be growing for a larger time interval.

Lets consider a simple example. If, for instance, switch size has changed from $N = 4$ to $N = 6$ than, during emulation process, in the first case each VOQ will be growing during 3 time slots (and remain with the same size during the 4-th time slot) and in the second case will be growing during 5 time slots (and remain with the same size during the 6-th time slot). Consequently, the same configuration (set of b,p,q,T) will lead to the lower probability for each next packet to be accepted.

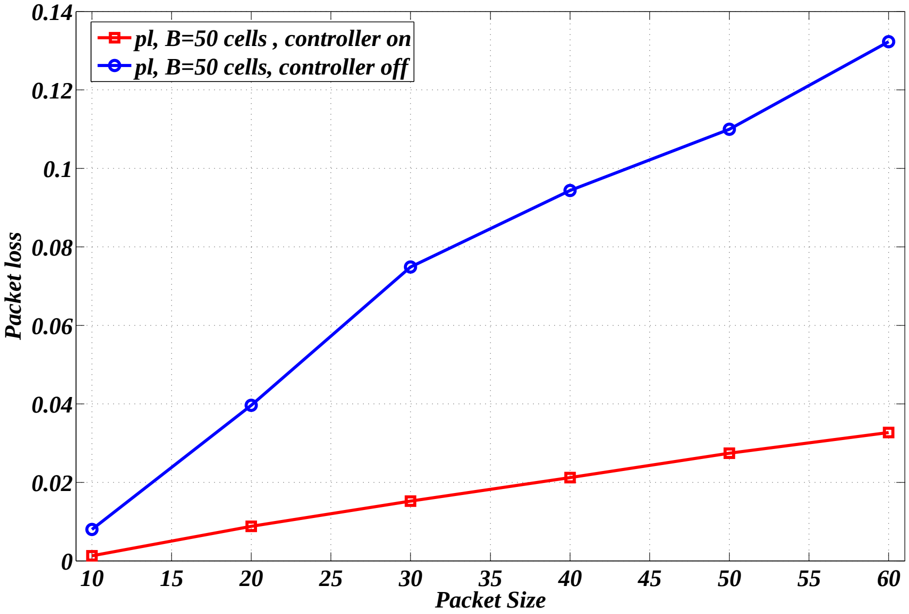


Figure 6.13: Joint packet loss as a function of packet size

Next, in Graph 6.13 we showed dependency of the packet loss enlargement as a function of the average packet length. In the traditional LB switch a packet length is considered to be a burst of cells going to the similar destination. Since all cells from an input are distributed to the central stage buffers according to the final destination, the larger the input burst is the more saturated will be corresponding VOQs after forwarding. Therefore, in the traditional system a packet loss is growing with the enlargement of the packet size. Due to the fact that in the current system the controller is performing decisions on the basis of the current central stage VOQs occupancies, the packet loss grows should be similar to the traditional system. Since in traditional LB switch, cells of incomplete packets are not removed from CSSs, the amount of joint packet loss with the novel scheme is considerably lower than in the standard LB switch (as shown in Figure 6.13).

6.2.3 Summary

In this section, the NoLoss LB switch was introduced. In order to completely avoid congestion of central stage buffers a novel management scheme with centralized control was proposed. In spite of the fact that NL switch can have considerable overheads to retrieve information from stages and perform traffic congestion control, the evaluations show that a total amount of packet loss in the NL switch is always lower than the corresponding packet loss of the traditional system. Moreover, due to the fact that packet drop at CS buffers can never happen, the NL loss switch can always guarantee the operational stability of the output re-sequencing and reassembly unit which, in its turn, makes easy to evaluate the amount of buffering capacity needed for reassembly process.

Chapter 7

Conclusion

The technological breakthrough over the last two decades makes impossible a human's life without computers and Internet. The modern Internet represents an extremely complex architecture which is continuously expanding in space and has extensive user demand. The possibility of powerful and mobile computing together with the continuous advent of novel real time services (like video and audio) and modes of access create new challenges for further evolution of the internet itself and its switching technologies.

