Ethernet for High-Performance Data centers: On the New IEEE Datacenter Bridging Standards

Through the Datacenter Bridging task group, IEEE will add four supplements to the 802.1 standard that will both close the performance gap between Ethernet and InfiniBand and make the converged network a reality. In a converged network, all applications use a single physical infrastructure, which is ideal for the emerging next generation of data centers.

Sven-Arne Reinemo
Tor Skeie
Simula Research Laboratory

Manoj K. Wadekar
QLogic

Through several evolutionary steps, Ethernet has become an almost ubiquitous communication technology. Today it is applied in many different fields—from local area and enterprise networking through carrier-grade (telecommunication) networking to backplane interconnects. Ethernet has also made inroads in real-time systems and has proven versatile enough to be used in new domains. However, even if Ethernet seems ubiquitous, it still faces competition in some important domains, such as cloud computing, advanced data centers (ADC), and high-performance computing (HPC). To a varying degree, these domains all need a unified fabric with support for high bandwidth, low latency, and high utilization in a multiprotocol environment. In recent years, advanced installations have tended to move away from Ethernet toward newer, more advanced networking technologies better suited to fulfill the vision of a high-performance unified fabric. This has spurred the major Ethernet vendors and IEEE to create the Datacenter Bridging (DCB) task group to adapt Ethernet to high-performance networking.

The new network technologies that Ethernet must compete with are dedicated to HPC and ADCs and were designed with high performance in mind. They outstrip today's Ethernet in both speed and functionality. InfiniBand, the most prominent of these technologies, has established itself as the de facto standard for HPC.[1] It is also gaining popularity in enterprise solutions where performance is crucial—for example, the Sun Oracle Database Machine[2] and DB2 pureScale.[3] According to the November 2009 Top 500 Supercomputers list (http://top500.org), 65 percent of the top 100 most powerful supercomputers deploy InfiniBand, and only 1 percent use gigabit Ethernet. Moreover, it shows that 10 gigabit Ethernet has not made an impact, and only one 10-gigabit Ethernet system is on the top 500 list. Gigabit Ethernet still dominated the complete list (51.8 percent), but the number of Ethernet

Published by the IEEE Computer Society

systems has decreased by 9 percent over the last year while InfiniBand has increased by 28 percent. With Quad Data Rate InfiniBand (40 Gbits per second [Gbps]) products readily available and a better set of network features, Ethernet might risk losing market share in the HPC and ADC markets.

With respect to networking principles, Ethernet has several weaknesses compared to newer technologies. As the performance numbers in the top 500 list clearly show, the average utilization of gigabit Ethernet systems is 50 percent compared to 77 percent for InfiniBand systems (see http://top500.org). The most striking differences are that Ethernet does not offer

- virtual channels with independent flow control (not to be mistaken for the existing virtual local area network [VLAN] support) for performance, effective routing, and service differentiation purposes;
- lossless flow control in combination with service differentiation; or
- persistent congestion control.

These features are required for high performance in environments where protocols other than TCP/IP are common. Examples include interprocess communication (IPC) protocols such as message passing interface (MPI) and storage protocols such as iSCSI Extensions for RDMA (iSER), Network File System (NFS), and Fibre Channel. These features are also required to make the converged network a reality where several applications with different network demands can use a single physical network for all their communication. IEEE realized this several years ago and started the standardization work for DCB to close the performance and functionality gaps.

Datacenter bridging

The DCB umbrella includes three IEEE projects; an additional initiative is closely related to DCB. For convenience, we divide our presentation of these extensions into five topics:

- priority-based flow control (IEEE 802.1Qbb),
- enhanced transmission control (IEEE 802.1Qaz),
- congestion notification (IEEE 802.1 Qau),
- Datacenter Bridging Exchange (DCBX) protocol (part of IEEE 802.1Qaz), and
- shortest path bridging (IEEE 802.1aq)

When applied together, these features allow Ethernet to serve a unified network with multiple virtual application-specific networks on the same physical network. Note that the standardization efforts we discuss in this article are works in progress and the final standard might deviate from this presentation.

