Journal of Computer Science 2 (3): 218-223, 2006
ISSN 1549-3636

Design of a High-Performance IP Switching Architecture

Hattab Guesmi, Belgacem Bouallegue, Ridha Djemal and Rached Tourki
Laboratoire d'électronique et de micro-électronique, Faculté des sciences de Monastir, Tunisie

Abstract: In this study we present the architecture for use in high-performance switching networks with support quality of service (QoS) guarantees. Quality of services guarantees in terms of delay, through-put and loss rate can be provided by using mechanism's support like scheduling and buffer management at switching architecture in packet switching networks. Our architecture is based on a new data structure for the scheduling and memories management which is the circular linked list and the pipeline for the active queues elements. In addition to being very fast, the architecture also scales very well to a large number of priority levels and to large queue size. We give a detailed description of the block that support QoS guarantees. However our proposed architecture is composed of three parts: input controller, backplane and output controller. And we give the corresponding algorithms and the corresponding implementation of this architecture.

Key words: IP switching architecture, high-performance, switching networks

INTRODUCTION

The growth of the Internet requires design and development of high-speed IP routers that forward exponentially increasing volume of traffic and provide QoS guarantees at the same time. The traffic in the internet is exploding as it is doubling every few months and the speed of technology is doubled every two years. Emerging multimedia applications will make the explosion even faster. Moreover, multimedia and real-time applications require timing and other quality of service (QoS) guarantees, besides bandwidth, which puts even more burden on the routers. In order to bridge the gap between the increase of computing power and the explosion of bandwidth demand, parallelism has been introduced into the routers design. From the viewpoint of the degree of the parallelism, the routers have evolved into the fourth generation. In order to meet the future requirements, the fourth generation routers are expected to be the next generation of high-performance QoS-scalable routers. By using new architectures, based on parallel and scalable switch fabric, higher degree of parallelism and scalability will be brought into the new system.

Extensive research has been done on the next generation of high-speed routers. Nick Mckeown's group at Stanford University did intensive research on high speed switching[1-3], R. Bhagwan proposed a Design of a high-speed packet switch for fine-grained quality of service guarantees[4]. Many other papers and proposals are dealing with some key issues in high-speed routing, such as high-speed routing table lookup[5], real-time packet scheduling[6], fine grain QoS control[7], high-speed switches for the data path[8] and so on.

Our work has been influenced by these existing systems. However, comparing to these architecture, our prototype focuses on the QoS issues. This study presents a novel, highly-scalable architecture for an IP switching architecture.

QUALITY OF SERVICE MECHANISM'S SUPPORT

QoS (Quality of Service) is a hot topic in both academic and industrial fields for many years. QoS means a series of service requirements that the network should satisfy while delivering data and can be represented by the parameters of delay, delay jitter, loss rate, bandwidth, etc[9]. QoS control is to provide consistent, predictable and controllable data delivery service and to satisfy different application requirements, in another word, to guarantee different class of packets receive different level of services. There're many mechanisms to support QoS, such as the resource reservation (RSVP), admission control in *Integrated Services* (IntServ) and traffic shaping/marking in *Differentiated Services* (DiffServ). The major difference between Int-Serv and Diffserv architecture is the granularity of service differentiation. The IntServ concept lies in resource reservation. Each application requests levels of service in terms of service rate or end-to-end delay. The network accepts or rejects requests according to its resources availability. However, the Int-Serv approach faces potential problems concerning scalability and manageability, since all routers must maintain per-flow state. The main strength of DiffServ, as proposed by the IETF Differentiated Services Working Group, is that it allows IP traffic to be classified into a finite number of service

