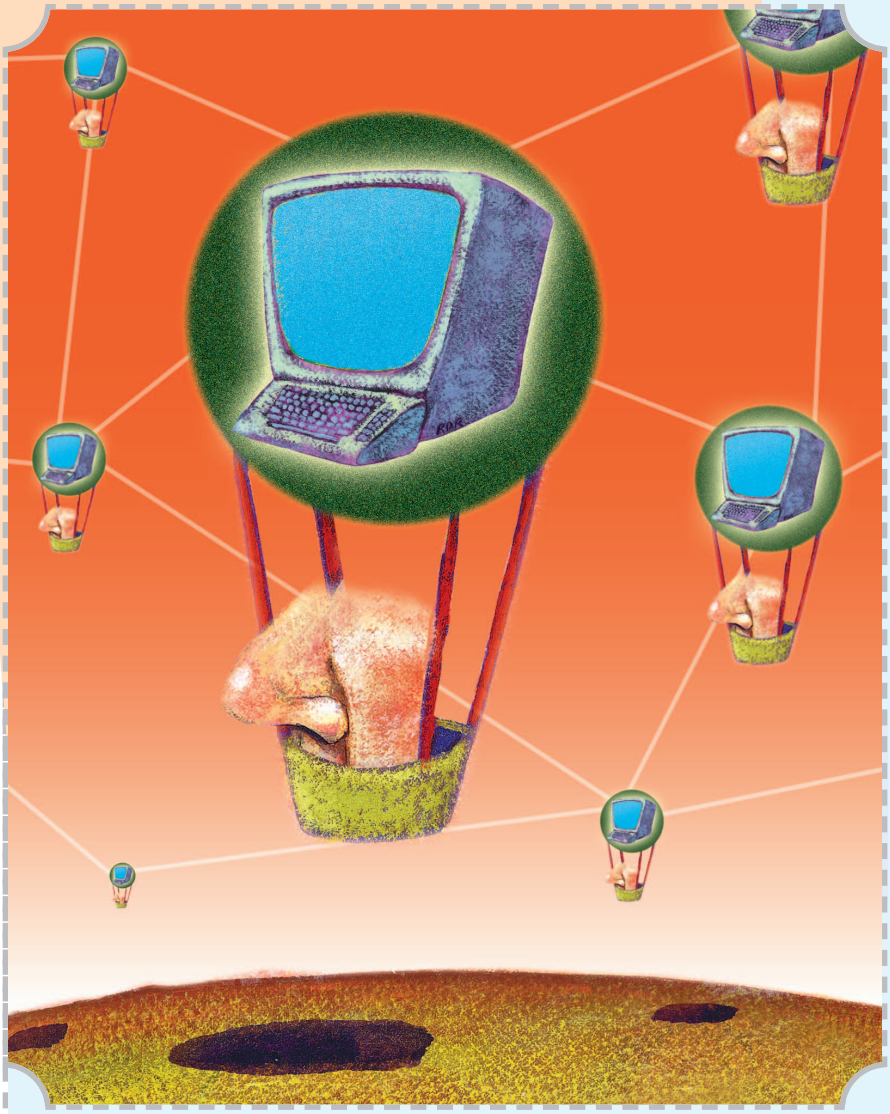

© MASTER SERIES

Decongestants for clogged networks

Pedro J. García, José Flich, José Duato, Ian Johnson, Francisco J. Quiles, and Finbar Naven

Digital Object Identifier 10.1109/MPOT.2007.906118

INTERCONNECTION NETWORKS ARE A KEY ELEMENT IN a wide variety of systems: massive parallel processors, local and system area networks, clusters of PCs and workstations, and Internet Protocol routers. They are essential to high performance in the form of high-bandwidth communications, with low latency, "quality of service" (guaranteed service levels), efficient switching, and flexibility of network topology, as embodied in Myrinet, InfiniBand, Quadrics, Advanced Switching, and similar interconnects.

But, despite all the advances that modern interconnects offer, congestion is a growing problem as "lossless" interconnection networks—those that do not allow data packets to be discarded—come to the fore. Congestion can degrade network performance drastically.

CONGESTION IN LOSSLESS NETWORKS

Congestion originates in contention, the situation that occurs when several flows of packets simultaneously request access to the same network resource (typically, a switch output port). If the internal speedup of switches is not enough for attending to several requests at the same time, the access to the requested output port will be granted to only one packet, while the rest must wait. (Switch speedup is the maximum speed at which packets can be forwarded from input to output ports, relative to link speed.) Figure 1(a) shows a contention situation caused by two incoming flows requesting the same output port at Switch C. In the figure, the switch speedup is 1, and the requested output port may be connected to an end node or to another switch.