The fact that current Internet routing systems consume an enormous amount of power [43], provoked development of routers with distributed control which, because of decentralized management, have lower need in electricity and associated infrastructure, e.g. power plants. Moreover, the simplicity of distributed scheduling allows to provide high switching scalability with relatively small cost.

The traditional switching architectures with distributed scheduler use graph matching techniques for interconnecting ingress with egress ports. In particular, a switching fabric is configured by iterative distribution of request tokens and their posterior granting, e.g. in a way a handshake protocol operates.

This dissertation is dedicated to the load-balancing switching architecture, which is, due to a simple distributed control with zero communication overheads, shows good performance characteristics, and, therefore, gives a prospective solution for the future internet routing development.

The initial set of papers attempted to review and resolve several key problematical issues of the switch. In particular the authors took into account the algorithms for resolution of mis-sequenced packet arrivals, fault tolerance and broadcasting possibilities of the LB switch. The initial set of papers on the topic was also representing some possible implementations of the switch in optic-electronic and all optical domain.

This thesis is focused on the analysis of the internal switch losses with novel assumptions applied to the incoming traffic and switch itself.

7.1 Summary

The load-balancing switching architecture firstly presented a decade ago, is considered to be a prospective architecture due to its simple control and high scalability. However, as it was shown through the thesis, the initial research presented on the topic did not cover all the negative aspects of the LB switching architecture when a set of realistic assumptions is applied. This thesis makes attempt to investigate the issue of the internal packet loss inside the switch, which can appear due to the central stage buffers overflow. The thesis can be divided into two main parts. The first part of the thesis presents mathematical models for evaluation of the internal packet loss inside the system both for fixed size data cells and for variable size packets. The obtained results show some interesting features of the switch as well as introduce a set of novel problems which can cause system's instability when internal packet loss is extremely high. The second part of the thesis presents solutions for minimization/avoidance of central stage packet loss while introducing additional control over the stages. These algorithms are greatly improving the overall stability of the system, however introducing extremely high input packet loss together with heavy communication and computation overheads.

To summarize, this study had uncovered a new important issue of the packet loss inside the load-balancing switch. The thesis presents effective algorithms for the evaluation of the amount of packet loss inside the system while using the assumptions close to practical. Additionally, the thesis provides several algorithms to minimize and avoid the packet loss inside the system while implementing complex management and evaluation procedures.

7.2 Future Work

Although the presented study delineate the significant results on the field of the load-balancing switching, and in particular in the packet loss analysis, there is a set of other open issues which was not resolved up to now. In this thesis we presented a novel issue of a possible cell and packet discarding at the central stage buffer. The significance of this issue is explained by the fact that although the packet loss itself can be extremely high, it is also has negative consequences for the system reassembly. In spite of the fact that several algorithms were proposed to cope with such a behavior, in the future, designers might want to reduce the amount of computation and communication overheads concentrated by the controller. As another open issue, this thesis made attempt to evaluate the amount of reassembly delays appearing at re-sequencing and reassembly unit. Although the results showed strong relation between the amount of buffering in the central stage and output, we did not propose any optimal, in terms of scalability and performance, design for re-sequencing and reassembly unit (except of traditional solutions given in [66]). Therefore, as a promising trend, researches might consider the possibility to introduce some novel designing schemes for the output and re-sequencing and reassembly unit and present methodology for assignment of optimal buffering space. Finally, as a stand alone future direction one may consider the support of multicasting and broadcasting by the switch. According to our expectations this issue might require some complex buffers management as well as additional functional block designs.

Bibliography

[1] S. Arekapudi, S.T. Chuang, I. Keslassy, and N. McKeown. Configuring a Load-Balanced Switch in Hardware. pages 48 – 53, Stanford, USA, August 2004. Proc. of the IEEE Hot Interconnects XII.

[2] Y. Audzevich, L. Bodrog, Y. Ofek, and M. Telek. Packet Loss Analysis of Load-Balancing Switch with ON/OFF Input Processes. pages 197 – 211, London, UK, July 2009. EPEW' 09.

