General Method for Finding the Most Economical Distributed Router Architecture

Lukas Kencl
IBM Research, Zurich Research Laboratory
Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland
lke@zurich.ibm.com

Božidar Radunović
Department of Communication Systems (DSC)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
bozidar.radunovic@epfl.ch

Keywords: Router architecture, distributed systems, packet processing, queuing, cost optimization.

Abstract

In this work we present a novel method to determine the optimal parameters of a router architecture when certain router performance constraints are given. The total financial expense, or cost, is the optimality criterion. We introduce a general, essentially distributed, router architecture model, consisting of locally or remotely located forwarding engines or processing units gathered around a switch of variable speed. Given the following constraints: number of inputs, maximum line interface bandwidth, and maximum packet delay in a router, the presented method finds the optimal amount and distribution of processing power among the various available processing units and the optimal parameters for the switching element. The optimization employs an estimated market-based cost function per element and finds the most economical system solution.

The results show that the optimal solutions gather around two extreme points of the solution space, distinguishable by the distribution of the processing power mass and corresponding switch speed. We discuss when, depending on the customer input, one or the other solution is appropriate.

1 INTRODUCTION

Latest developments in transmission technologies have led to an enormous increase in the potential amount of data transported over the links of a complex network like the Internet. Such rapid evolution places significant strain on the interconnecting equipment, primarily routers, to scale with the pace of the transmission speed increase. Recent works [1], [2] have provided a basis for the new generation of interconnecting devices by presenting the first gigabit and terabit router architectures. These works have built on new developments in the areas of switch architectures [3], [4] and fast lookup algorithms [5], [6].

In order to eliminate the packet processing bottleneck, *multiple* processing units, known as forwarding engines (FE), or, more sophisticated, network processors (NP), are typically deployed in contemporary routers. A router system thus consists of multiple processing units gathered around a switch element and packet processing within such a system is essentially *distributed*. Various architectures have emerged in the industry with respect to the exact locations and capacities of the individual router elements. With the ever-increasing number of processing units and the total processing power present in the system, more emphasis is being put on the most economical use of these resources.

In principle, packet processing can either be carried out directly at router inputs, using local forwarding engines (LFE), or at remote master forwarding engines (MFE), reachable through the switch element. Both of these paradigms may be combined in one system. Given the performance demands, a single MFE may not be sufficient and thus MFEs are often grouped into pools of parallel MFEs. The critical question is–what are the best capacities, locations and schemes of cooperation of all the elements (switch, LFEs, MFEs) in order to satisfy given system performance demands, and what is the most economical alternative to satisfy those demands?

In this paper, a general model of an essentially distributed router architecture is optimized, spanning a large set of hybrid architectures. Our model (see Fig. 1) contains a fixed number of LFEs (one per router input) of variable processing power (including null processing power, meaning that the LFEs are absent), a variable number of MFEs with variable processing power, and a switch of variable port speed. This work extends the model presented in [7] by a switch element and an LFE queuing and queue overload model. Furthermore, we present a hybrid, more general router architecture model, encompassing a large space of possible, in essence distributed, router architectures, including the centralized, fully distributed and parallel cases presented in [7].

The *objective* of the method is to serve as a general means for optimizing a router architecture with a given set of constraints. The constraints, which can be interpreted as customer input, are maximum line interface bandwidth, number of router LCs, and maximum packet processing delay within the system. To carry out the optimization on a realistic model, the system model contains further constraints, which can be interpreted as technological limits, such as the maximum processing power of the FEs or the maximum switch port speed. The full set of constraints defines a space of feasible solutions over which the optimization is carried out. The optimization cost function is an aggregate of estimated market-based costs of the individual elements. The cost function takes the technological parameters of each element (FE processing power, number of switch ports, and switch port speed) as input. The optimization output consists of variables describing the optimal architecture—the processing power of the LFEs and MFEs, number of MFEs, switch port speed, and the distribution of packet processing among LFEs and MFEs.

A *second objective* of this work is to reach some general conclusions about the most economical router architecture for a given set of constraints.

The paper is organized as follows: Section 1 presents the router architecture model, and, in Section 3, the cost optimization problem is described in detail. In Section 4, results of the most interesting optimizations are discussed. Finally, Section 6 contains some concluding remarks.

2 GENERAL DISTRIBUTED ROUTER ARCHITECTURE

2.1 Router Model

We consider a router having k LCs, with an LFE attached at every LC (see Fig. 1). A switch element interconnects all the LFEs and the pool of m parallel MFEs. All the possible sequences the processing of a packet may take–at an LFE, at an MFE, or at both processing units—are accounted for. A fraction $r \in [0, 1]$ of the incoming traffic is processed locally at the LFEs; the fraction $1 - r$ is diverted directly to the MFEs without being enqueued at the LFEs (see Fig. 2). Thus, if $r = 0$, LFEs are not used at all and the router consists only of a pool of parallel MFEs and a switch. When $r > 0$ the LFE may not be able to handle all the traffic destined for it locally, for various reasons, such as for being overloaded (see Section 2.2). Such traffic is sent to the MFEs as well, but only after passing through the LFE. Thus, if $r = 1$, all the traffic is enqueued at the LFEs, yet a fraction that the LFEs will not be able to process will still subsequently be sent to the MFEs.