Priority-based flow control

Modern interconnection network technologies for HPC are lossless networks, where packet loss only occurs as a result of link transmission errors. Hence, the available link bandwidth is used effectively because retransmissions are seldom necessary. Low-latency operation of IPC protocols such as MPI and high throughput I/O protocols such as Fibre Channel and iSER require such lossless networks. The performance improvement is not free, however; it introduces the possibility of routing deadlock and head-of-line blocking, which can severely impact performance if not handled correctly.

To avoid packet loss due to buffer overflows when transient congestion occurs, most modern interconnection networks use point-to-point credit-based flow control. In credit-based flow control, the downstream side of a link tracks the available buffer resources (credits) by decreasing a credit counter whenever buffer space is allocated and increasing the credit counter whenever buffer space is de-allocated. Similarly, the upstream node tracks the available credits (that is, the number of bytes it may send) and decreases this amount whenever it sends a packet. Whenever credits arrive from the downstream node, it increases the amount of available credits. A packet is never sent downstream unless there is room for it.

Ethernet was originally a lossy fabric where congestion was handled by dropping packets and solving congestion was left to upper-layer protocols such as TCP or to the

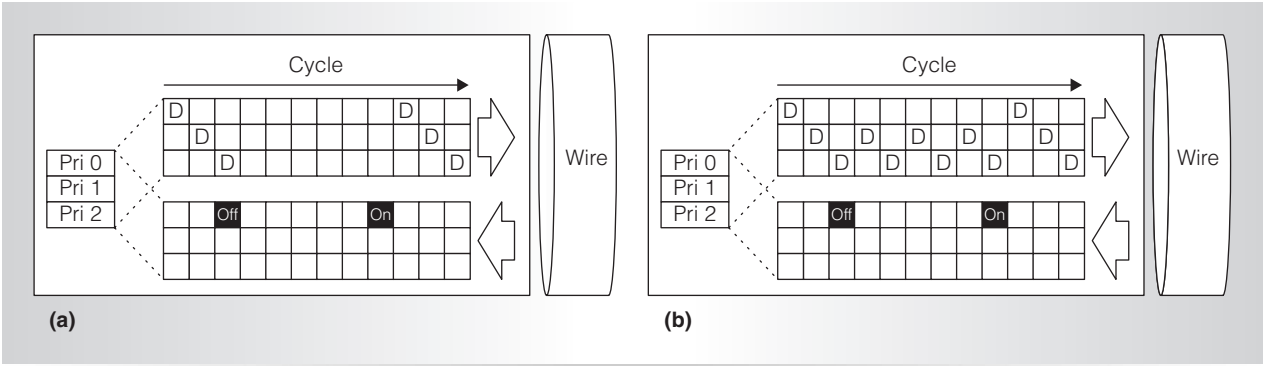

Figure 1. The effect of per-link (a) and per-priority (b) flow control on link activity. With per-link flow control, all transmission on all priorities is halted when transient congestion occurs. With per-priority flow control, only transmission on the priority where transient congestion occurs is halted.

application itself. By dropping packets, Ethernet avoids head-of-line blocking and the deadlock problem found in lossless networks. This approach has worked well in TCP/IP-centric scenarios where the TCP congestion-control mechanism detects packet loss and takes the appropriate measures. However, many applications in modern data centers bypass TCP/IP for performance reasons and cannot deal well with packet loss. MPI, iSER, and Fibre Channel over Ethernet applications can all experience a dramatic loss of performance due to retransmission (as a result of packet loss), which increases latency and reduces utilization of the link bandwidth. Therefore, support for lossless operation was added to Ethernet in the late 1990s but was seldom used. Today, however, there is growing demand for high-performance data centers where support for a lossless unified fabric is crucial. Since both approaches have benefits and drawbacks, the DCB standard aims to allow both lossy and lossless operation in the same fabric, all depending on the application's requirements and the network administrator's preference.