[3] Y. Audzevich, L. Bodrog, Y. Ofek, and M. Telek. Scalable Model for Packet Loss Analysis of Load-Balancing Switches with Identical Input Processes. pages 249 – 263, Madrid, Spain, June 2009. IEEE ASMTA' 09.

[4] Y. Audzevich, L. Bodrog, M. Telek, Y. Ofek, and B. Yener. Variable Size Packets Analysis in Load-Balanced Switch with Finite Buffers. Technical report, January 2009.

[5] Y. Audzevich, M. Corra, G. Fontana, Y. Ofek, and D. Severina. Energy Efficient All-Optical SOA Switch for the Green Internet. pages 1 – 4, Pisa, Italy, May 2009. FOTONICA' 09.

[6] Y. Audzevich and Y. Ofek. Assessment and Open-Issues of the Load-Balanced Switching Architecture. pages 54 – 61, Hainan Island, China, December 2008. IEEE FGCN' 08.

[7] M. Baldi and Y. Ofek. Fractional Lambda Switching Principles of Operation and Performance Issues. *Simulation*, 80(10):527–544, 2004.

[8] P. Baran. On Distributed Communications Networks. *IEEE transactions on Communications Systems*, 12(1):1–9, March 1964.

[9] P. Billingsley. *Ergodic Theory and Information*. Wiley, London/New York, 1965.

[10] E. Blanton and M. Allman. On Making TCP More Robust to Packet Reordering. *SIGCOMM Comput. Commun. Rev.*, 32(1):20–30, 2002.

[11] Andrei Z. Broder and Michael Mitzenmacher. Using Multiple Hash Functions to Improve IP Lookups. In *IEEE INFOCOM' 01*, pages 1454–1463, 2001.

[12] T. Chaney, J.A. Fingerhut, M. Flucke, and J.S. Turner. Design of a Gigabit ATM Switch. In *IEEE INFOCOM '97*, volume 1, pages 2–11, Apr 1997.

[13] C. S. Chang, Y. T. Chen, J. C., and D.-S. Lee. Multistage Constructions of Linear Compressors, Non-Overtaking Delay Lines, and Flexible Delay Lines. In *INFOCOM' 06*, 2006.

[14] C.S. Chang, Y.T. Chen, and D.S. Lee. Constructions of Optical FIFO Queues. *IEEE/ACM Transactions on Networking*, 52(6):2838–2843, 2006.

[15] C.S. Chang, D. Lee, and Y.J. Shih. Mailbox Switch: A Scalable Two-Stage Switch Architecture for Conflict Resolution of Ordered Packets. volume 3, pages 1995 – 2006, Hong Kong, March 2004. IEEE INFOCOM' 04.

[16] C.S. Chang, D.S. Lee, and Y.S. Jou. Load-Balanced Birkhoff-von Neumann Switches. pages 276–280, Dallas, May 2001. IEEE HPSR' 01.

[17] C.S. Chang, D.S. Lee, and Y.S. Jou. Load-Balanced Birkhoff-von Neumann switches, Part I: One-Stage Buffering. *Computer Communications*, 25:611–622, 2002.

[18] C.S. Chang, D.S. Lee, and C.M. Lien. Load-Balanced Birkhof-von Neumann switches, Part II: Multi-Stage Buffering. *Computer Communications*, 25:623–634, 2002.

[19] J. Chao, S. Liew, and Z. Jing. A Dual-Level Matching Algorithm for 3-Stage Clos-Network Packet Switches. *High-Performance Interconnects, Symposium on*, 0:38, 2003.

[20] Shang-Tse Chuang, Sundar Iyer, and Nick Mckeown. Practical Algorithms for Performance Guarantees in Buffered Crossbars. In *IEEE INFOCOM' 05*, 2005.

[21] I. Cidon and Y. Ofek. Metaring - A Full-Duplex Ring With Fairness and Spatial Reuse. In *INFOCOM' 90*, pages 969–981, 1990.