Arriving traffic is modeled as a Poisson process, which simplifies the analysis. Some form of bursty traffic is typically observed in networks [8]. Bursty traffic would require a different, worst-case analysis with respect to the system capacity. Recent works [9] have suggested that the aggregate arrival traffic on an uncogested Internet link does tend to Poisson and thus our assumption may not be far from reality. The mean arrival rate at each input LC is λ_i packets per second (pps). The total router load is thus $\lambda = k\,\lambda_i$. In the analysis we consider all the links to be fully loaded. With respect to the optimization, parameters k and λ_i are a part of the optimization input, whereas values of r and m are a part of the output.

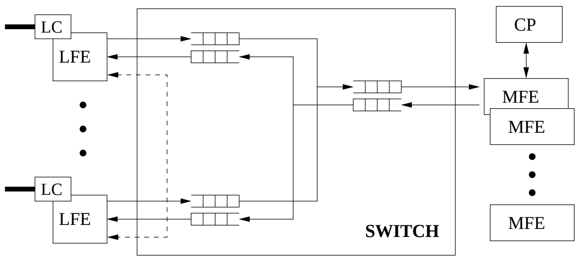

Figure 1: LFEs, switch and MFEs within the general distributed router architecture model.

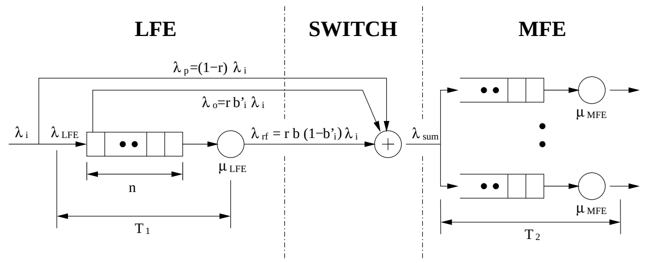

Figure 2: Model of the general distributed router architecture with multiple LFEs and MFEs.

2.2 LFE and MFE Model

The processing power of an MFE is μ_{MFE} pps, and that of an LFE is μ_{LFE} pps. We use $\mu_{\max}$ as a bound on the maximum number of packets an FE can handle per second. The possible scenarios of packet-processing distribution among LFEs and MFEs are depicted in Fig. 2. The fraction of traffic arriving at an LFE is $\lambda_{\text{LFE}} = r\lambda_i$. A fraction $\lambda_p = (1-r)\lambda_i$ is pre-scheduled directly for processing at the MFEs.

Regarding the packets sent for resolution through the switch to the MFEs, we assume that it is only the packet control information, i.e. the packet header, that travels through the switch (as in [10]). The packet payload is assumed to be buffered until a resolution of the packet processing task arrives from the MFE pool, again, traveling through the switch.

Furthermore, we assume that a single LFE workload, λ_{LFE}, can be greater than its processing power μ_{LFE}. The LFE queue size n is introduced as a parameter to model the LFE overload. When the LFE queue is full, a packet *cannot* be processed by the LFE and is forwarded to the MFE pool. Note that the overload traffic does not have a Poisson distribution because the probability that an LFE queue is full depends on the LFE load, but, for the sake of simplicity, we approximate it with a Poisson distribution as follows: the LFE queue is an $M/M/1/n$ queue. Thus, the probability of a packet arriving at a full LFE queue is (see [11]):

$$b'(\lambda_{\text{LFE}}) = P_n = \frac{1-\rho}{1-\rho^{n+1}}\rho^n, \quad \rho = \lambda_{\text{LFE}}/\mu_{\text{LFE}}. \quad (1)$$

Thus, we assume the fraction of traffic $\lambda_o = b'\,\lambda_{\text{LFE}} = r\,b'\,\lambda_i$ to be sent to the MFE pool due to LFE overload.

Furthermore, as in [7], even if a packet *is* being processed by an LFE, with a fixed probability b the LFE will not be able to find the packet next hop, and the packet is likewise forwarded to the MFE pool. Such packets account for route table misses, for example when the LFE acts only as a cache, storing a fraction of the routing table. We denote such a fraction of traffic as $\lambda_{rf} = r\,b\,(1-b')\lambda_i$.

Finally, the fraction of traffic that actually does get resolved at the LFE and is forwarded directly to the outgoing switch port is $\lambda_q = \lambda_i - (\lambda_p + \lambda_o + \lambda_{rf}) = r(1-b)(1-b')\lambda_i$.

In Fig. 2 we observe that there are three possibilities for a packet to be queued. Either a packet is queued at an LFE and waits for time T_1, it is forwarded to the MFE and waits for T_2, or it is queued at both the LFE and the MFE owing to the LFE resolution failure.