Ethernet currently defines link-level flow control in the IEEE 802.3x standard.[4] Rather than using credit-based flow control, Ethernet uses Xon/Xoff flow control, where upstream nodes are informed about the buffer situation through Xon/Xoff messages (*pause frames* in Ethernet terms). High and low buffer thresholds at the receiver side trigger the sending of Xon/Xoff messages. When the downstream node has available buffer space, it sends an Xon message to the upstream node telling it to start sending frames if any are available. As the transmission proceeds and the downstream node runs out of buffer space, it sends an Xoff message telling the upstream node to halt frame transmission (see Figure 1). For this scheme to work, these messages must be sent in a timely manner—that is, we need to account for the round trip time (RTT) between the upstream and downstream nodes. When the downstream node sends an Xoff message, it must do so at a point in time when it has enough space to buffer the frames received while it waits $1 \times$ RTT for the Xoff message to be processed and take effect. The delay between the Xoff message's transmission and its activation is due to the propagation and processing delay, the RTT that must be taken into account when calculating the buffer sizes, and threshold values. A drawback of Xon/Xoff flow control is that it requires twice the buffer space of credit-based flow control to sustain the link bandwidth. Xon/Xoff flow control must be able to buffer data equal to $2 \times$ RTT of the link because it takes $1 \times$ RTT to activate the Xon message and $1 \times$ RTT to activate the Xoff message.

Credit-based flow control, however, only needs to buffer data equal to $1 \times$ RTT because the delay between credit updates equals $1 \times$ RTT. With Xon/Xoff flow control, the signaling overhead is less than for credit-based flow control as there is no regular transmission of flow-control credits; however, the reduced signaling can lead to a less robust mechanism because a lost Xon/Xoff frame will lead to frame loss.

The main problem to be addressed, however, is that the current Ethernet Xon/Xoff flow-control implementation is port-based. Thus, when a downstream node tells an upstream node to halt frame transmission, it affects all traffic on the link (see Figure 1a). This action has consequences for service differentiation and is why the IEEE 802.1Qbb project is defining a new flow-control mechanism that can halt frames with a given priority while allowing frames with other priorities to proceed (Figure 1b).[5] The mechanism adds a pause frame containing the information necessary to exercise flow control per priority. Combined with enhanced transmission selection (described in the next section), the mechanism can pause low-priority traffic while high-priority traffic proceeds and vice versa. The new pause frame, called *priority flow control pause*, extends the standard MAC control frame with two new parameters: the priority enable vector (PEV) and the time array. The PEV is a bit vector in which the values in the least significant octet indicate if flow control is active for any of the eight available priorities. The time array has a corresponding time value for each of the eight priorities in the PEV that specifies how long the corresponding priority must wait (the pause time) before sending its next frame to avoid buffer overflow at the receiver. The pause time will eventually expire to avoid permanently pausing a link if a pause frame is lost. This requires, however, refreshing the pause by sending another pause frame if the situation persists beyond the expiration.

Both the PEV and the time array will be configurable properties in 802.1Qbb-compliant switches and network interface cards (NICs). The network administrator will configure them using the DCBX protocol.

Enhanced transmission selection

Service differentiation is the unequal treatment of traffic based on a set of predefined properties. The granularity of differentiation ranges from no differentiation (that is, best effort) at one extreme to flow-level differentiation at the other extreme. In the middle is class-level differentiation, which many modern interconnection network technologies support.

Table 1. Allocation of priorities and bandwidth to virtual channels. Each virtual channel is directly linked to a priority group.

Channel	Application	Priorities	Bandwidth share (%)
Virtual channel 1	LAN traffic	0, 3, 4, 5	40
Virtual channel 2	SAN traffic	1, 6	40
Virtual channel 3	IPC traffic	2, 7	20

The InfiniBand architecture supports class-level differentiation by combining a priority mechanism and a virtual channel scheme. The independent resources (such as buffers) dedicated to each virtual channel in combination with the priority mechanism form the core of InfiniBand's quality-of-service capabilities.[1,6]

Ethernet has its own implementation of service differentiation. In the current version, it supports priority tagging of packets and up to eight queues within the switches to discriminate packets based on eight priority levels. These features will be enhanced to provide virtual channels in which each virtual channel has its own buffers and flow-control resources that can help improve network performance through traffic isolation, efficient deadlock-free routing, fault-tolerance, and service differentiation.[7,8]

Virtual channels can be visualized as a physical link providing multiple virtual interfaces for different traffic classes (for example, LAN, system area network [SAN], and IPC traffic). Note that virtual channels differ from VLANs, which are already supported by Ethernet. VLANs are separated by tags in the frame header, but there are no dedicated resources (buffers and flow control) for a given VLAN.[9] Each traffic class is mapped into a virtual interface and consists of multiple priorities, as Table 1 shows. The virtual channels do not need a separate identifier on the link since they are constructed by associating a set of 802.1p priority values to a single virtual channel.

