# Network-on-Chip Quality-of-Service through MultiProtocol Label Switching

Manho Kim, Daewook Kim and Gerald E. Sobelman
Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455 USA
Email: {mhkim,daewook,sobelman}@ece.umn.edu

***Abstract*— Providing Quality-of-Service (QoS) in networks-on-chip (NoCs) will be an important consideration for the complex multiprocessor chips of the future. In this paper, we discuss the difficulties encountered in addressing these requirements. Then, we propose a promising solution to this problem that is based on applying the well-known MPLS technology of large-scale computer networks to the on-chip environment. A network simulator is used to evaluate the concept for a typical communications scenario that must support several classes of traffic having a range of QoS requirements.**

## I. Introduction

Systems-on-chip (SoCs) for multimedia or telecommunication applications will contain a large number of processing elements (PEs) such as a DSP processor, RISC CPU, embedded RAM, graphics engine, etc. As a result, there is a need for high-throughput communications links between these blocks. There exist many bus based SoCs which are widely used in industry such as AMBA [1], IBM CoreConnect [2], Pi-Bus [3], etc. However, traditional bus-based SoCs may not be able to meet the scalability, reliability, and high throughput requirements for complex multiprocessor systems of the future. The network-on-chip (NoC) methodology is gaining attention as a promising alternative approach. An NoC breaks the communication bottleneck by applying packet-based switching which is widely used in computer systems and networks. A number of currently proposed NoC solutions have borrowed ideas from parallel computer architecture and computer networks. There are similarities between on-chip communications and Ethernet, ATM, fiber optic networks and wireless network which can be exploited. We can view an NoC system platform as a very complex and multi-protocol network. As such, providing application-wide, end-to-end quality-of-service (QoS) is crucial for optimum system performance. System-wide performance constraints require predictability of inter-block communication and QoS guarantees for the end-to-end communication. QoS is characterized by diverse parameters, such as reliability, delay, jitter, bandwidth, packet loss, and throughput [4]. In this paper, we characterize the system requirements of NoC-based systems and propose a novel architecture for providing QoS on NoCs.

## II. Requirements of Future NoCs

Future SoCs will merge previously different target applications onto a single platform to support computation, communication, multimedia, etc. This convergence make systems susceptible to unpredictable data flows and capacity constraints due to the disparity between average and worst-case data throughput requirements. The essence of QoS is the ability to offer a predictable system behavior to designers. In order to obtain the maximum benefit of the NoC methodology, the ability to provide a guaranteed QoS for an application is a critical requirement [5]. System designers will require NoC communication platforms which have a certain degree of multi-protocol support since the various applications will have a wide variety of communication patterns.

## III. Guaranteeing QoS in NoCs

Most of the existing NoC architectures are packet-switched *(connectionless)* NoCs. They are targeting *Best-Effort (BE)* traffic. Architectures offering only BE services do not reserve bandwidth and hence can have a better average resource utilization, at the cost of unpredictable or unbounded worst-case behavior [5]. They show a reasonable performance on constant bit-rate and variable bit-rate workloads. However, they are not suitable for real-time applications. This shortcoming can be solved by (1) building a connection-oriented communication scheme on top of the packet-switched network (e.g. virtual circuits), or (2) implementing additional services to approximately meet specified QoS parameters (e.g. prioritization of flows) [4]. In most NoC implementations, all traffic types are treated by a network equally and are subject to similar deterioration during network congestion. For large-scale computer networks, QoS schemes were defined for the TCP/IP-based internet. The Integrated Service (*IntServ*) with the Resource Reservation Protocol (*RSVP*) was first introduced, and this was followed by the Differentiated Services (*DiffServ*).

*IntServ* is well suited for reliable real-time communication and provides a connection-oriented distinction between flows. Connection-oriented communication is characterized by resource reservation. This means that flows must set up paths through the network and reserve resources at each network node. The main disadvantages of these systems are non-efficient resource utilization, the overhead introduced by the connection setup and their non-scalability.

*DiffServ* provides different levels of QoS to each class by aggregating traffic into classes, and by scheduling packet forwarding for each class within the network. This results in a connectionless communication, which offers a better adaptation of communication to the varying network traffic and a better utilization of network resources [6].