[22] R. Cole, K. Ost, and S. Schirra. Edge-Coloring Bipartite Multigraphs in O(ElogD) Time. *Combinatorica*, 21(1):5 – 12, 2001.

[23] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden. Routers with Very Small Buffers. Barcelona, Spain, April 2006. IEEE INFOCOM' 06.

[24] Anja Feldmann and Ward Whitt. Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models. In *INFOCOM '97*, page 1096, Washington, DC, USA, 1997. IEEE Computer Society.

[25] M. Fomenkov, K. Keys, D. Moore, and K Claffy. Longitudinal Study of Internet Traffic in 1998-2003. In *Proceedings of WISICT*, Mexico, 5-8. January 2004.

[26] P. Giaccone, B. Prabhakar, and D. Shah. Randomized Scheduling Algorithms for Input-Queued Switches. *IEEE Journal of Selected Areas in Communication*, 21(4):642 – 655, 2003.

[27] A. Gravey, J. R. Louvion, and P. Boyer. On the Geo/D/1 and Geo/D/1/n queues. volume 11, pages 117 – 125. Performance Evaluation journal, 1990.

[28] M. Gusat, F. Abel, F. Gramsamer, R. Luijten, C. Minkenberg, and M. Verhappen. Stability of CIOQ Switches with Finite Buffers and Non-Negligible Round-Trip Time. pages 444 – 448, Cambridge, UK, October 2002. IEEE Computer Communications and Networks.

[29] J. U. Hui and T. Renner. Queuing Strategies for Multicast Packet Switching. volume 3, pages 1431 – 1437, San Diego, USA, December 1990. IEEE GLOBECOM' 90.

[30] I.Keslassy, S.T. Chuang, and N. McKeown. A Load-Balanced Switch with an Arbitrary Number of Linecards. volume 3, pages 2007 – 2016, Hong Kong, March 2004. IEEE INFOCOM' 04.

[31] I.Keslassy, S.T. Chuang, N. McKeown, and D.S. Lee. Optimal Load-Balancing. volume 3, pages 1712 – 1722, Miami, March 2005. IEEE INFOCOM' 05.

[32] I.Keslassy and N. McKeown. Maintaining Packet Order in Two-Stage Switches. volume 2, pages 1032 –1041, New York, USA, June 2002. IEEE INFOCOM '02.

[33] S. Iyer and N. McKeown. Making Parallel Packet Switches Practical. volume 3, pages 1680 –1685, Anchorage, USA, April 2001. IEEE INFOCOM '01.

[34] J.J. Jaramillo, F. Milan, and R.Skrikant. Padded Frames: A Novel Algorithm for Stable Scheduling in Load-Balanced Switches. pages 1732 –1737. The 40th Annual Conference on Information Sciences and Systems, March 2006.

[35] Michael Jurczyk. Performance and Implementation Aspects of Higher Order Head-of-Line Blocking Switch Boxes. pages 49 – 53, Bloomington, USA, August 1997. IEEE Parallel Processing' 97.

[36] M. Karol, M. Hluchyj, and S. Morgan. Input Versus Output Queueing on a Space-Division Packet Switch. *IEEE Transactions on Communications*, 35(12):1347–1356, Dec 1987.

[37] M. Katevenis, G. Passas, D. Simos, I. Papaefstathiou, and N. Chrysos. Variable Packet Size Buffered Crossbar (CICQ) Switches. volume 2, pages 1090 – 1096, Paris, France, June 2004. IEEE ICC' 04.

[38] F. P. Kelly, M. Taqqu, S. Zachary, W. Willinger, Murad S. Taqqu, and A. Erramilli. A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks, 1996.

[39] I. Keslassy. *The Load-Balanced Router*. PhD thesis, Stanford University, Stanford, 2004.

[40] I. Keslassy, S.T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaad, and N. McKeown. Scaling Internet Routers Using Optics. pages 189 – 200, Karlsruhe, Germany, May 2003. ACM SIGCOMM'03.

[41] Leonard Kleinrock. *Queueing Systems: Volume I Theory*. Wiley, New York, 1975.