When contention persists, congestion appears. The buffers containing the blocked packets become filled and the flow control, in lossless networks, then prevents other switches from sending packets to the congested ports. Although flow control is essential to avoid discarding packets, it rapidly propagates congestion to other switches, as packets stored at some of their ports also become blocked.

Figure 1(b) shows a congestion situation that results from the contention in Fig. 1(a). The network is a lossless one, so packets are not discarded when buffers are full, and the sources of the contending flows continuously inject packets into the network. These congestion-inducing flows are usually referred to as "hot flows."

Congestion progressively spreads in this way through the network, even reaching the end nodes that are injecting hot flow packets. All the network resources affected by the spreading of congestion are commonly known as a "congestion tree." In a congestion tree, the "root" is the point where the hot flows causing congestion finally meet, the "branches" are series of consecutive congested points along any path followed by hot flows, and the "leaves" are the ending points of each branch. Figure 1(c) shows the final

0278-6648/07/$25.00 © 2007 IEEE

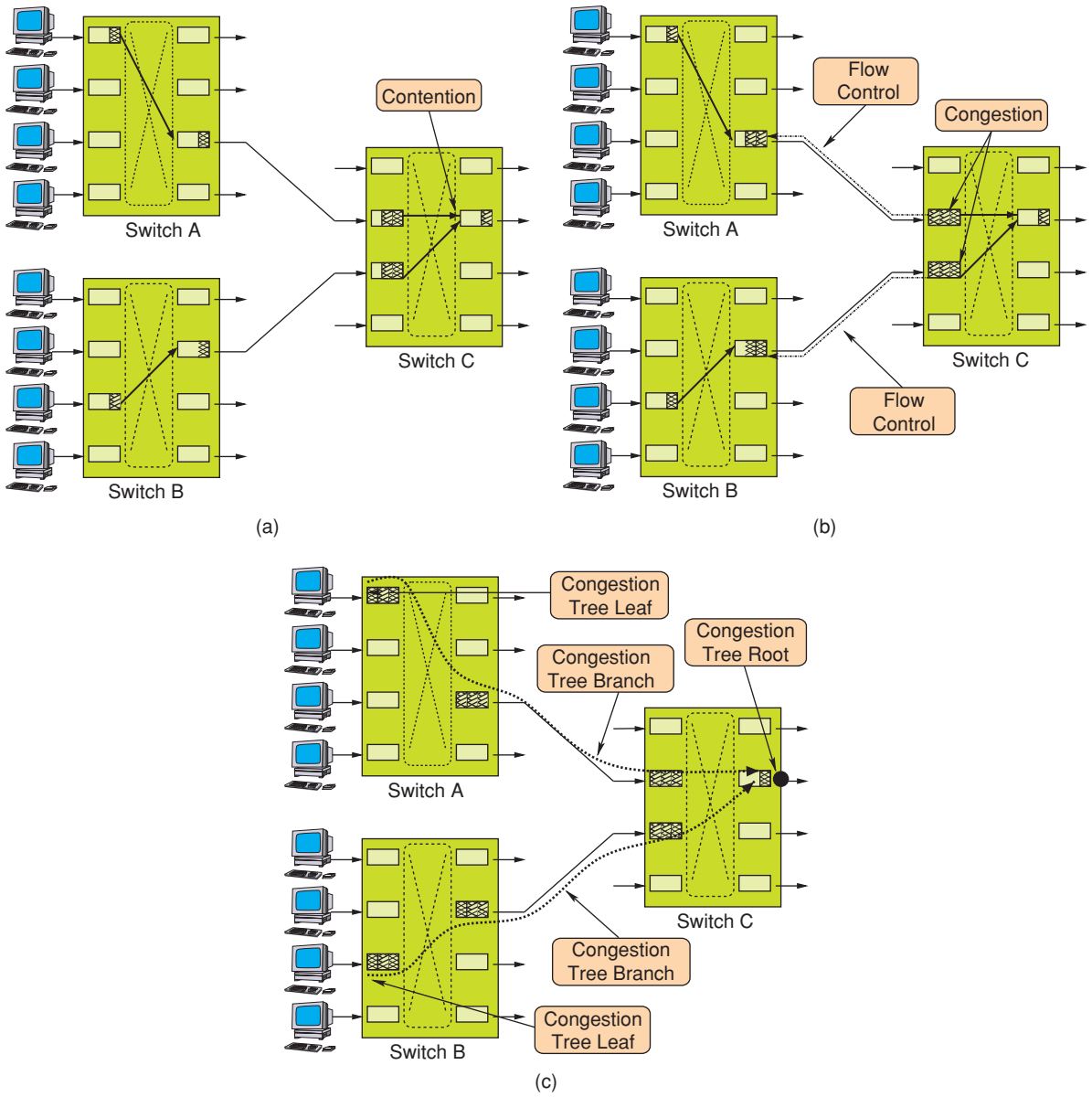

Fig. 1 Contention and congestion in switches of an interconnection network

congestion tree that is formed from the congestion situation.

The tree in Fig. 1(c) is a very simple example; in the real world, congestion trees may behave in complexly dynamic ways. In fact, congestion trees may evolve in very different ways, depending on traffic patterns and switch architecture. For example, a congestion tree may grow from leaves to root and vice versa. Several congestion trees may grow independently and later merge, and it is even possible that some trees completely overlap while remaining independent.

Regardless of the way a congestion tree forms, its existence in a network severely degrades network performance. For instance, Fig. 2(a) shows what congestion does to network throughput for a bidirectional multistage interconnection network with 64 sending and 64 receiving end nodes (a 64×64 BMIN). Fig. 2(b) shows average packet latency for the network under the same conditions. The congestion tree has been generated by hot-spot traffic from several sources to the same destination during a very short time interval (less than a millisecond, as indicated in the figure). Of course, the root of the tree is at the hot-spot destination. The rest of the end nodes send only traffic addressed to random destinations.

The network throughput not only drops severely (by approximately 70%) when congestion appears, but it also remains low well beyond the end of the hot-spot traffic. As for latency, it increases from a few hundreds of nanoseconds (before congestion) to several hundreds of microseconds (three orders of magnitude). Latency eventually returns to its usual low values, but only after several milliseconds.