LFE processing time. The average number of packets in a processor, the average workload arriving at the LFE queue, and the average LFE response time are

$$\overline{N}(\lambda_{\text{LFE}}) = \rho\,\frac{n\rho^{n+1} - (n+1)\rho^n + 1}{\rho^{n+2} - \rho^{n+1} - \rho + 1} \quad (2)$$

$$\lambda_a = \lambda_{\text{LFE}}\,(1-P_n) = \lambda_{\text{LFE}}\,\frac{1-\rho^n}{1-\rho^{n+1}} \quad (3)$$

$$\overline{W}(\lambda_{\text{LFE}}) = \frac{(1-\rho^{n+1})\,(n\rho^{n+1} - (n+1)\rho^n + 1)}{\mu_{\text{LFE}}\,(1-\rho^n)\,(\rho^{n+2} - \rho^{n+1} - \rho + 1)} \quad (4)$$

In reality, workloads on different LCs are generally not uniform and may vary significantly over time. This implies that LFEs with a higher workload would forward more packets to the MFEs for processing than LFEs with a smaller workload, and one can imagine a feasible problem solution where for some periods of time, individual LFEs would be overloaded. We have experimented with nonuniform workload distributions on different LCs and the LFE overload, but the optimization results did not differ significantly from the uniform model. Nonuniform LC workloads are therefore not included in the model. The LFE model, as presented here, is applicable for the overload modeling, however the optimization never finds an overloaded LFE solution to be the optimal one. Therefore, we simplify the model assuming an infinite LFE buffer and no queue overflows. Time T_1 is simply the average response time $\overline{W}(\lambda_{\text{LFE}})$ for very large n, and it corresponds to average $M/M/1/\infty$ waiting time.

MFE pool processing time. The MFE pool queue is, as in [7], a simple infinite $M/M/m$ queue, with the input workload representing the sum of nonprocessed packets from the LFEs, together with the pre-scheduled packets. As the part of workload sent for resolution to the MFE represents a sum of Poisson processes, the sum is a Poisson process as well. This workload and the corresponding $M/M/m$ queue waiting time are on average [11]:

$$\lambda_{\text{sum}} = k\,(\lambda_p + \lambda_o + \lambda_{rf}) \quad (5)$$

$$T_2 = \frac{1}{\mu_{\mathrm{MFE}}} + \frac{\rho P_Q}{\lambda_{\mathrm{sum}}(1-\rho)}, \quad (6)$$

where

$$\rho = \frac{\lambda_{\mathrm{sum}}}{m\mu_{\mathrm{MFE}}}, \quad (7)$$

$$P_0 = \frac{1}{\sum_{j=0}^{m-1} \frac{(m\rho)^j}{j!} + \frac{(m\rho)^m}{m!\,(1-\rho)}}, \qquad P_Q = \frac{P_0(m\rho)^m}{m!\,(1-\rho)}. \quad (8)$$

2.3 Switch Model

General input/output switch. The switch is characterized by two parameters—the switch port speed s and the number of input ports k. As k is an input to the optimization, s is the only parameter optimized by the method. The switch port speed is expressed in terms of transmission time per the fixed size switch cell, that is, in seconds per cell. The parameter s is constrained from below by the technological limit of $s_{\min}$, $s > s_{\min}$, which means that a fixed-size switch cell cannot be transmitted at a switch port in less than $s_{\min}$ seconds. To avoid confusion with the intensity indicators in packets per second, the intensity indicators in switch cells per second are denoted with *, e.g. λ^* instead of λ.

We include the switch in our model by introducing the switch delay. In order to model the switch delay time, we consider a formula for an input/output-queued switch derived in [12] and [13]. In the interest of simplicity, we assume that the switch has infinitely large output queues, and that the number of inputs is large (i.e. greater than 16, [12]). Note that the head-of-line congestion at the input port considered in [12] and [13] has been eliminated in the latest switch architectures; however, in this work, we conform to this model in order to obtain a simple analytical formula for computing the switch delay.

A cell arriving at an input port first waits at the input queue, then at the head of the input queue because of head-of-line congestion, and, finally, at the switch output port (see Fig. 1). We denote $W_i(\lambda_x^*)$ as the average waiting time until the head of the input line is reached. The term λ_x^* denotes the intensity of the input traffic (in switch cells per second). This is an $M/G/1$ queue with service time $W_b(\lambda_x^*)$ equal to the time a cell spends waiting at the head of the input queue owing to head-of-line congestion. We denote $D_{\mathrm{out}}(\lambda_y^*)$ as the average delay from the instant a cell appears at the head of its input queue until the instant it begins transmission at the output port. As shown in [12], this is an $M/D/1$ queuing system, as we assume the switching matrix speed to be constant and the input traffic to be Poisson. The term λ_y^* is the intensity of the aggregated Poisson flow arriving at the output port (in switch cells per second).

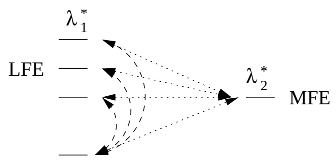

Figure 3: Traffic flows within the switch; λ_1^* represents the amount of traffic leaving the local port (equal to the amount arriving at the local port), and λ_2^* represents the amount of traffic arriving at the master port (equal to the amount that leaves the master port).