Corresponding Author: Hattab Guesmi, Laboratoire d'électronique et de micro-électronique, faculté des sciences de Monastir, Tunisie

classes that receive different routing treatment. Routers at the network edges classify packets into predefined service classes based on the demand requirements and characteristics of the associated application. Core routers forward each packet according to its class. By this way, the model provides service differentiation on each node (Per-Hop behaviors) for large aggregates of network traffic. DiffServ achieves scalability and manageability by providing quality per traffic aggregate and not per application flow. While the common and key ones are buffer management and packet scheduling that are called interestedly queue management. Buffer management determines how to allocate buffers and whether to drop an arriving packet according to a certain policy, which mainly influences the packet loss rate and fairness. While packet scheduling is responsible for the management of link capacity, namely, it determines from which queue to select a packet to transmit according certain rules. Packet scheduling mainly influences the bandwidth, delay/jitter and fairness, etc. There has been a great amount of research work on packet scheduling in the past years and many algorithms appeared. The key ideas of most packet scheduling algorithms are to compute an index for each queue and sort them. The scheduling decision is made by selecting the queue with the minimum or maximum value of these indexes. WRR (Weighted Round Robin) and DRR (Deficit Round Robin) are kinds of round robin and easy to implement, but they have weakness in providing delay guarantees. EDF (Earliest Deadline First) and its variants are based on time (queuing delay). Their key ideas are to allocate a delay parameter D_i to each queue as the delay upbound and each arrived packet is tagged with the time stamp $T_i = A_i + D_i$ where A_i is the arrival time. Every time, the packet with the minimum T_i is scheduled. A category of algorithms called PFQ（Packet Fair Queuing) are based on service rate. Their key ideas are to maintain a virtual system time Vi(t), a virtual start time Si(t) and virtual finish time Fi(t) for each queue. Si(t) or Fi(t) is sorted and the queue with its maximum or minimum value is scheduled. By providing the guarantee of service rate to each flow, their delay bounds are controlled. One weakness of PFQ algorithms is the coupling of service rate (bandwidth) and delay that results in the inflexible resource allocation. Floyd and Jacobson have proposed a link-sharing and resource allocation scheme called *class-based queuing* (CBQ) which employs DRR queuing algorithm and differentiate flows into different queue classes. Each queue is serviced in round-robin fashion and receives bandwidth equal to its allocated share. However, the research work on buffer management and packet scheduling are mostly separated, which consider only one or some performance metrics that is insufficient[6,10,11]. Since buffer management is the manipulation of enqueuing and packet scheduling is the manipulation of dequeuing, they have tight relationship. As a matter of fact, both buffer management and packet scheduling mechanisms have effects on almost all the performance metrics (Fig. 1). Thus, to satisfy these performance targets simultaneously, both of them should be taken into consideration, which means the integrated schemes are expected to be more reasonable.

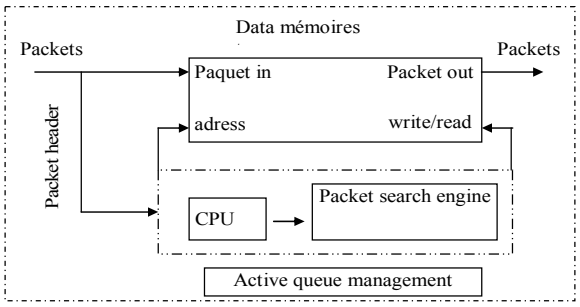

Fig. 1: Active queue management

HIGH-PERFORMANCE FOR IP SWITCHING ARCHITECTURE

Router performance: Parameters that can be used to grade the performance of router architectures witch reflect the exponential traffic growth and convergence of voice, video and data[5,7,12,13].

* High packet transfer rate: increasing Internet traffic has made the packets per second capacity of a router as the single most important parameter for grading its performance. The capacity of the router must be scalable.
* Guaranteed short deterministic delay: real-time voice and video traffic requires short and predictable delay through the system. Unpredictable delay results in a discontinuity, which is not acceptable for these applications
* Quality of service: routers must be able to support service level agreement, guaranteed line-rate and differential quality of service to different applications or flows. This quality of service support must be configurable.
* Multicast traffic: Internet traffic is changing from predominantly point-to-point to multicast and, therefore routers must support large number of multicast transmission simultaneously.
* High availability: high-speed routers located in the backbones handle huge of data and cannot be turned down for upgrades etc. Therefore, features such as hot swappable software tasks allowing in-service software upgrades are required.