[42] H. Kogan and I. Keslassy. Optimal-Complexity Optical Router. pages 706 – 714, Anchorage, Alaska, May 2007. IEEE INFOCOM' 07.

[43] Jonathan G. Koomey. Estimating total power consumption by servers in the u.s. and the world. Technical report, Lawrence Berkeley National Laboratory, Berkeley, USA, February 2007.

[44] W.E. Leland, M. S. Taqqu, W. Willinger, and D.V. Wilson. On the Self-Similar Nature of Ethernet Traffic. *IEEE/ACM Transactions on Networking*, 2(1):1 – 15, 1994.

[45] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan. On the Stability of Input-Queued Switches with Speed-Up. *IEEE/ACM Transactions on Networking*, 9:104–118, 2001.

[46] Bill Lin and Isaac Keslassy. The Concurrent Matching Switch Architecture. pages 1 – 12, Barcelona, Spain, April 2006. IEEE INFOCOM' 06.

[47] Bill Lin and Isaac Keslassy. Frame-Aggregated Concurrent Matching Switch. pages 107 – 116, Orlando, December 2007. ACM/IEEE ANCS '07.

[48] N. Maxemchuk. Routing in the Manhattan Street Network. *IEEE Transactions on Communications*, 35(5):503–512, May 1987.

[49] N. McKeown. The iSLIP Scheduling Algorithm for Input-Queued Switches. *ACM Trans. on Networking*, 7(2):188 – 201, 1999.

[50] N. McKeown, V. Anantharan, and J. Walrand. Achieving 100 Throughput in an Input-Queued Switch. pages 296 – 302, San Francisco, USA, March 1996. IEEE INFOCOM '96.

[51] D. Nassimi and S. Sahni. A Self Routing Benes Network. In *ISCA '80*, pages 190–195, New York, USA, 1980. ACM.

[52] M.J. Nelly, E. Modiano, and Y.S. Cheng. Logarithmic Delay for N x N Packet Switches Under the Crossbar Constraint. *IEEE Transaction on Networking*, 15(3):657–668, 2007.

[53] A.M. Norros. A Storage Model with Self-Similar Input. *Queueing Systems*, 16:387–396, 1994.

[54] Y. Ofek and M. Yung. Routing and Flow Control on the Metanet: an Overview. *ACM Computer Networks and ISDN Systems*, 26(6-8):859–872, 1994.

[55] Y. Ofek and M. Yung. METANET Principles of an Arbitrary Topology LAN. *IEEE/ACM Transactions on Networking*, 3(2):169–180, 1995.

[56] R. Rojas-Cessa, E. Oki, and H. Jonathan Chao. CIXOB-k: Combined Input-Crosspoint-Output Buffered Packet Switch. volume 4, pages 2654 – 2660, San Antonio, USA, November 2001. IEEE GLOBECOM' 01.

[57] Y. Shen, S. Jiang, S.S. Panwar, and H.J. Chao. Byte-Focal: A Practical Load-Balanced Switch. pages 6 – 12, Hong Kong, May 2005. IEEE HPSR' 05.

[58] Rishi Sinha, Christos Papadopoulos, and John Heidemann. Internet packet size distributions: Some observations. Technical Report ISI-TR-2007-643, USC/Information Sciences Institute, May 2007. Orignally released October 2005 as web page `http://netweb.usc.edu/~rsinha/pkt-sizes/`.

[59] W. Stallings. *High-Speed Networks and Internets, Performance and Quality of Service*. Prentice Hall, New Jersey, 2001.

[60] J.K. Sundararajan. Extending the Birkhoff-von Neumann Switching Strategy to Multicast Switching. Master's thesis, Cambridge, USA, February 2005.

[61] Hideaki Takagi. *Queueing Analysis: Volume 3 - Discrete Time Systems*. North Holland, Amsterdam, 1993.

[62] T.Anderson, S. Owicki, J. Saxe, and C. Thacker. High Speed Switch Scheduling for Local Area Networks. *ACM Trans. Comput. Syst.*, 11(4):319 – 352, 1993.

