Out-of-Sequence Prevention for Multicast Input-Queuing Space-Memory-Memory Clos-Network

Yu, Hao; Ruepp, Sarah Renée; Berger, Michael Stübert

Published in:
I E E E Communications Letters

Link to article, DOI:
10.1109/LCOMM.2011.051011.102535

Publication date:
2011

Out-of-Sequence Prevention for Multicast Input-Queuing Space-Memory-Memory Clos-Network

Hao Yu, Sarah Ruepp, Michael S. Berger, *Member, IEEE*

***Abstract*—This paper proposes two cell dispatching algorithms for the input-queuing space-memory-memory (IQ-SMM) Clos-network to reduce out-of-sequence (OOS) for multicast traffic. The frequent connection pattern change of DSRR results in a severe OOS problem. Based on the principle of DSRR, MF-DSRR is able to reduce OOS but still suffers from it under high traffic load. MFRR maintains the connection pattern separately for each input and can eliminate the in-packet OOS and thus significantly reduces the reassembly buffer size and delay.**

***Index Terms*—Clos-network, out-of-sequence, multicast scheduling, multistage, cell dispatching.**

I. Introduction

MULTICAST switching in crossbar fabrics have been studied by various publications [1]-[4]. However the scalability is limited by the growth of number of cross points. The Clos-network, in contrast, is able to reduce the number of cross points by using multiple switching stages and thus is more scalable and cost-effective. The ATLANTA [5] switch constitutes a commercially successful example using Clos-network to provide high-speed switching. Various scheduling algorithms for unicast have been developed for different Clos-network architectures in terms of buffer allocation, such as Distro [6] for space-space-space (S^3), CRRD/CMSD [7] and MWMD [8] for memory-space-memory (MSM), and DSRR [9] for space-memory-memory (SMM).

Due to the cell contention problems of S^3 and MSM, and the implementation cost of the fully buffered memory-memory-memory (MMM) architecture, the SMM architecture with desynchronized static round-robin (DSRR) is proposed and evaluated in [9] for unicast. The study demonstrates that SMM achieves 100% throughput for DSRR under admissible unicast traffic. The SMM in [9] uses output-queuing (OQ) for both buffered stages, where speed-up is required, which hinders the scalability. To the best of our knowledge, SMM architecture has not been evaluated under multicast traffic. This paper proposes an input-queuing SMM (IQ-SMM) architecture for multicast, where no speed-up is required. When considering multicast, DSRR can result in a serious out-of-sequence (OOS) problem, causing different delays within a multicast group. The OOS problem can be categorized into two types: *inter-packet* and *in-packet*. Inter-packet OOS means cells of different packets are disordered, and in-packet OOS implies cells of the same packet are disordered. The motive behind this subdivision is to further analyze cell dispatching (CD) schemes and to differentiate their performances. This paper proposes two CD schemes, i.e. Multicast flow-based DSRR (MF-DSRR) and Multicast flow-based round-robin (MFRR) to reduce the OOS problem.

Manuscript received December 22, 2010. This work was supported in part by the Danish Advanced Technology Foundation in the project *The Road to 100 Gigabit Ethernet*.

Hao Yu, Sarah Ruepp, and Michael S. Berger are with the Technical University of Denmark, Kgs. Lyngby, 2800, Denmark. (emails: {haoyu, srru, msbe}@fotonik.dtu.dk)

II. Model of the Space-Memory-Memory Clos-Network

The Clos-network switch consists of three stages of switching elements (SE) and is denoted as $C(n,m,r)$. The switch model has r input/output modules (IM/OM) of size $n \times m$, and m central modules (CM) of size $r \times r$ as shown in Fig. 1. Each IM/OM has n connections to line cards and m interstage connections to CMs. There exists only one interstage connection between an IM/OM and a CM. The number of input/output ports of the switch is $N = nr$.