THE CULPRIT: HEAD-OF-LINE (HOL) BLOCKING

Although the effects of congestion trees on network performance are clear, their mechanism is not so obvious. The situation depicted in Fig. 3 will help explain the mechanism. In the figure, four sending end nodes (sources s1, s2, s3, and s4) inject packets at the maximum speed allowed by the network at any time, while two receiving end nodes (destinations d1 and d2) accept the flows. Three sources (s1, s2, and s3) inject packets addressed to the same

destination (d1), forming a congestion tree whose root is at the output port connected to d1. Although another flow exists in the network (from s4 to d2), it's not a hot flow (so it's a "cold flow"), since packets belonging to this flow do not cross the root of the congestion tree.

The three hot flows meet at switch 8, thus they share the bandwidth of the link that connects the root point to d1. Therefore, assuming a fair arbitration from the switch scheduler, each hot flow will cross switch 8 at a speed equal to one-third, or 33%, of the maximum link rate. Moreover, as a result of the spreading of congestion by means of flow control, all the packets along the congestion tree branches will advance at the same speed, and finally even the sources will be forced to reduce their injection down to this rate. Figure 3 indicates the link utilization percentages when the network reaches this "stable" state.

The utilization of the link that connects the hot-spot destination (d1) to the network is at its maximum (100%), so it's not possible to improve network throughput at this point. In fact, network throughput will be maximum if only hot flows exist. This means that the mere existence of the congestion tree does not justify by itself the network throughput degradation observed in congested networks. On the contrary, note that the link leading to d2 is used only at 33% of the maximum link rate. However, there is no contention for this link, since only one flow requests to access it.

So why isn't this link used at full rate? The reason is that the cold flow requesting this link follows a path partially affected by the congestion tree, in such a way that packets belonging to the cold flow share some network resources (specifically, a link and a queue) with hot flow packets. Because of this sharing, cold flow packets advance at the same speed as hot flow ones. Without this interaction between cold and hot packets, the link leading to d2 would be used at maximum link rate.

Therefore, the real problem associated to congestion is that packets belonging to hot flows may prevent other packets from advancing at the injection rate, even if they belong to cold flows. This phenomenon is known as head-of-line (HOL) blocking, and, in general, it happens when a packet ahead of a first in, first out (FIFO) queue blocks, preventing the rest of packets in the same queue from advancing.