[63] K. Thompson, G.J. Miller, and R. Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*, 11:10 – 23, 1997.

[64] F.A. Tobagi, T. Kwok, and F.M. Chiussi. Architecture, Performance, and Implementation of the Tandem Banyan Fast Packet Switch. *Selected Areas in Communications, IEEE Journal on*, 9(8):1173–1193, Oct 1991.

[65] C.Y. Tu, C.S. Chang, D.S. Lee, and C.T. Chiu. Design a Simple and High Performance Switch Using a Two-Stage Architecture. volume 2, pages 6 – 11, St. Louis, USA, November 2005. IEEE GLOBECOM '05.

[66] Jonathan Turner. Resilient Cell Resequencing in Terabit Routers. WUCS 03-48, Washington University, Department of Computer Science, June 2003.

[67] Jonathan Turner. Strong Performance Guarantees for Asynchronous Crossbar Schedulers. pages 1 – 11, Barcelona, Spain, April 2006. IEEE INFOCOM '06.

[68] L.G. Valiant and G.J. Brebner. Universal Schemes for Parallel Communication. pages 263 – 277. Proc. of the 13th ACM Symposium on Theory of Computation, 1981.

[69] Y.Audzevich, Y. Ofek, M. Telek, and B. Yener. Analysis of Load-Balanced Switch with Finite Buffers. pages 1 – 6, New Orleans, USA, December 2008. IEEE GLOBECOM' 08.

[70] J. Ye and S. Li. Courier dover publication. *Folding Algorithm: A Computational Method for Finite QBD Processes with Level-Dependent Transitions*, 42(2/3/4):652–639, February/March/April 1994.

[71] B. Yener, Y. Ofek, and M. Yung. Combinatorial Design of Congestion-Free Networks. *IEEE/ACM Transactions on Networking*, 5(6):989–1000, 1997.

[72] B. Yener, Y. Ofek, and M. Yung. Convergence Routing on Disjoint Spanning Trees. *Computer Networks*, 31(5):429–443, 1999.

[73] Kwan L. Yeung, Bing Hu, and N.H. Liu. A Novel Feedback Mechanism for Load-Balanced Two-Stage Switches. pages 6193 – 6198, Glasgow, Scotland, June 2007. IEEE ICC '07.

[74] Kwan L. Yeung, Bing Hu, and N.H. Liu. Load-Balanced Three-Stage Switch Architecture. pages 1 – 6, New York, USA, June 2007. IEEE HPSR '07.

[75] C. L. Yu, C.S. Chang, and D.S. Lee. CR Switch: A Load-Balanced Switch with Contention and Reservation. pages 1361 – 1369, Anchorage, USA, May 2007. IEEE INFOCOM' 07.

Appendix A

Acronyms

LAN Local Area Network

WAN Wide Area Network

TCP Transmission Control Protocol

VOQ Virtual Output Queue

VIQ Virtual Input Queue

I-VOQ Virtual Output Queue with Insertion

CS Central Stage (buffers)

CSS Central Stage buffer Set

FIFO First In First Out (policy)

FCFS First Come First Served (policy)

HOL Head-Of-Line (blocking)

TDS Time Driven Switching

PIM Parallel Iterative Matching

RRU Re-sequencing and Reassembly Unit

CR Contention and Reservation (switch)

CMS Concurrent Matching Switch

FA-CMS Frame Aggregated Concurrent Matching Switch

MSN Manhattan Street Network

LB Load-Balancing (switch)

UFS Uniform Frame Spreading

PF Padded Frames

VWT Virtual Waiting Time

FOFF First Ordered Frames First

MWM Maximum Weight Matching

SPFA Success:Persist/Failure:Advance (scheme)

EDF Earliest Deadline First

NL NoLoss (switch)

MOV Maximal Occupancy Vector

CSSV Central Stage Space Vector

PMF Probability Mass Function

QBD Quasi-Birth Deathlike (process)

DTMC Discrete Time Markov Chain

DMAP Discrete Markovian Arrival Process

DPH Discrete Phase Type