## IV. MPLS FOR GUARANTEEING QOS ON NOC

The MultiProtocol Label Switching (MPLS) technology has emerged as a connection-oriented protocol serving connectionless internet IP networks, and thus it provides the means for traffic engineering. This means that paths are set up for aggregated flows of a certain type between specified end points of the IP traffic. Also, DiffServ can support a scalable QoS. The combination of these two approaches leads to a scalable hard QoS on internet IP networks because MPLS creates paths that can be traffic engineered [7]. When adapted to NoCs, this approach can be viewed as building a connection-oriented communication on top of the packet-switched on-chip network.

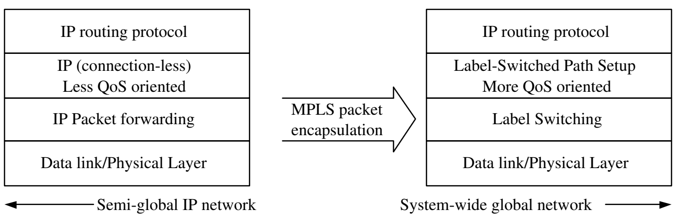


Fig. 1. Simplified protocol stack of IP vs MPLS [8].

Consider how MPLS protocols are organized, as shown in Fig. 1. The left side of the figure shows the widely used packet-switched IP network and the right side of the figure is the MPLS protocol. By encapsulating an MPLS header onto the packet we can construct a QoS-oriented on-chip communication environment.

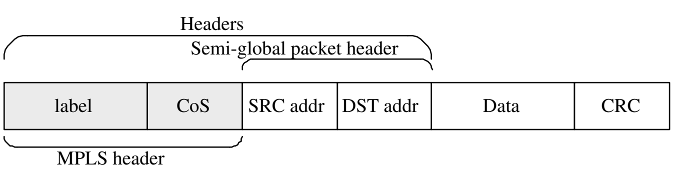


Fig. 2. Frame format.

The frame format of MPLS is as shown in Fig. 2. The *label* field contains the index which is used to identify a Forwarding Equivalence Class (FEC). It is placed between the data link layer header and network layer header. The *CoS* field indicates the *class of service* which is related to QoS. New MPLS [8] headers are added in front of each packet header. The routing is carried out by looking at the label rather than the destination address of the packet. Making the label an index to a table helps to find the correct output line. Packet forwarding is carried out based on the short, fixed length labels which are added on top of the semi-global packet and labels are assigned when a packet enters the global MPLS network. The MPLS router forwards packets based on label value, not on the semi-global packet header information. By doing so, any necessary resources (e.g. bandwidth) along the path can be easily reserved for the corresponding node. This scheme applies a combination of virtual circuit (VC) switching and routing. The main behavior is as follows. First, paths are set up in advance (or by the first packet to reach a given destination), as in circuit switching. Second, the routing algorithm determines the choice of path, as for packet routing. The flows that are grouped together under a single label are said to be in the same forwarding equivalence class (FEC). This class tells where the packets are to go. The FEC packets are treated in the same way.

Note that with traditional virtual-circuit routing, it is not possible to group several distinct paths with different end points onto the same virtual-circuit identifier because there would be no way to distinguish them at the final destination. However, in MPLS routing, the packets still contain their final destination address, in addition to the label. Thus, at the end of the labeled route the label header can be removed and forwarding can continue in the normal way, i.e. by looking at the network layer destination in the semi-global packet header.

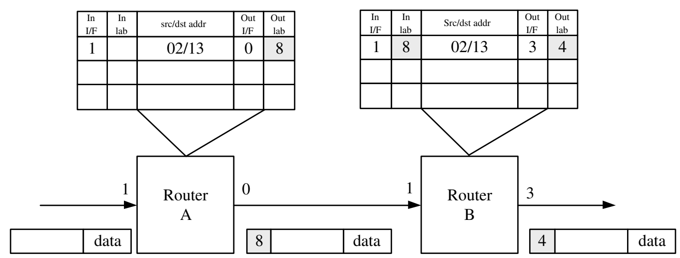


Fig. 3. MPLS example.