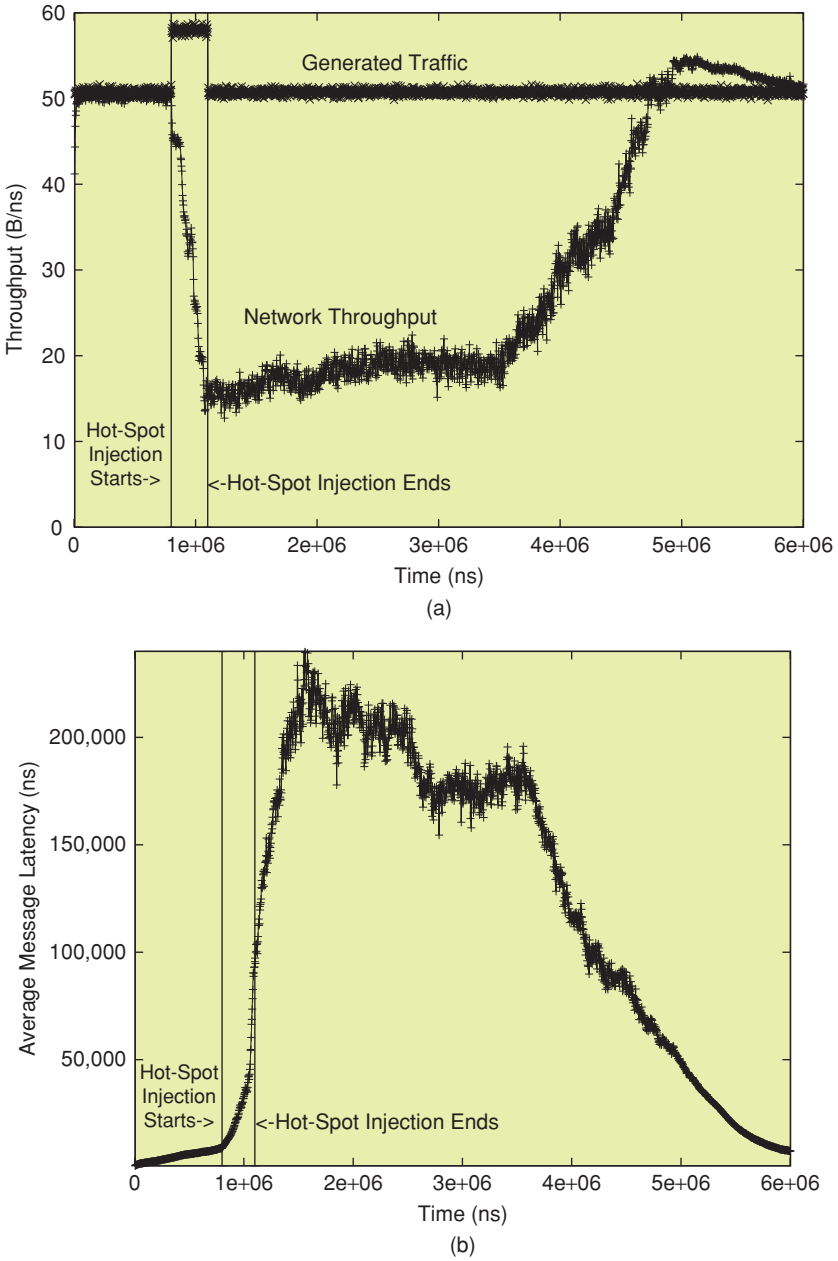

Fig. 2 Network performance drops after congestion appears in a 64 × 64 bidirectional multistage interconnection network (BMIN)

The real cause of the performance degradation that networks suffer during congestion is the HOL blocking that hot flow packets present to cold flow packets. Of course, the wider the congestion tree, the greater the probability of HOL blocking by hot flow packets.

Surprisingly, the key role of HOL blocking in congestion is not considered by many congestion management strategies. The most effective strategies, however, do take HOL blocking into account: VOQnet, VOQsw, and RECN. Of them, RECN is a clear winner. These strategies (and their acronyms) are described here.

FACING CONGESTION AND HOL BLOCKING

Managing congestion has been an open issue in lossless interconnection networks for the last 20 years. An obvious way to minimize congestion is to use many more network components than the minimum required for connecting all the system end nodes—that is, to overdimension the network. The resulting network will be underutilized, reducing so the probability of congestion.