We assume that a first-in-first-out (FIFO) queue, where multicast packets are temporally stored, is installed before each input of IMs. Variable-length packets are assumed to be segmented into fixed-size cells before entering IMs, and to be reassembled after traversing three stages. We also assume that each packet carries a fan-out vector $\boldsymbol{b} = \langle b_j \rangle, b_j \in \{0,1\}, 0 \leq j \leq N-1$, where $b_j = 1$ indicates the packet is bound for the j^{th} output, else $b_j = 0$. Cells generated from the same packet have the same fan-out vectors. The FIFO queue is assumed to

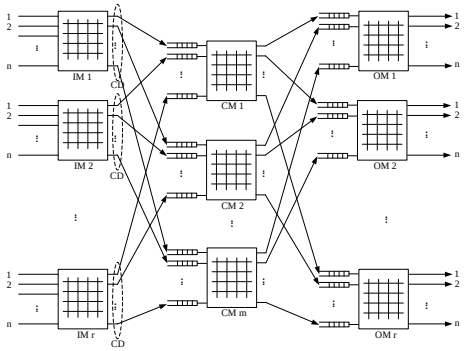

Fig. 1. Space-memory-memory Clos network with input-buffer at central and output switching elements

be able to examine the fan-out vector of each packet and inform the switch fabric of any fan-out change. IMs forward incoming cells to CMs according to cell dispatching algorithms. Since the IQ-SMM architecture is used, CMs and OMs are input-buffered crossbar switches, therefore no speed-up is required. Each CM/OM has r FIFO input FIFO queues, each of which is connected to one interstage link. The following notations are used throughout this paper:

IM_i	i^{th} *input module, where* $1 \le i \le r$
CM_k	k^{th} *central module, where* $1 \le k \le m$
OM_j	j^{th} *output module, where* $1 \le j \le r$
$I_{i,p}$	p^{th} *input of* IM_i, *where* $1 \le p \le n$
$O_{j,q}$	q^{th} *output of* OM_j, *where* $1 \le q \le n$
$Q_{C_{i,k}}$	i^{th} *input queue of* CM_k *connected to* IM_i
$Q_{O_{k,j}}$	k^{th} *input queue of* OM_j *connected to* CM_k
$IL_{i,k}$	*the interstage link connecting* IM_i *and* CM_k
$CL_{k,j}$	*the interstage link connecting* CM_k *and* OM_j

We assume that multicast capability is implemented in CMs and OMs. A bit-cluster $\boldsymbol{c}_d$ is defined as a set of n bits in a fan-out vector: $\boldsymbol{b} = \langle \boldsymbol{c}_d \rangle, 0 \le d \le r-1$, and $\boldsymbol{c}_d = \langle b_j \rangle, nd \le j \le nd + n - 1$.We define $|\boldsymbol{c}_d| = \min\left(1, \sum b_j\right)$. CM_k examines r bit-clusters of the fan-out vector of a cell, and if a bit-cluster is not zero, i.e. $\boldsymbol{c}_d \ne \boldsymbol{o}$, the CM_k sends a copy of the cell to OM_d.

III. Cell Dispatching Algorithms

Discussed in [9], DSRR runs independently in each IM and connects each input to all outputs in a round-robin fashion. The connection pattern is changed after each cell time as shown in Fig. 2, resulting in a well balanced distribution of cells to the CMs. However, using the IQ-SMM architecture shown in Fig. 1, this causes a serious OOS problem. Independently treating the incoming cells is not optimal, since same-packet cells should be kept in a sequential order.

A. Multicast flow-based DSRR (MF-DSRR)

Instead of changing the connection pattern after each cell time as DSRR, MF-DSRR modifies the IM connection pattern each time when a change of received fan-out vector is detected. Take a 4×6 IM for example, the connection pattern of IM_i can initially be $(I_{i,1} \to IL_{i,1},\ I_{i,2} \to IL_{i,2},\ I_{i,3} \to IL_{i,3},\ I_{i,4} \to IL_{i,4})$, as shown in Fig. 3. Each input is serving a multicast flow (cells of the same fan-out vectors). When a change of fan-out vector from f_1 to f_1' is detected on $I_{i,1}$, each input maps the connection to the next output in a round-robin manner as $(I_{i,1} \to IL_{i,2},\ I_{i,2} \to IL_{i,3},\ I_{i,3} \to IL_{i,4},\ I_{i,4} \to IL_{i,5})$. Further changes in the received fan-out vectors will configure the connection pattern to $(I_{i,1} \to IL_{i,3},\ I_{i,2} \to IL_{i,4},\ I_{i,3} \to IL_{i,5},\ I_{i,4} \to IL_{i,6})$ and so on.