Router architecture for the differentiated services: Providing any form of differentiated services requires the network to keep some state information. The majority of the installed routers use architectures that will experience a degraded performance if they are

configured to provide complicated QoS mechanisms. Therefore, the traditional approach was that all the sophisticated techniques should be in the end systems and network should be kept as simple as possible. But recent research and advances in hardware capabilities have made it possible to make network more intelligent[4,14-16].

Component of differentiated services: Following operations need to be performed at a high speed in the router to provide differentiated services:

* Packet classification, which can distinguish packets and group them according to different requirements.
* Buffer management, which determines how much buffer space should be allocated for different classes of network traffics and in case of congestion, which packets should be dropped.
* Packet scheduling, which decides the order in which the packets are serviced to meet different delay and throughput guarantees.

IMPLEMENTATION OF THE QoS SUPPORT BLOCK OF THE IP ROUTER

This block implements the mechanisms that support the QoS such as buffer management and scheduling witch are called active queue management. It consists of five blocs (Fig. 2):

* Data memories
* Dynamic Memory management
* Selector and queue management
* Active queue management
* Scheduler

Data memories: The proposed architecture of an IP switcher is output queueing in ordre to optimize mechnisms of QoS management[8,17]. This memories are intended for the storage of incoming packets of every input ports. It's necessary that the memory is N time faster than input port (LC) which is limited by the material capacity. Our adobted solution for resolving this problem is to use N parallel memories instead of only one memory. So, this unit consists essentially of N memories for the stored packets. Each memory is segmented into fixed size cells (64 bytes : the minimal size of an IP packet). This fragmentation facilate the management of the free space (memoies). These memories are managed by the unit of dynamic memories management.

Dynamic memories management: This unit manages the control of the transmissions memories in the reception/emission of packets. For each packet arrived it must allocate a free space for its storage. Thus for each transmitted packet, it must restore the free space and add it to the transmission memories. Indeed, the use of the parallel treatment to manage the flows of the lines of entries makes it possible to highlight N controllers of the N transmission memories. Each controller is represented by a circular linked list pointed by two pointers of address, a pointer of header addresses and a pointer of tail addresses. On its arrival the packet is stored in the plug of reception to the address indicated by the pointer running of the free addresses. In continuation, the descriptor associated with this packet is related to the file which corresponds to the same class defined by the unit of selector. Instead of storing each address of cell in the file, only two addresses are used for each packet. The first address corresponds to the address of the first cell in the packet and the second indicates that of the last cell. This makes it possible to reduce the cost in memory as well as possible and to optimize the use of its space. Although this organization has a complexity more significant than the others, it was selected for the implementation of the memories management.

This unit is consisted of N circular linked lists (a linked list for each buffer memory) whose each cell contains an address memory for the data storage, an order number of transmission memory and a linked pointer. A controller who selects each time the suitable buffer memory for the storage of the arrived packet.

In reception operations will be realized by this unit are:

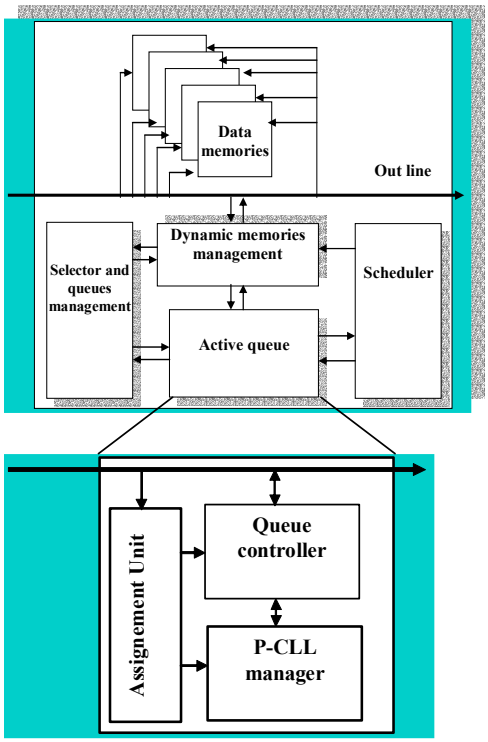

Fig. 2: Architecture of the QoS support block of The IP switch

* determine the length of the packet
* select one of the N buffer memories
* storage of M cells (paquet)
* send the adress of storge of the paquet (head pointer and tail pointer) to the queues unit
* send parameters of the packet (the heading of the packet)
* update of pointers of the of memory management unit

In emission this unit when it recieves the cells of the transmitted packet, determines the suitable memory which store this packet and carries out the update of the pointers of heading and the tail of this memory.