Fig. 3 shows the general example of level-switching. Router A classifies the IP packet into an FEC and imposes the corresponding label. Router B forwards labeled packets by comparing the label value against the label table. No packet classification on the FEC is done [9]. We have applied an IP address format header for semi-local communication inside of an SoC and MPLS headers are added when two or more blocks must communicate with each other. At that moment, a global path between them is set up to guarantee QoS. Consider, for example, VoD (Video-on-Demand) service using a mobile device. A substantial amount of bandwidth and associated QoS will be required between the wireless communications unit and multimedia processing unit.

## V. MPLS-NOC ARCHITECTURE

MPLS-NoC Architectures are composed of a Label Switched Router (LSR), a Label-Switched Path (LSP) and labeled packets, as shown in Figs. 4 and 2. The main concept of MPLS is that LSRs forward labeled packets to LSPs.

### A. Label-Edge Router

The Label Edge Router (LER), as shown in Fig. 4, operates at the edge of an MPLS network. It is necessary to include an interface (IF) module to interact with different semi-global SoC/NoC communication channels. LERs route traffic and are used as an interface between layer 2 networks and an MPLS

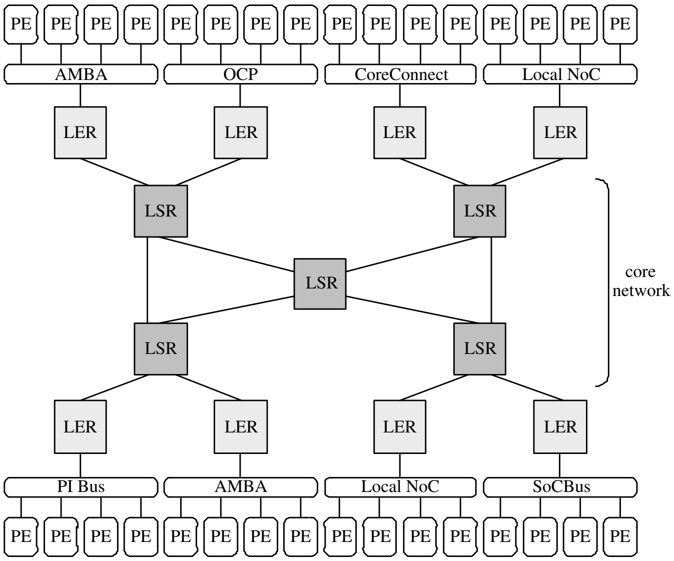


Fig. 4. MPLS NoC platform.

core network. When LERs receive a packet from the semi-global SoC network, a label is attached and the new MPLS packet frame is sent to the MPLS core network. A packet frame will follow a particular path, going from one LER to another. This path is called a label switched path (LSP). When an LER receives a packet from the MPLS network, the label is removed and the packet is sent to the appropriate network. They participate in the establishment of LSPs before exchanging packets [10].

### *B. Label-Switched Router*

A Label Switch Router (LSR) is the main component of the MPLS network. It sets up a path to other MPLS routers and forwards packets to them. They receive packets from an edge router or other LSRs, analyze the label and forward the packet depending on the label contents. Fig. 4 illustrates a typical MPLS network.

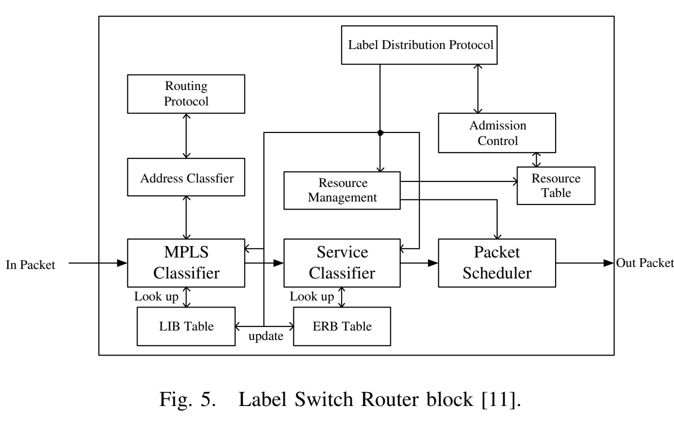


Fig. 5. Label Switch Router block [11].