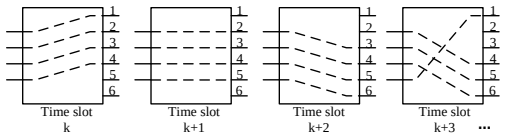

Fig. 2. Desynchronized static round robin connection pattern (DSRR)

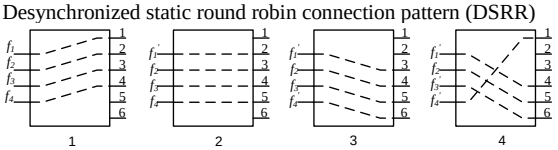

Fig. 3. Multicast flow-based DSRR cell dispatching (MF-DSRR)

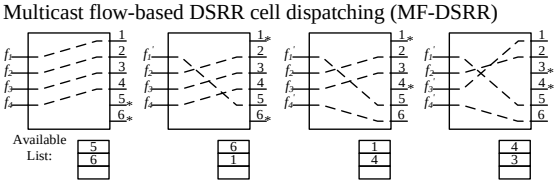

Fig. 4. Multicast flow-based round robin cell dispatching (MFRR)

B. Multicast flow-based round-robin (MFRR)

MFRR is independently run in each IM. Each input monitors the change of received fan-out vectors. The *AvailableList* is used to record the idle outputs as its elements and the number of elements is thus $(m-n)$. Elements can only be popped from the top and inserted to the bottom. When a change of fan-out vector is detected by $I_{i,p}$, one output is popped from the *AvailableList* and the connection of $I_{i,p}$ is changed to that output. Meanwhile, the output that is released by $I_{i,p}$ is inserted to the bottom of the list. If changes of fan-out vectors are detected at the same time, ties are broken randomly. Considering the 4×6 switch, the connection pattern of IM_i can initially be $(I_{i,1} \to IL_{i,1},\ I_{i,2} \to IL_{i,2},\ I_{i,3} \to IL_{i,3},\ I_{i,4} \to IL_{i,4})$ with $\{5,6\}$ in the *AvailableList*, as shown in Fig. 4. When an incoming cell with a different fan-out vector is detected by $I_{i,1}$, it connects to $IL_{i,5}$ and inserts $IL_{i,1}$ to the list, which becomes $\{6,1\}$. The connection pattern becomes $(I_{i,1} \to IL_{i,5},\ \ I_{i,2} \to IL_{i,2},\ \ I_{i,3} \to IL_{i,3},\ \ I_{i,4} \to IL_{i,4})$. Upon detecting another change, $I_{i,4}$ connects to $IL_{i,6}$ and inserts $IL_{i,4}$ to the list. The connection pattern thus becomes $(I_{i,1} \to IL_{i,5},\ I_{i,2} \to IL_{i,2},\ I_{i,3} \to IL_{i,3},\ I_{i,4} \to IL_{i,6})$.

IV. Analysis and Simulation Results

A. Out-of-Sequence Probability Analysis

Assume the traffic to each $I_{i,p}$ is an *i.i.d. Poisson arrival process* with arrival rate of λ. Variable-length packets are segmented into L fixed-size cells before entering IMs, where L is a random variable uniformly distributed with $E(L) = \bar{L}$. Each packet is bound for a destination with a probability of p, i.e. $P(b_j = 1) = p$. Since a packet is bound for at least one destination, the fan-out $F \triangleq \sum b_j, \forall j$ has $P(F = f) = \frac{\binom{N}{f} p^f (1-p)^{N-f}}{1-(1-p)^N}$ and $E(F) = \frac{Np}{1-(1-p)^N}$. Since CMs observe the fan-out vectors in bit-clusters, the fan-out seen by CMs becomes $P(F_C = f) = \frac{\binom{r}{f} [1-(1-p)^n]^f [(1-p)^n]^{r-f}}{1-(1-p)^N}$ and $E(F_C) = \frac{r[1-(1-p)^n]}{1-(1-p)^N}$, where $F_c \triangleq \sum |\boldsymbol{c}_d|, \forall d$. All traffic is admissible, which means no input or output port is oversubscribed. The total traffic load of all outputs is $N\lambda E(F)$, and the offered load seen on each output is $\lambda E(F)$. For $I_{i,p}$ under the MF-DSRR scheme, the probability of a connection pattern change is:

$$P(j, \bar{L}) = \binom{\bar{L}}{j} (\dot{\lambda})^j (1-\dot{\lambda})^{\bar{L}-j} \quad (1)$$

where $\dot{\lambda} = (n-1)\lambda$, $j = 0,1,2 \ldots$ is the number of connection pattern changes, and $\bar{L}$ is the mean cell time for $I_{i,p}$ to complete the transmission of a packet. Since the connection

pattern resumes after m changes, we thus have the probability that same-packet cells are distributed to different CMs:

$$1-\sum_{\theta} P(\theta,\bar{L}) = 1-\left(1-\dot{\lambda}\right)^{\bar{L}}\left[1+\binom{\bar{L}}{m}\left(\frac{\dot{\lambda}}{1-\dot{\lambda}}\right)^{m}+\cdots\right] \quad (2)$$

where $\theta = 0, m, 2m, \ldots$ For $I_{i,p}$ under the MFRR scheme, the cells of the same packet are always sent to the same CM regardless of the changes of other inputs. Therefore the probability that same-packet cells are distributed to different CMs becomes 0. With the *AvailableList*, the time complexity for each input to find the idle output becomes $O(1)$.

B. Simulation Results

Comparisons between Static, DSRR, MF-DSRR, and MFRR in a $C(4,7,4)$ IQ-SMM switch are carried out in OPNET Modeler [10]. The scheduling algorithm proposed in [2] is used in the CMs and OMs. The *Static* scheme does not change the connection patterns in IMs and is thus used as a reference. Admissible traffic with $E(F) = 4$ and $\bar{L} = 12$ is provided to each input. Fig. 4 compares the percentage of total OOS cells including both inter-packet and in-packet OOS in all the cells received. As for the in-packet OSS cells, DSRR performs the worst, causing many in-packet OSS cells due to its frequent connection pattern change. MFRR has no in-packet OOS cells and outperforms the other two. MF-DSRR can reduce the in-packet OOS but cannot completely eliminate it due to the non-zero probability in (2) that cells of the same packet are distributed to different CMs. The three schemes have similar performance in inter-packet OOS but MFRR still outperforms the others. Fig. 5 compares the reassembly buffer size. MFRR can significantly reduce the buffer size and performs close to Static. Fig. 6 shows the cell delays. DSRR outperforms all the others because of its load balancing feature, evenly distributing cells to CMs. However, DSRR can cause serious reassembly delay which is about 75% of the mean packet transmission time under high load illustrated in Fig. 7. MFRR and MF-DSRR both reduce the reassembly delay and MFRR has the best performance.

V. Conclusion

This paper proposes an input-queuing space-memory-memory (IQ-SMM) Clos-network architecture for multicast with two cell dispatching schemes, MF-DSRR and MFRR. MF-DSRR has a low implementation complexity. MFRR requires independent controllers on each input and achieves a low complexity by proper design. Simulation results show that MF-DSRR is able to reduce OOS, which is a seious problem from DSRR, but still suffers from it under high traffic load. MFRR can eliminate the in-packet OOS and thus significantly reduces the reassembly buffer size and delay.

References

[1] B. Prabhakar, N. McKeown, and R. Ahuja, "Multicast scheduling for input-queued switches", *IEEE J. on Selected Areas in Commun.*, vol. 15, no. 5, Jun. 1997.

[2] H. Yu, S. Ruepp, and M. S. Berger, "A novel round-robin based multicast scheduling algorithm for 100 Gigabit Ethernet switches", in *Proc. IEEE INFOCOM SW*, 2010.

[3] D. Pan, and Y. Yang, "FIFO-based multicast scheduling algorithm for virtual output queued packet switches", *IEEE Trans. Comput.*, vol. 54, no. 10, Oct. 2005.

[4] S. Sun, S. He, Y. Zheng, and W. Gao, "Multicast scheduling in buffered crossbar switches with multiple input queues", in *Proc. HPSR*, 2005.