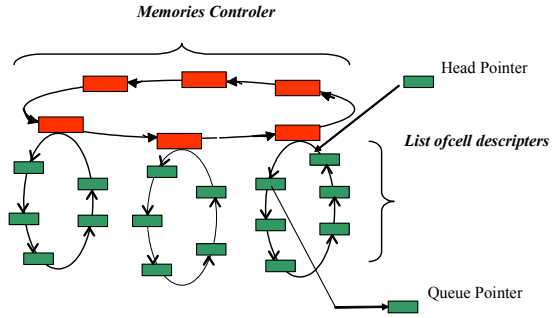

Fig. 3: Structure of the linked list

The linked list is consisted of a list of service which makes it possible to manage the order to be used by the buffer memories pursuing a policy as scheduling. This structure of list forms a logical representation of the distribution of the memory between various connections. The basic idea used by the technique of the linked list consists in managing the memory in a dynamic way. This subdivision is carried out using two types of descriptors to knowing:

Descriptors of heading: These descriptors contain the context of the buffer memory which is identical for all cells of the same memory. The format of this descriptor proposed is made of 2 fields represent the pointers of head and tail. The pointer of head (respectively the pointer of tail) makes it possible to point on the first cell of the plug to fill (respectively the last cell).

Descriptors of cells: The allowance of these structures of data projected on a memory is done in a dynamic way. Once released, these memory capacities must be recovered to be used later on by other cells. During the procedure of initialization, the linked list is built with descriptors. In continuation, this list is updated at each emission/reception of packets. This descriptor contains a pointer of bond making it possible to bind all the cells belonging to the same memory, a field of address of the cell associated in the external plug of storage (dual port memory) and a number of the buffer memory.

Storage in the buffer memory of each cell is made by the descriptor cells which provides the associated address. The descriptors belonging to a memory are chained the ones after the others by the means of a field of bond (pointer of bond). With each list a descriptor with heading is associated which makes it possible to provide the pointers of heading and tail.

Unit of the active queue management: This unit represents the data structure of the scheduling algorithm. This structure depends primarily on the scheduling algorithm witch will be implemented, since it represent its data base. It must be structrured in a way that all the stored packets are visible and accessible by the scheduler so that the operations of decision is fast and effective according to the QoS parametres desired for each flow. As we have to show that the parameters of QoS are optimized by the mechanism of scheduling and memories management. Therefore, creation, acces and release of a flow are in a dynamic way according to the number of declared flows and memory capacity. It is adjustable according to the dynamic parameters of the algorithm implemented in unit of selection and queues management. In the event of congestion of a service class, we borrow a lot of memory space to store packets of the memory capacity of the other classes who are not congested. The number of flows per class is not static, but it is dynamic according to the allocated memory capacity. To meet these needs, the suitable data structure of these architectural choices is the sorted circular linked list (Fig. 4) witch is implemented in active queues management unit (Fig. 2). This unit is consisted of a label assignement unit (of starting priority), a control queue unit and a priority circular linked list (P-CLL) management unit. The assignement unit assigns the incoming packets with a certain priority which is calculated according to the scheduling algorithm to be implemented, in our case the label is the virtual finish time calculated according to WFQ algorithm. The unit of P-CLL manager contains P-CLL priority queues. Each of the element in the P-CLL holds a priority value and it is on these values that the queue is sorted. The queue controller maintains a lookup table with entries corresponding to each piority value. Each entry consists of a pointer to a list of packets of the same priority (same value of virtual finish time). We refer to this list as a priority list. This structure is an implementation per-flow queuing (per-priority queuing), rather than an implementation of priority per class of flow and it is more general to also handle various different priorities in the same way priority for same flow. At every moment the P-CLL contains only the values of active priorities (lists of priority are not empty). Figure 4 presents the architecture of active queue