However, this solution is not practical because of the high cost and power consumption of modern interconnect components. On the contrary, current

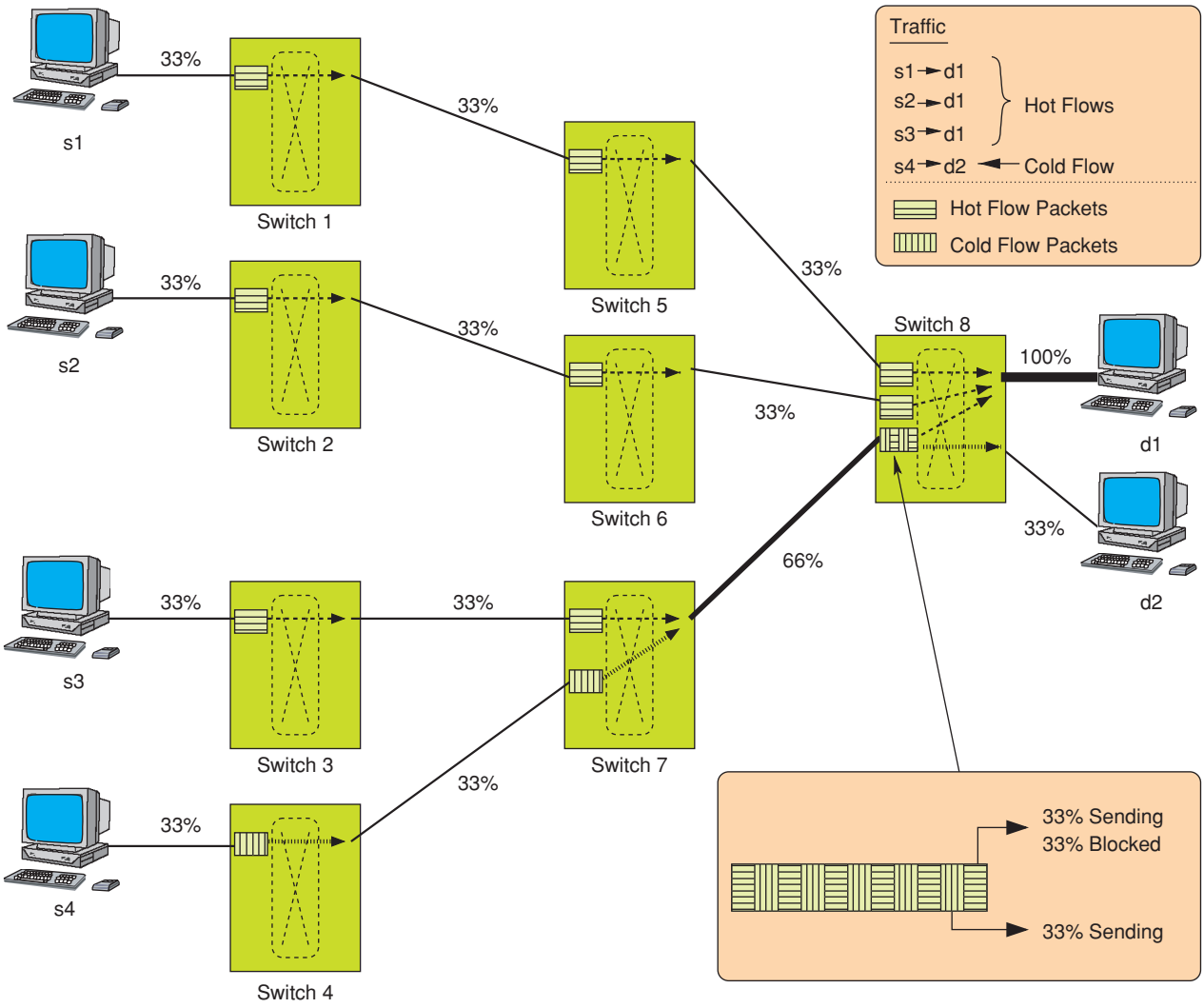

Fig. 3 Congestion tree causing HOL blocking to a cold flow

development trends toward reducing the number of network components for a given number of end nodes. This leads to a greater link utilization and thereby increases the probability of congestion. The solution therefore is not to overdimension but to utilize congestion management techniques.

In fact, in the last decades, researchers have proposed many techniques for dealing with congestion and HOL blocking (see "Read More About It"). From our point of view, these techniques can be classified into two basic groups, depending on the way the congestion problem is addressed:

- techniques that try to eliminate congestion itself (they try to prevent the existence of congestion trees in the network)
- techniques that don't try to eliminate congestion trees, but instead try to eliminate HOL blocking.