The conceptual model of MPLS Label-Switched Router supporting QoS is shown in Fig. 5. The Label distribution protocol block generates a label distribution protocol such as CR-LDP [11]. An MPLS classifier executes label operations such as push, pop and swap for an MPLS packet. The service classifier determines the services that should be applied to the incoming packet by using the label and interface information or the CoS field of the MPLS header and associates each packet with the appropriate reservation. The admission control block looks at the traffic parameter of the label distribution protocol and determines whether the MPLS LSR has sufficient available resources to supply the requested QoS. The resource manager creates and deletes queues on demand and also manage the information related to the resource. The packet scheduler manages the packets in the queues so that they receive the required service.

### *C. Label-Switched Paths*

Within an MPLS domain, a path is set up for a given packet to travel on an FEC. The Label-Switched Path (LSP) is set up prior to data transmission. MPLS provides the following two options to set up an LSP. 1) *hop-by-hop routing*: each LSR independently selects the next hop for a given FEC. The LSR uses any available routing protocol. 2) *explicit routing*: This is similar to source routing. The LSR specifies the list of nodes through which the LSP traverses.

## VI. MPLS-NoC Modeling

In this study, we have modeled our MPLS-NoC architecture concepts with the widely used network simulator *ns-2* [12]. This tool has been widely applied in research related to the design and evaluation of computer networks and to evaluate various design options for NoC architectures [13], including the design of routers, communication protocols, etc.

### *A. Case Study*

We have simulated a prototype MPLS-NoC. The simulation scenario is as follows. Multimedia systems, wireless communication systems, network security systems, and general purpose computation processors are located on-chip as shown in Fig. 6. A mesh-based wireless communication processing network sends packets to an MPEG-4 processing network. The wireless communications network has 5 PEs which have to send packets to the MPEG-4 PEs and they have different priorities. Also, the computation processing network has 1 PE which wants to send packets to the MPEG-4 PEs. This can be simulated by setting up 6 traffic sources and 6 traffic sinks which are connected with 5 LSRs in *ns-2*. The traffic sources have three different priorities. PE6 and PE5 generate real time traffic every 15 seconds with highest priority, PE4 generates constant bit rate traffic every 6 seconds with medium priority and PE1, PE2, PE3 generate traffic during the entire simulation time with the lowest priority.

The simulation results as shown in Fig. 7 show that the MPLS-NoC maintains priority levels with efficient bandwidth utilization.

## VII. Conclusions and Future Work

In this paper, we have discussed the major research challenges in future NoC platforms related to providing QoS and

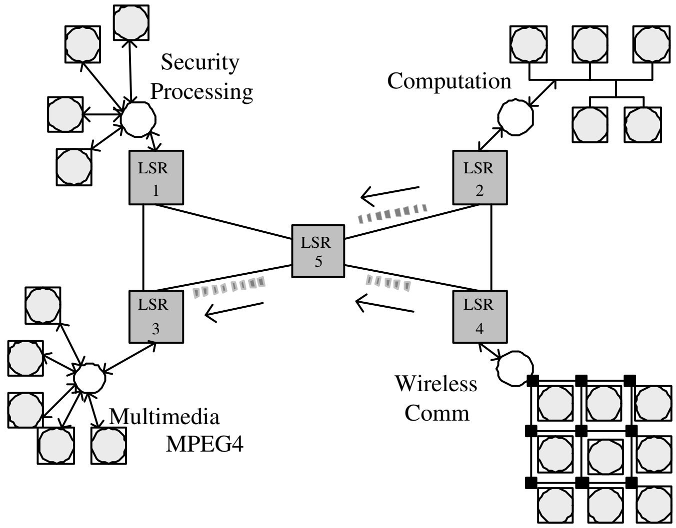


Fig. 6. Complete system on NoC.