The switch traffic consists of k equally loaded *local ports* and one additional port, the *master port*, used for transferring the packet headers to and from the MFE pool (see Fig. 3). As we consider infinite output queue sizes, waiting time due to head-of-line congestion is $W_b(\lambda_x^*) = 0$ because a cell can be queued at the output the moment it arrives at the head of an input queue. It follows from [12, Eq. (2.9)] that

$$W_i(\lambda_x^*) = \frac{\lambda_x^* s^2}{2(1-\lambda_x^* s)},$$

where λ_x^* denotes the intensity of the input traffic (in switch cells per second) and s denotes the switching matrix processing time. From the analysis of an $M/D/1$ queue we have

$$D_{\mathrm{out}}(\lambda_y^*) = \frac{\lambda_y^* s^2}{2(1-\lambda_y^* s)},$$

where λ_y^* is the intensity of the aggregated Poisson flow arriving at the output port (in switch cells per second).

In order to exploit the above switch model in conjunction with the FE models, we need to transform the load variables to a common unit: packets per second. As Internet packets are *variably sized*, we need to use some approximations to establish a relationship to the *fixed-size* switch cells. Two kinds of packets travel through the switch—entire packets, and the packet headers traveling between the LFEs and the MFE pool (see Fig. 3). Let $\bar{S}_P$ denote the average size of a packet in the router incoming traffic, and $\bar{S}_H$ the average size of a header message traveling in the switch. We assume that a packet header corresponds in size to a single switch cell. We denote c as the ratio between the packet header cell size and the average packet size, $c = \bar{S}_H/\bar{S}_P$. Thus an average packet accounts for $1/c$ switch cells. Measurements and analysis have shown that the Internet traffic distribution is highly nonuniform. Empirical values in recent studies show $\bar{S}_P \doteq 300$ B [14]. A typical switch cell size is 64 B, with the payload part being equal to 60 B. Thus, a 40–44 B packet header usually fits into a single such cell, $\bar{S}_H = 64$ B and, consequently, $c \doteq 0.2$ is a good typical value.

A fraction of traffic is sent to the MFE pool for processing. With reference to Fig. 3, we have $\lambda_1^* = \lambda_i/c + (\lambda_p+\lambda_o+\lambda_{rf})$ and $\lambda_2^* = k\,(\lambda_p+\lambda_o+\lambda_{rf})$, where λ_1^* is the total switch outgoing traffic at an LFE port (comprising both packet headers and entire packets) and λ_2^* is the switch outgoing traffic at the master port (comprising packet headers only). The values of λ_1^* and λ_2^* are expressed in switch cells per second, whereas $\lambda_i, \lambda_p, \lambda_o$, and λ_{rf} are expressed in packets per second.

From the above discussion, we see that the waiting time for a packet header traveling to the MFEs is $W_i(\lambda_1^*) + D_{\text{out}}(\lambda_2^*)$ and that the waiting time for the way back to the output port is $W_i(\lambda_2^*) + D_{\text{out}}(\lambda_1^*)$, hence the total switch delay for a packet header trip is

$$D_H(\lambda_i) = \frac{\lambda_1^* s^2}{1-\lambda_1^* s} + \frac{\lambda_2^* s^2}{1-\lambda_2^* s}. \quad (9)$$

An entire packet traveling through the switch after its destination output port has been found by the next-hop resolution process experiences a per-cell delay of $W_i(\lambda_1^*)$ at the switch input and a per-cell delay of $D_{\text{out}}(\lambda_1^*)$ at the output port, and occupies on average $1/c$ switch cells. Thus the mean switch delay for the entire packet is

$$D_P(\lambda_i) = \frac{1}{c}\left(W_i(\lambda_1^*) + D_{\text{out}}(\lambda_1^*)\right) = \frac{\lambda_1^* s^2}{c\,(1-\lambda_1^* s)}. \quad (10)$$

2.4 Time a Packet Spends within the System

The mean time T a packet spends in the system is

$$T = \frac{\lambda_i - \lambda_p - \lambda_o}{\lambda_i}\, T_1 + \frac{\lambda_p + \lambda_o + \lambda_{rf}}{\lambda_i}\, T_2 + \frac{\lambda_p + \lambda_o + \lambda_{rf}}{\lambda_i}\, D_H(\lambda_i) + D_P(\lambda_i), \quad (11)$$

where the first element represents the fraction of packets processed by the LFEs, the second the fraction of packets processed by the MFEs, the third the switch delay for a remotely processed packet header, and the fourth the switch delay for the entire packet traveling through the switch.

3 COST OPTIMIZATION

Forwarding Engine Cost. To simplify the problem, the cost associated with an FE is assumed to be a linear function of processing power of the form cost_{MFE} (US \$) $\doteq c_1\ \mu_{\text{MFE}}$, cost_{LFE} (US \$) $\doteq c_2\ \mu_{\text{LFE}}$, where μ_{LFE} and μ_{MFE} are expressed in packets per second. We denote a as the ratio of the two coefficients, $a = c_1/c_2$.

Effective switch throughput. Recall that s is the switch speed, and k the number of input ports. In order to establish a general measure of the switch performance, we define the *saturation throughput* of the switch to be $\lambda_s^* = k/s$ (in switch cells per second). The notion comes from the fact that the switch delay,

$$D(\lambda_i^*) = \frac{\lambda_i^*\, s^2}{(1-\lambda_i^*\, s)}\ , \quad (12)$$

tends to infinity as the per-input switch load λ_i^* (in switch cells per second) approaches $1/s$ (see [12]).