CONGESTION-FOCUSED TECHNIQUES

These techniques try to eliminate congestion trees in such a way that all the flows in the network are cold flows. They may be proactive, preventing the first appearance of congestion, or reactive, acting as soon as congestion is detected:

- **Proactive techniques** are based on controlling each data transmission, either by reserving network resources in advance or by limiting the routes followed by packets. This type of technique requires knowledge of the network status and application requirements that is not always available. Moreover, the larger the network, the greater the difficulty in knowing network status and the more conservative the behavior of the mechanism would become. In fact, proactive techniques are appropriate for providing quality of service (QoS) support, but not for congestion management.
- **Reactive techniques** detect the existence of congestion situations and activate mechanisms for eliminating the hot flows causing these situations. Most of these mechanisms consist of notifying the hot flow sources about congestion so that they throttle the injection of packets. The notifications must travel from the point where congestion is detected to the traffic sources, so there is a delay between congestion detection and reaction at the traffic sources—and this delay tends to increase with network size. And the greater the link bandwidth, the greater the number of packets that can be transmitted before the sources can react—packets that contribute to congestion. Therefore, reactive techniques do not scale, either with network size or with link bandwidth.

Both proactive and reactive techniques present scalability problems. In general, those techniques are not able to guarantee an effective elimination of congestion.

HOL-BLOCKING FOCUSED TECHNIQUES

Several techniques have been proposed for managing congestion by eliminating HOL blocking. The most effective ones are based on storing packets belonging to different flows in separate queues at each network port. The most important of these are 1) virtual output queues at network level (VOQnet), 2) virtual output queues at switch level (VOQsw), and 3) regional explicit congestion notification (RECN).

VOQnet requires, at each switch port, as many queues as end nodes in the network, and stores every incoming packet in the queue assigned to its destination. It is effective in HOL blocking elimination, since flows addressed to different destinations will always be stored in different queues. However, VOQnet requires a lot of resources (queues at each port) for networks with many end points. So it does not scale with network size and, in large networks, it is very expensive in terms of silicon area.

VOQsw maintains, at each switch port, as many queues as output ports in the switch, and it stores every incoming packet in the queue assigned to the output port requested by the packet. Therefore, the number of queues at each port depends on the number of switch ports, but not on the number of network end points. However, this technique can not guarantee complete elimination of HOL blocking, since it is still possible that cold and hot packets are stored in the same queue (if they request the same output port in a switch). Therefore, VOQsw eliminates HOL blocking just partially.

RECN is based on the idea that, if HOL blocking is completely eliminated, congestion trees may exist while being harmless. HOL blocking elimination implies that hot flows will not affect the forwarding of cold flow packets. Then the only packets delayed by congestion will be the ones contributing to congestion and so network throughput would be maximum. The RECN procedure detects the origin (root) of any congestion tree appearing in the network. This allows it to exactly identify hot flow packets as the packets passing through the root. Once it has detected a congestion tree, RECN dynamically assigns resources [set-aside queues (SAQs)] for storing only the hot flow packets from this tree, which are thus separated from any cold flow packets. HOL blocking by that tree is thereby avoided.

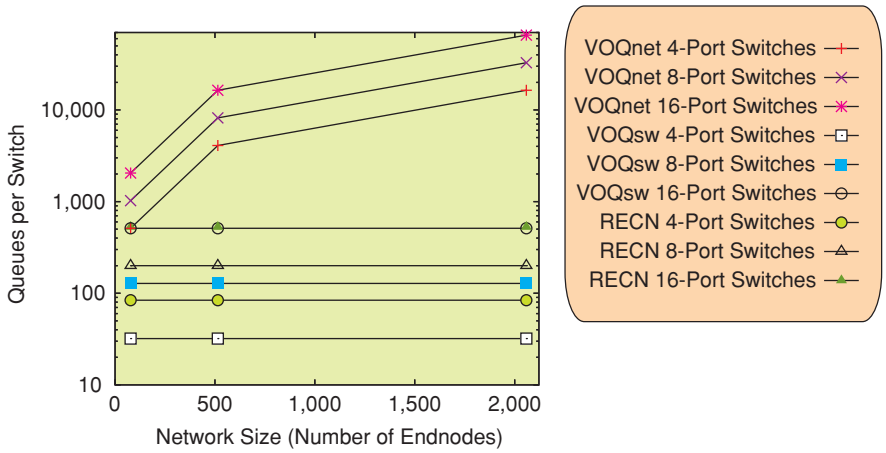

Fig. 4 Number of queues per switch when VOQnet, VOQsw, and RECN are used

RECN sends through the network the information about the position of the root of any congestion tree, so that SAQs are assigned for this tree at any point crossed by hot flows, thereby avoiding HOL blocking in the entire network. SAQs are dynamically deallocated when the associated congestion tree vanishes, so they can be later reallocated for any new congestion tree appearing in the network.