When a packet needs to be inserted into the queue, the priority assignment unit stamps the packet with a suitable priority value. The queue controller unit

determines whether a priority list already exists for the stamped priority value. If it does it simply adds the new packet to the corresponding priority list. However, if the list does not exist, the queue controller unit creates a new priority list. It also signals the P-CLL manger unit to perform an enqueue operation, which inserts the new priority value into the P-CLL in a sorted manner. This is done to make sure that the highest priority stays at the top of the P-CLL so that when a dequeue of a packet is required, the priority list with the highest priority can be accessed.

When a packet needs to be removed from the queue, the P-CLL manager unit determines the non-empty priority list of highest priority by looking at the topmost element of the P-CLL and sends this priority value to the queue controller unit. The queue controller accesses the corresponding priority list and removes a single packet from it. If this causes the priority list to become empty, the P-CLL manager unit initiates a procedure called dequeue which removes the topmost element from the P-CLL while making sure that the P-CLL remains sorted.

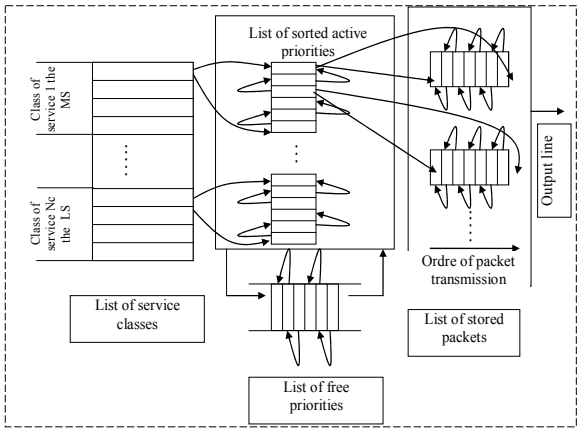

Fig. 4: Structure of the priority circular linked list P-CLL

Unit of selection and queues management: This unit implements essentially two algorithms[18,19].

* An algorithm of selection which determines the class of the packet in progress, by testing the DSCP field of the packet and determines its class. Then this algorithm read the occupancy rates of each class and sends these parameters to the algorithm of queue management. Also, algorithm manage the bandwidth between all classes in a dynamic way. If the average occupancy rate of a class reaches the minimal threshold and one or more classes their occupancy rates average are inferior than the threshold of emprunt, then the algorithm emprunts a free space for this class. If the thresold of emprunt of all classes is equal to threshold max, the algorithm activates the algorithm of queues management to decide the drop or the acceptance of packets.
* An algorithm of queues management: it is the algorithm of queues management proposed by Sally floyd which is RED (Random Early Discard).

Unit of scheduling: This unit implements the algorithm of scheduling which decides the packet that will be extracted from the queues to be outputed. It selects the packet to be transmitted and extracts it from the unit of active queues. Then, it sends towards the unit of memories management the adress of the packet in order to transmit its data and to free the space memory that will be added to the free space by udating the linked list of free space. The scheduling algorithm chosen to be implemented in our architecture is the CBWFQ (Class Based Weighted Fair Queueing), its base of data is structured into three classes of services[20].

CONCLUSION

The growth of the Internet led to a whole of reflexions on the manner of optimizing the use of this network so that the applications can profit from suitable services instead of undergoing overall a policy known as "BEST-EFFORT". Qualities of Service became impossible to circumvent and require the installation of various mechanisms of QoS management at various level of network architecture. Through this study, we presented the concepts of quality of service as well as the components of optimization of the QoS parameters (loss, delay, bandwidth and gigue). This study we guided to propose the architecture of an IP switch optimized for the differentiation of service. This architecture is made up primarily by three blocks which are: the input port, the back plane and the output port. The output port supports the fine grained QoS parameters differentiation. This bloc is composed of buffer memories, dynamic memories management, selection and queue management, scheduling unit and active queue management. The last implements the data structure of the priority circular linked list which optimize the differentiation of the QoS. This bloc, contains the assignement unit, the queue controller and the P-CLL manager unit, represent the data base of the scheduling algorithm and the buffer management that are mechanisms to respect parameters of QoS for each flow. This architecture is an implementation per-flow queuing (per-priority queuing), rather than an implementation of priority per class of flow and it is more general to also handle various different priorities in the same way priority for same flow.