We denote as the *effective switch throughput* $\lambda_e^* = \alpha\lambda_s^*$ a fraction α of the saturation throughput for which the delay remains within reasonable bounds. For Internet applications, we assume $\alpha = 0.9$. A common switch performance metric used by the industry is the switch effective throughput λ_e' measured in bps. The relationship between λ_e^* and λ_e' holds as $\lambda_e' = 8\,\bar{S}_H\,\lambda_e^*$.

Switch cost function. To establish a switch cost function $\text{cost}_S(s,k)$ dependent on the switch performance, we establish a relationship between λ_e' in bps and the switch cost in US \$. We assume that within certain limits of $k \le k_0$, there is a linear dependency of the switch cost on the switch effective throughput λ_e', which can be characterized as $\text{cost}_S(s,k)$ (US \$) $\doteq c_3\,\lambda_e'(s,k)$, where $k \le k_0$. The limit of k_0 denotes the limit for single-stage switches. For $k > k_0$, we assume more aggressive growth of the switch cost with λ_e', because we assume that such a switch can only be built using a multistage architecture. A typical empirical value of k_0, which is also used in this work, is $k_0 = 64$. From the fixed point of k_0, we assume a cost function dependency of linear-logarithmic form, approximately $\text{cost}_S(s,k)$ (US \$) $\doteq c_3\,\lambda_e'(s,k)\,\log(\lambda_e'(s,k))$, where $k > k_0$. In the interest of a smooth transition and reasonable values, we normalize the function as follows:

$$\text{cost}_S(s,k)\ (\text{US }\$) \doteq c_3\,\lambda_e'(s,k)\,\log\left(e + \frac{\lambda_e'(s,k) - \lambda_e'(s,k_0)}{\lambda_e'(s_{\min},k_0)}\right), \quad (13)$$

where $k > k_0$.

Total System Cost. The total system cost formula holds as

$$\text{cost} = m\,c_1\,\mu_{\text{MFE}} + k\,c_2\,\mu_{\text{LFE}} + \text{cost}_S(s,k). \quad (14)$$

Figure 4: Total system cost.

The linear cost functions for the FEs and the switch are of course simplified from reality. Some form of exponential growth of cost with capacity should rather be expected.

Optimization Problem. The cost is optimized over the tunable system parameters: $(r, \mu_{\mathrm{MFE}}, \mu_{\mathrm{LFE}}, s, m)$. Given the maximum allowed mean packet processing time $T_{\max}$, we derive the following *optimization problem*:

Optimization function	Constraints
$m\, c_1\, \mu_{\mathrm{MFE}} + k\, c_2\, \mu_{\mathrm{LFE}} + cost_S(s,k)$	$0 \le r \le 1$
	$0 \le \mu_{\mathrm{MFE}} < \mu_{\max}$
	$0 \le \mu_{\mathrm{LFE}} < \mu_{\max}$
Parameters	$s_{\min} < s$
$\{r, \mu_{\mathrm{MFE}}, \mu_{\mathrm{LFE}}, s, m\}$	$T < T_{\max}$
	$1 \le m.$

Note that m is an integer, whereas all the other optimized parameters are rational numbers.

4 NUMERICAL RESULTS

4.1 Optimization

Numerical results have been obtained using the Matlab Optimization Toolbox environment. The system variables in our optimizations have been set to the following values: $T_{\max} = 10^{-6}$ s, $c_2 = 6 \times 10^{-5}$, $c_3 = 10^{-8}$, $\mu_{\max} = 6 \times 10^6$ pps, $s_{\min} = 10^{-9}$ s (meaning a fixed-size switch cell would have to be transmitted at a switch port within 1 ns), $n = 100$. The values have been selected to approximate the limited market information available.

The ratio of costs of equally powerful LFEs and MFEs, a, is alternated in our simulations over values $a \in \{2, 10, 100\}$ and influences several variables. Variable c_1 is dependent on c_2 and a, $c_1 = ac_2$. The value of b reflects a in the following manner: we assume that a can be interpreted as a difference in memory size available to the processing engines. Thus a determines a fraction of the memory available at an LFE, and b, the fraction of nonresolvable packets at an LFE, reflects the cache miss rate at an LFE. A sample dependency between a cache hit-rate and cache size can be found for example in [15]. Based on that, we use the following pairs of (a, b): (2, 0.1), (10, 0.2), (100, 0.9).

Figures 4 - 8 show plots of the output variables over a spectrum of the number of inputs k and link speeds λ_i. The number of inputs k grows geometrically with a coefficient of 2, $k \in \{8, 16, 32, ..., 1024\}$. The values on the maximum interface bandwidth λ_i axis grow geometrically with a coefficient of $\sqrt{2}$, thus including link speeds approximately corresponding to the capacities of 10 Mb – 1 Gb Ethernet and OC-48 – OC-768 links, in Mpps, that is, $\lambda_i \in \{0.025, 0.035, 0.050, ..., 6.40, 9.05\}$ Mpps.