In addition to the SAQs, RECN requires a content addressable memory (CAM) at each input or output port for controlling the SAQs. It also needs detection queues at input ports for detecting the origin of congestion.

Unlike VOQnet and VOQsw, RECN assigns resources (SAQs) for eliminating HOL blocking only when a congestion tree has been detected and its root identified, and since these resources are deallocated when the congestion tree disappears, SAQs are used at any network point only when necessary for eliminating HOL blocking at this point. This policy reduces the number of resources required for completely eliminating the HOL blocking caused by congestion trees, thereby allowing RECN to deal with congestion in a very efficient way.

Figure 4 compares the number of queues per switch (on a logarithmic scale) required for implementing VOQnet, VOQsw, and RECN on networks of different size (64, 512, and 2048 end nodes) with switches with different numbers of ports (4, 8, and 16). For RECN, 8 SAQs per port are assumed. The number of queues per switch required by RECN is quite similar to that of VOQsw and, in both cases, the number of queues just grows with the number of ports. On the other hand, the VOQnet queue requirements grow with the number of ports and particularly with network size. Note that queue requirements for VOQnet differ from those of RECN and VOQsw by approximately two orders of magnitude for medium and large networks. In fact, the excessive number of queues required by VOQnet for medium or large networks makes the VOQnet scheme infeasible.

Figure 5 shows the network throughput obtained with VOQnet, VOQsw, and RECN for two networks of different size (64×64 and 512×512 BMINs) and for different traffic generation rates. Congestion was generated in each simulated point by injecting packets from several sources to hot spot destinations. In both cases, results for RECN and VOQnet are very similar (and far better than VOQsw results). This means that HOL blocking is almost completely eliminated by RECN and VOQnet regardless of the traffic rate. Indeed, both RECN and VOQnet allow maximum injection rates without throughput degradation. However, RECN, for both networks, used the same number of queues per switch port, while VOQnet's queues increased with network size (64 and 512 queues per switch port, respectively). Therefore, RECN has the advantage of eliminating HOL blocking with a constant (and affordable) number of resources.