REFERENCES

1. Nick Mckeown, 1997. A fast switched backplane for gigabit switched router. Business Commun. Rev.
2. Shang-Tse Chuang, A. Goel, N. McKeown and B. Prabhakar, 1999. Matching output queueing with a combined input/output-queued switch. IEEE J. on Selected Areas in Commun., 17: 1030-1039.
3. Pankaj, G. and N. McKeown, 2001. Algorithms for packet classification. IEEE Network, 15; 24-32.
4. Bhagwan, R. and B. Lin 2000. Design of a high-speed packet switch for fine-grained quality of service guarantees. IEEE Intl. Conf. Commun.(ICC'00) 2000, New Orleans, 3: 1430-1434.
5. Degermark, M., A. Brodnik, S. Carlsson and S. Pink, 1997. Small forwarding tables for fast routing lookups. Proc. ACM SIGCOMM'97, France, pp: 3-14.
6. Paolo, G., 2002. Queueing and scheduling algorithms for performance routers, thesis de l'université de polytechnique de Torino, Itali.
7. Kam, A.C-K., 2000. Efficient scheduling algorithms for quality of services guarantees in the internet. Thesis de l'institut de technologie de Massachust.
8. Donpaul, C.S., C.R.J. Bennett and Hui Zhang, 1999. Implementing scheduling algorithms in high-speed networks. IEEE J. Selected Areas in Commun., 17: 1145-1158.
9. Varvaja, O.N-M., 2001. étude des algorithmes d'attribution de priorités dans un internet à différentiation de service, thèse de l'université de Renne1, France.
10. Sally, F., R. Gummadi and S. Shenker 2001. Adaptive RED: an algorithm for increasing the robustness of RED's active queue management. Rapport, Longer Technical Report.
11. Sally, F. and V. Jacobson, 1993. Random early detection gateways for congestion avoidance. IEEE/ACM Trans. on Networking, 1: 397-413.
12. Feng, J., G. Rubino and J-M. Bonnin, 2001. A QoS routing algorithm to support classes of service. Conf. Proc. of ICIIEM 2001, Pekin.
13. Ni, N. and L-N. Bhuyan, 2002. Fair scheduling and buffer management in internet router. Conf. on Computer Computing, INFOCOM2002, vol. 3, New York.
14. Gupta, P., S. Lin and N. Mckeown, 1998. Routing lookups in hardware at memory acces speeds. Proc. IEEE INFOCOMM'98, session 10b-1, San Francisco, CA, pp: 1240-1247.
15. Nen-Fu, H. and S.-M. Zhao, 1999. A novel IP-routing lookup scheme and hardware architecture for multigigabit switching routers. IEEE J. on Selected Areas in Commun., 17: 1093-1104.
16. Henry, C.B.C., H.M. Alnuweiri and V.C.M. Leung, 1999. A framework for optimizing the cost and performance of next-generation ip routers. IEEE J. Selected Areas in Commun., 17: 1013-1029.
17. Guest Editorial, 1999. Next-generation of IP switches and routers. Conf. IEEE JSAC, vol. 17.
18. Mahajan, R., S Floyd and D. Wetherall, 2001. controlling high-bandwidth flows at the congested router. 9th Intl. Conf. Network Protocols (ICNP).
19. Huang, N.F. and S.M. Zhao 1999. A novel IP routing lookup scheme and hardware architecture for multigigabit switching routers. IEEE J. on selected Areas in commun., ISACEM, 17: 1093-1104.
20. Henry, H.Y.T. and T. Przygienda, 1999. On fast address-lookup algorithms. IEEE J. Selected Areas in Commun., 17: 1067-1082.