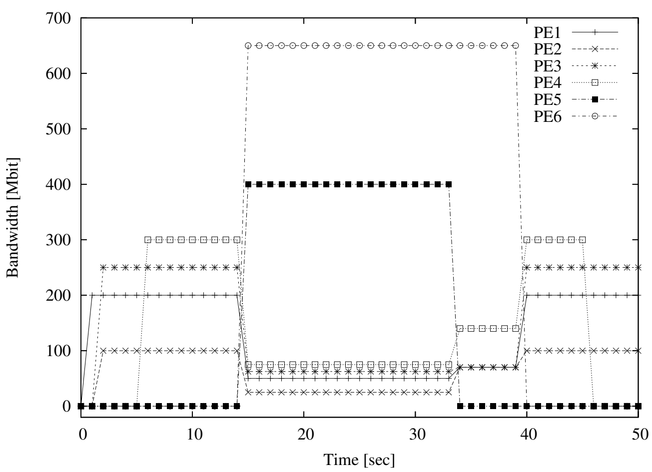


Fig. 7. QoS traffic simulation.

supporting MultiProtocol traffic. As a solution for this, we have presented a novel NoC communication platform that supports several QoS and multi-protocol concepts which were adapted from the MultiProtocol-Level Switching technique with DiffServ. We simulated our MPLS-NoC architecture using the widely used network simulator *ns-2* and have obtained good performance. However, because the original MPLS protocol is targeted at large-scale, wide area networks, its complexity of implementation will be high. Therefore, further research should be done in order to find additional simplifications and improvements to the methodology.

## Acknowledgments

The authors would like to thank Sang Woo Rhim, Bumhak Lee and Euiseok Kim of Samsung Advanced Institute of Technology (SAIT) for their help with this manuscript. This research work is supported by a grant from SAIT.

## References

[1] J. Park, I. Kim, S. Kim, S. Park, B. Koo, K. Shin, K. Seo, and J. Cha, "MPEG-4 video codec on an ARM core and AMBA," in *Proc. of Workshop and Exhibition on MPEG-4*, Jun. 2001, pp. 95–98.

[2] R. Hofmann and B. Drerup, "Next generation CoreConnect processor local bus architecture," in *15th Annual IEEE International ASIC/SOC Conference*, Sep. 2002, pp. 221–225.

[3] C. Roark and F. Jackson, "New developments in a PI-Bus specification by the JIAWG and SAE," in *Proc. of the IEEE National Aerospace and Electronics Conference*, vol. 2, May 1992, pp. 760 – 766.

[4] M. Harmanci, N. Escudero, Y. Leblebici, and P. Ienne, "Quantitative modelling and comparison of communication schemes to guarantee quality-of-service in networks-on-chip," in *IEEE International Symposium on Circuits and Systems*, vol. 2, May 2005, pp. 1782–1785.

[5] K. Goossens, J. Dielissen, J. van Meerbergen, P. Poplavko, A. Rdulescu, E. Rijpkema, E. Waterlander, and P. Wielage, "Guaranteeing the quality of services in networks on chip," pp. 61–82, 2003.

[6] M. Harmanci, N. Escudero, Y. Leblebici, and P. Ienne, "Providing QoS to connection-less packet-switched NoC by implementing DiffServ functionalities," in *Proc. of the International Symposium on System-on-Chip*, Nov. 2004, pp. 37–40.

[7] V. Fineberg, C. Chen, and X. Xiao, "An end-to-end QoS architecture with the MPLS-based core," in *IEEE Workshop on IP Operations and Management*, 2002, pp. 26–30.

[8] J. Lawrence, "Designing multiprotocol label switching networks," *IEEE Communications Magazine*, vol. 39, no. 7, pp. 134–142, Jul. 2001.

[9] www.cisco.com

[10] R. Peterkin and D. Ionescu, "Embedded MPLS Architecture," in *Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium*, Apr. 2005, pp. 170a–170a.

[11] G. Ahn and W. Chun, "Design and implementation of MPLS network simulator (MNS) supporting QoS," in *Proc. of the 15th International Conference on Information Networking*, 31 Jan. - 2 Feb. 2001, pp. 694–699.

[12] www.isi.edu/nsnam/ns

[13] R. Lemaire, F. Clermidy, Y. Durand, D. Lattard, and A. Jerraya, "Performance Evaluation of a NoC-Based Design for MC-CDMA Telecommunications Using NS-2," in *The 16th IEEE International Workshop on Rapid System Prototyping*, Jun. 2005, pp. 24–30.