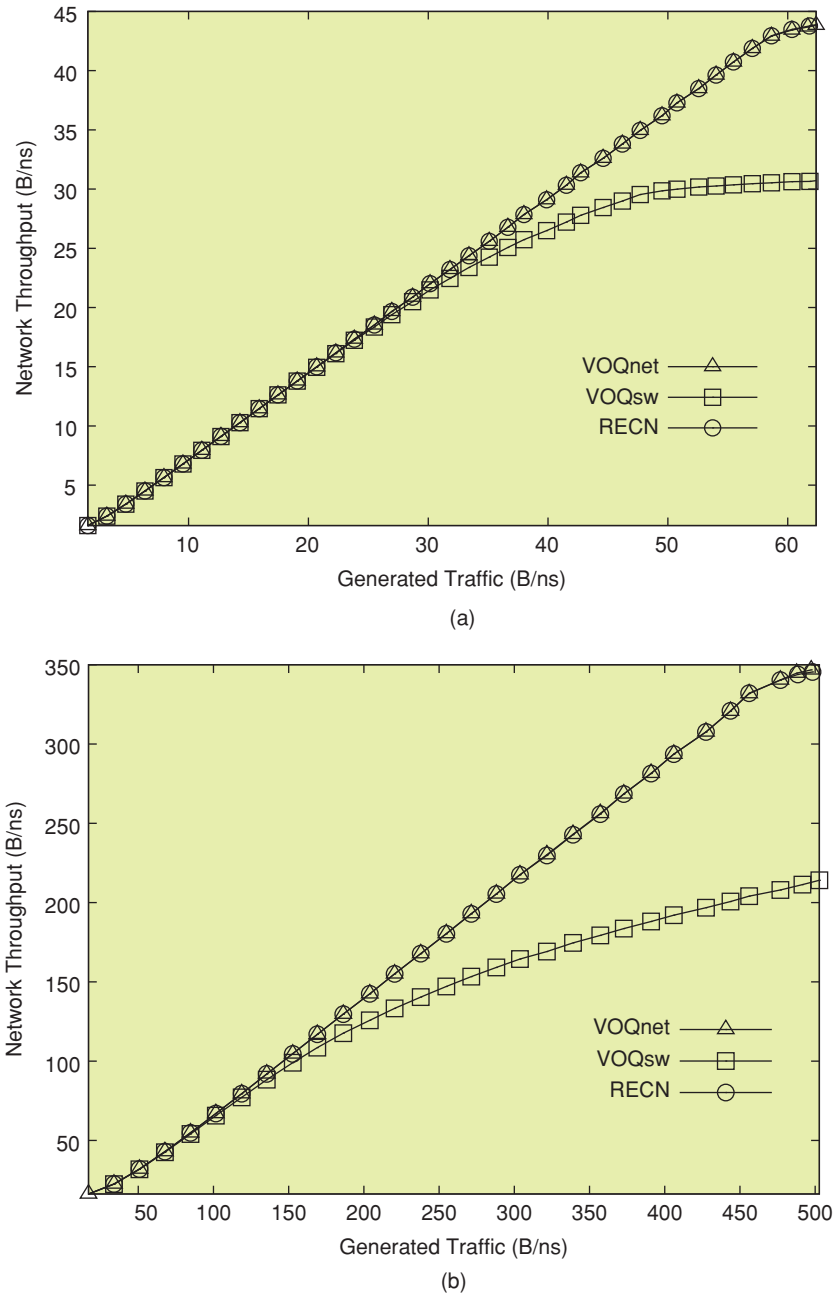

Fig. 5 Network throughput when VOQnet, VOQsw, and RECN are used on different BMINs (a) 64 × 64 and (b) 512 × 512

CONCLUSIONS

Congestion in interconnection networks can drastically degrade the bandwidth and speed of network-based computing and communication systems. The cause of this performance degradation is head-of-line blocking, in which congested packets (hot flows) impede the progress of noncongested packets (cold flows). Researchers have developed several techniques for dealing with the congestion problems. The most successful of these is regional explicit congestion notification (RECN), which operates by completely eliminating HOL blocking. In fact, RECN is the first technique that eliminates HOL blocking in a scalable and efficient way.

ACKNOWLEDGMENTS

This work was supported by Spanish MEC under CONSOLIDER-INGENIO CSD2006-46 and TIN2006-15516-C04 grants, and by Junta de Comunidades de Castilla-La Mancha under grant PBC-05-005.

READ MORE ABOUT IT

- G. Pfister and A. Norton, "Hot-spot contention and combining in multistage interconnect networks," *IEEE Trans. Comput.*, vol. C-34, pp. 943–948, Oct. 1985.
- P. Yew, N. Tzeng, and D.H. Lawrie, "Distributing hot-spot addressing in large-scale multiprocessors," *IEEE Trans. Comput.*, vol. 36, pp. 388–395, Apr. 1987.
- C.Q. Yang and A.V.S. Reddy, "A taxonomy for congestion control algorithms in packet switching networks," *IEEE Network*, pp. 34–45, July–Aug. 1995.
- S.P. Dandamudi, "Reducing hot-spot contention in shared-memory multiprocessor systems," *IEEE Concurrency*, vol. 7, pp. 48–59, Jan. 1999.
- P.J. Garcia, J. Flich, J. Duato, I. Johnson, F.J. Quiles, and F. Naven, "Dynamic evolution of congestion trees: Analysis and impact on switch architecture," *Lecture Notes in Computer Science (HiPEAC-2005)*, vol. 3793, pp. 266–285, Nov. 2005.
- P.J. Garcia, J. Flich, J. Duato, I. Johnson, F.J. Quiles, and F. Naven, "Efficient, scalable congestion management for interconnection networks," *IEEE Micro*, vol. 26, pp. 52–66, Sept.–Oct. 2006.

ABOUT THE AUTHORS

Pedro J. García (pgarcia@dsi.uclm.es) is an assistant professor of computer architecture and technology in the Computer Systems Department (DSI) at the University of Castilla–La Mancha, Spain.

José Flich (jflich@disca.upv.es) is an associate professor of computer architecture and technology in the Department of Computer Engineering (DISCA) at Technical University of Valencia, Spain.

José Duato (jduato@disca.upv.es) is a professor in the Department of Computer Engineering (DISCA) at Technical University of Valencia, Spain.

Ian Johnson (ian_johnson@xyratex.com) is chief scientist of Xyratex, Havant, U.K.

Francisco J. Quiles (paco@dsi.uclm.es) is a professor of computer architecture and technology in the Computer Systems Department (DSI) and vice-rector of research at the University of Castilla–La Mancha, Spain.

Finbar Naven (finbar_naven@virtensys) is a chief architect of switch architectures at VirtenSys, Cheadle, U.K.