4.2 Distribution of Resources—LFEs, MFEs and Switch Capacity

Figure 4 depicts the system cost on a linear and a logarithmic plot for $a = 10$. The cost grows exponentially along both the k and λ_i dimensions, with a steeper increase in the high λ_i segment, which is due to the increasing influence of the switch cost on the total cost.

Figure 5 shows how the optimization selects between the two extreme points of the solution space. For a smaller part of the problem space, r attains the value of 0, meaning no LFEs are needed. When r is equal to 1, the results indicate that it is cheaper to add and fully stress the LFEs to be able to handle the load. It is clearly visible that a certain boundary divides the problem space into two regions. Typically, for systems with a small number of inputs and small link loads, deploying LFEs is too expensive and $r = 0$. This somewhat counter-intuitive result comes from the fact that the processing capacity needed for one $M/M/1$ queue serving the aggregate load is smaller than the processing capacity needed for k $M/M/1$ queues, each serving $1/k$ of the

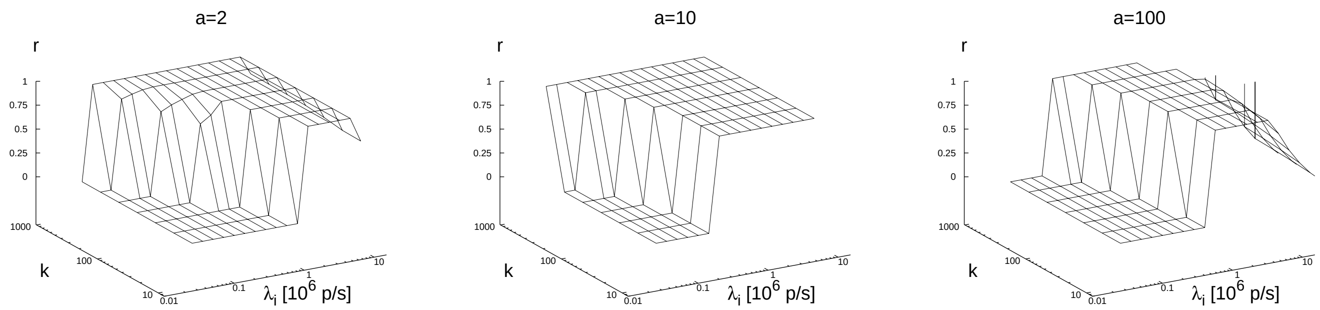

Figure 5: Fraction of traffic enqueued at LFEs, r.

aggregate load, if both systems need to conform to the same time constraint. As the relationship is nonlinear, with the increase in the total amount of processing capacity required, at a certain boundary it becomes more economical to use the multiple, less powerful processors, and thus, for systems with a higher number of ports or higher link loads, LFEs are more economical to fulfill the performance requirements and $r = 1$. The exact curve of the shift differs for various a. The lower a is, the higher the influence of the switch cost on the boundary shape, because when a and b are low, the total system cost remains lower and the switch cost influences the trade-off between $r = 0$ and $r = 1$, described above. However, when a is large, b becomes large as well, and a large fraction of the packets processed at the LFEs have to travel to the MFEs anyway, thus the introduction of LFEs is beneficial only for very high link speeds. The MFEs and the system in total are then very expensive, and the switch cost is no longer a factor in the trade-off.

Two further observations can be made. For $a = 2$ and $a = 100$, r starts to decrease from 1 towards 0 for very high λ_i. Furthermore, for $a = 100$ and very high k and λ_i, r is undefined, meaning that it is not feasible to build a router with the constraints given.

4.3 Distribution of Processing Capacity—LFEs and MFEs

Figure 6 shows the optimal amount of processing power at each individual LFE. For $a = 2$ and $a = 100$, the LFE processing power μ_{LFE} reaches the upper limit $\mu_{\max}$ in the very high λ_i region, which is the reason for decrease in r mentioned at the end of Section 4.1.

The boundary where the LFE processing power limit is reached differs for various a. This phenomenon is again dependent on the particular pairing of (a, b), and, in particular, on the fraction of the LFE resolution failures b, which increases the required LFE processing power. When LFE reaches power limit, packets in excess are transfered to MFEs, if the additional traffic can be handled by the switch (Fig. 7).

4.4 Switch Speed

Figures 8 and 9 depict the optimal switch port transmission speed and the header traffic at the master port. We see why there is no feasible solution for $a = 100$. Given the dependence of b on a, a large fraction of traffic ($b = 0.9$) fails to be resolved at the LFEs and still travels through the switch to the MFE pool. Thus, LFEs cannot be used to a great extent to decrease the packet delay in a router, and the switch, the master port in particular, is placed under increased demand to compensate the delay. For the very high-speed, many-input case, the switch is not able to cope with the demand and reaches the limit of its port transmission speed. Therefore a feasible solution for this region does not exist. Note that a similar phenomenon is only narrowly avoided in the case of $a = 2$, when the LFEs are already saturated as well.