[5] F. M. Chuissi, J. G. Kneuer, and V. P. Kumar, "Low-cost scalable switching solutions for broadband networking: the ATLANTA architecture and chipset", *IEEE Comm. Mag.*, vol. 35, pp. 44-53, 1997

[6] K. Pun and M. Hamdi, "Distro: A distributed static round-robin scheduling algorithm for bufferless Clos-network switches", in *Proc. GLOBECOM'02*, pp. 2298-2302, vol. 3, Nov. 2002

[7] E. Oki, Z. Jing, R. Rojas-Cessa, and J. Chao, "Concurrent round-robin-based dispatching schemes for Clos-network switches", *IEEE Trans. On Networking*, pp. 830-844, vol. 10, issue 6, Dec. 2002.

[8] R. Rojas-Cessa, E. Oki, and J. Chao, "Maximum weight matching dispatching scheme in buffered Clos-Network packet switches", in *Proc. ICC'04*, pp. 1075-1079, vol. 2, Jun. 2004.

[9] X. Li, Z. Zhou, and M. Hamdi, "Space-memory-memory architecture for CLOS-network packet switches", in *Proc. ICC'05*, vol. 2, pp. 1031-1035, 2005

[10] OPNET Modeler, available at: http://www.opnet.com/

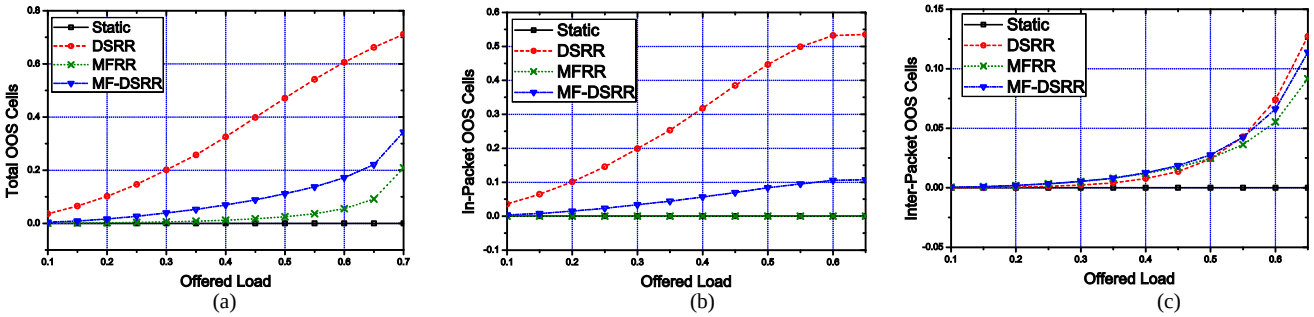

Fig. 4. Percentage of OOS cells in all received cells. (a) total OOS, (b) in-packet OOS, (c) inter-packet OOS

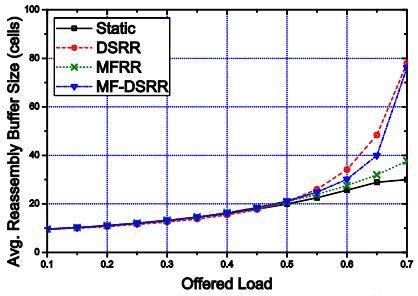

Fig. 5. Average size of reassembly buffer

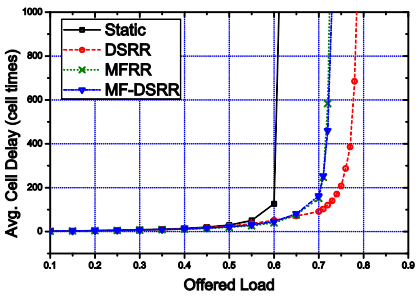

Fig. 6. Average cell delay

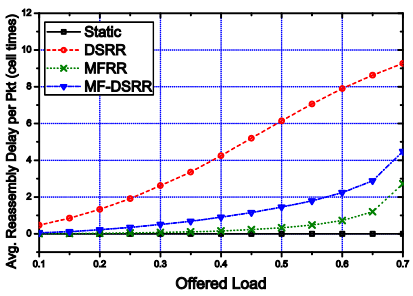

Fig. 7. Average reassembly delay per packet