To summarize the optimization results, we observe that for each a and b, a similar trend of dividing the problem space according to the optimum solution exists. However, the boundary line and the appropriate switch element parameters vary significantly for different a and b, which suggests that they are important factors in any router design. Note that the results are, to a large extent, also dependent on c, the ratio of the average packet size and the packet header size. If, for example, the average packet size were to decrease, the influence of the packet-header-passing overhead traffic on the system would gain a much higher significance, and vice versa. Furthermore, with a more realistic cost function, less capable elements would be favored and thus a shift in the division boundaries could be expected.

5 FUTURE WORK

There are many possible improvements to the model and to the way the system behavior is studied. Such improvements comprise:

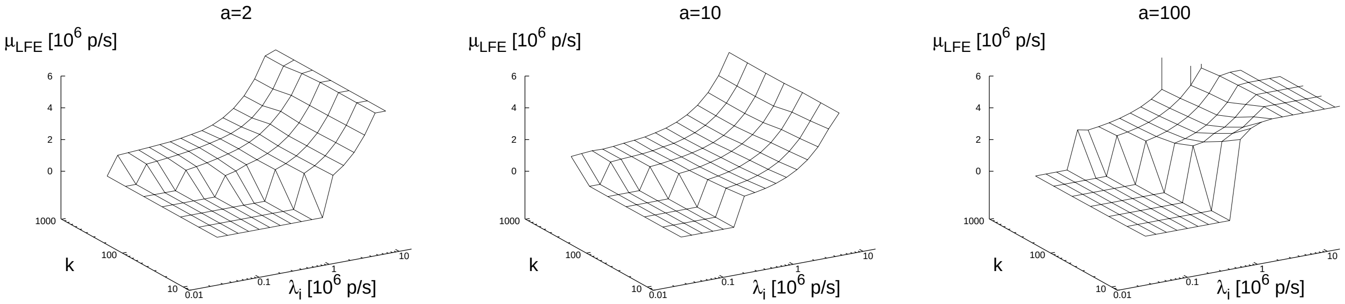

Figure 6: Optimal processing power per individual LFE.

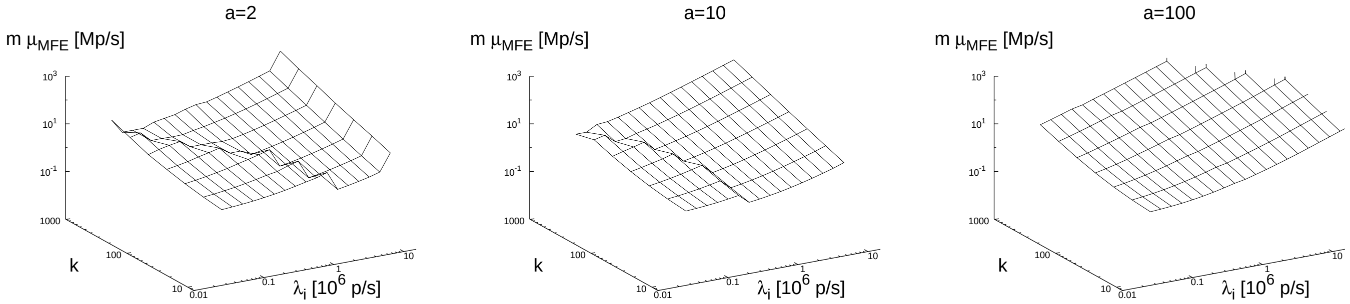

Figure 7: Total processing power of the MFEs.

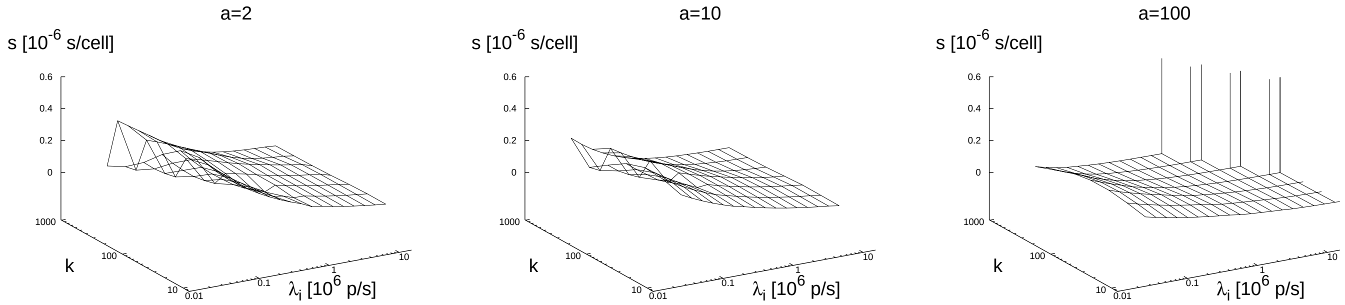

Figure 8: Switch port transmission speed.

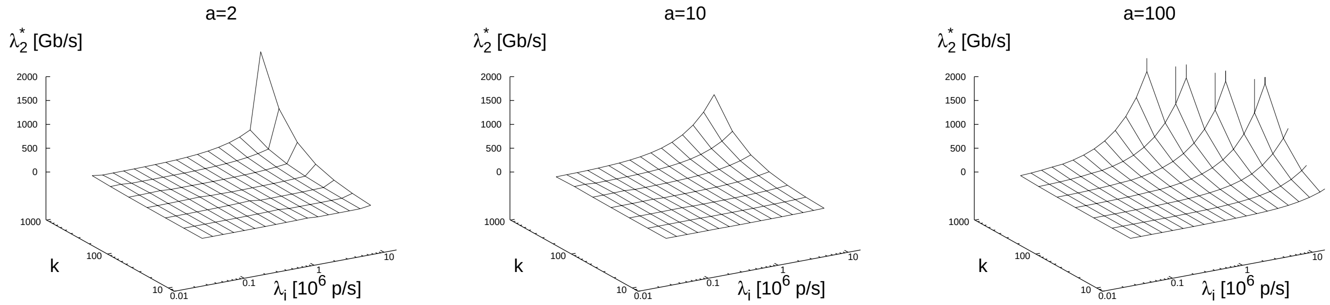

Figure 9: Traffic at the switch master port—the header passing overhead within the switch.

– decoupling the packet-header passing from the switching of entire packets within the switch element;
– more realistic traffic modeling, such as self-similar traffic [8];
– including payload buffering and packet reassembly in the processing model, and
– evaluating load balancing on a per-flow basis and addressing packet reordering prevention.
We have focused here on the worst-case performance: all router links are fully loaded. However, a useful analysis could concentrate on the system behavior within a certain load range.

6 CONCLUSION

This work presents a model for analyzing a general distributed router architecture and determining its most economical variant, given a set of performance requirements. We bring new insight as to how the individual elements influence the most economical architecture for a given set of constraints. The introduction of a nonlinear LFE overload model allows us to easily combine both LFEs and MFEs within one scalable hybrid system. Using a simple switch model, we have demonstrated that the introduction of a switching delay can significantly influence the optimization results. The introduction of the two elements enables us to enhance the scalability limits of the fully distributed router architecture, as they are reached only for the most extreme points of the spectrum studied. For low-end systems, we have demonstrated that the LFE alternative is too costly. However, for most systems, deploying LFEs is advantageous because the switch bottleneck and the high MFE cost are avoided. We show that the cost ratio of the equally powerful LFEs and MFEs and the corresponding fraction of load the LFE is able to handle are a decisive factor in determining the location of the boundary between the two optimal solutions as well as in selecting the appropriate switch element.

References

[1] C. Partridge, et al. 1998. “A 50-Gb/s IP Router”. *IEEE/ACM Transaction on Networking*, 6(3), June: 237–248.

[2] V. P. Kumar, T. V. Lakshman, D. Stiliadis. 1998. “Beyond Best Effort: Router Architecture for the Differentiated Services of Tomorrow’s Internet”. *IEEE Communication Magazine*, May: 152–164.

[3] A. Engbersen, C. Minkenberg. 2000. “A Combined Input and Output Queued Packet-Switched System based on a Prizma Switch-on-a-Chip Technology”. *IEEE Communications Magazine*, Vol. 38, No. 12, December: 70-77.

[4] N. McKeown, et al. 1996. “Achieving 100% throughput in an input-queued switches”. In *Proceedings of the 1996 IEEE INFOCOM* (San Francisco, CA, March). 296–302.

[5] A. Brodnik, et al. 1997. “Small forwarding tables for fast route lookups”. In *Proceedings of the 1997 ACM SIGCOMM* (Cannes, France, September). 3–14.

[6] M. Waldvogel, et al. 1997. “Scalable high speed IP route lookups”. In *Proceedings of the 1997 ACM SIGCOMM* (Cannes, France, September). 25–37.

[7] H. Chan, H. Alnuweiri, V. Leung. 1998. “A Framework for Optimizing the Cost and Performance of Next-Generation IP Routers”. *IEEE Journal on Selected Areas in Communications*, 17(6), June: 1013–1029.

[8] W. Willinger, M. Taqqu, R. Sherman, D. Wilson. 1997. “Self Similarity through High Variablity: Statistical Analysis of Ethernet LAN Traffic at the Source Level”. *IEEE/ACM Transactions on Networking*, (5): 71–96.

[9] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun. 2001. “On the Nonstationarity of Internet Traffic”. In *Proceedings of the 2001 ACM SIGMETRICS*. 29:102–112.

[10] Juniper Networks, Inc.. 2001. “M160 Internet Backbone Router Datasheet”. `http://www.juniper.net`, August.

[11] L. Kleinrock. 1975. *Queuing Systems*. John Whiley & Sons.

[12] I. Iliadis, W. Denzel. 1993. “Analysis of Packet Switches with Input and Output Queueing”. *IEEE Trans. on Communications*, 41(5), May: 731–740.

[13] I. Iliadis, W. Denzel. 1992. “Performance of a Packet Switch with Input and Output Queueing under Unbalanced Traffic”. In *Proceedings of the 1992 INFOCOM* (May). 743–752.

[14] K. Thompson, G. Miller, R. Wilder. 1997. “Wide-Area Internet Traffic Patterns and Characteristics”. *IEEE Network*, Nov./Dec.: 10–23.

[15] N. McKeown, B. Prabhakar. 1999. “High Performance Switches and Routers: Theory and Practice” *Tutorial M2 of the 1999 ACM SIGCOMM* (Cambridge, MA, August).