The concept of virtual channels requires three changes to the existing priority scheme, which the IEEE 802.1.Qaz Enhanced Transmission Selection (ETS) project is addressing.[10]

First, the current IEEE 802.1p standard classifies traffic flows using 3-bit tagging.

Switches use this classification to queue different traffic types in up to eight queues. The standard specifies strict priority scheduling for these queues, allowing high-priority traffic to be serviced before low-priority traffic, and thus achieving lower latency and lower loss probability for high-priority traffic. But this creates unfairness to other queues, and due to the strict priority scheduling, high-priority queues can cause starvation of low-priority queues. Replacing the strict priority scheduling with a weighted round robin[11] or deficit weighted round robin[12] algorithm for scheduling between priorities can avoid the starvation problem.

Second, an obvious conflict exists between the Ethernet priority-tagging concept and the use of a port-based flow-control mechanism as described previously. The consequence is that it cannot offer virtual channels to enhance performance and isolate traffic. So, a priority-based flow-control supplement is a prerequisite for ETS.

Third, when multiple traffic classes are consolidated into a set of virtual channels, there is no inherent priority of traffic between these virtual channels. But each virtual channel would like to maintain its current usage model—that is, a single interface with support for multiple priorities. It would also like to maintain bandwidth allocations for the given virtual channel independent of the traffic on other virtual channels. Therefore, each virtual interface must be made responsible for the amount of bandwidth used by the priorities belonging to this interface. Furthermore, the scheduling of packets from individual priorities must be defined by each packet's relative priority within the virtual channel where it belongs (Table 1). ETS achieves this by defining a set of priority groups, where each priority group contains one or more of the eight available priorities.

The assignment of priorities to priority groups is defined in a *priority to priority group table*, and the priority group's share of the total link bandwidth is defined in the *priority group table*. The priority group distributes its assigned bandwidth among the priorities belonging to the group. The priorities within a group must all be of the same type—that is, they must all be either lossy or lossless; mixing lossy and lossless priorities within the same priority group is not allowed. The relationship between priorities, priority groups, and virtual channels is maintained in 802.1Qaz-compliant switches and NICs, and it is configured by the network administrator using the DCBX protocol.

Congestion notification

To avoid frame loss due to network congestion and to reduce the effect of congestion trees, Ethernet should support both transient and persistent congestion control. Transient congestion is handled by point-to-point flow control, which avoids packet loss across a single link without reducing link utilization. However, when congestion persists (for example, due to a hot spot), it spreads upstream through the network and results in the growth of congestion trees, which eventually terminate at the end nodes. Obviously, this is a bad situation for a network as the congestion trees' growth can quickly preclude transmission of other flows (victims of congestion) that are not even destined for the congested area. The phenomenon causing the spread of congestion to parts of the network that do not contribute to the congestion is called *head-of-line blocking*. Head-of-line blocking occurs, for instance, when the head of a first-in/first-out queue is stalled due to heavy traffic. The head of the queue is then rightly stalled because it is headed for a congested destination, but the packets behind it that are not headed for a congested destination are unjustly stalled. This problem becomes even worse when we use lossless flow control, as the effect of back pressure creates congestion trees upstream from the congestion point. In other words, flow control solves the problem of packet loss in the case of transient congestion, but introduces the problem of congestion trees when the congestion persists. Therefore, a persistent congestion-control mechanism is required, especially in multihop topologies.

Users of technologies such as InfiniBand have experienced persistent congestion in combination with lossless flow control. As a result, a revision of the InfiniBand specification includes congestion control.[1,13] IEEE has recently standardized a similar set of

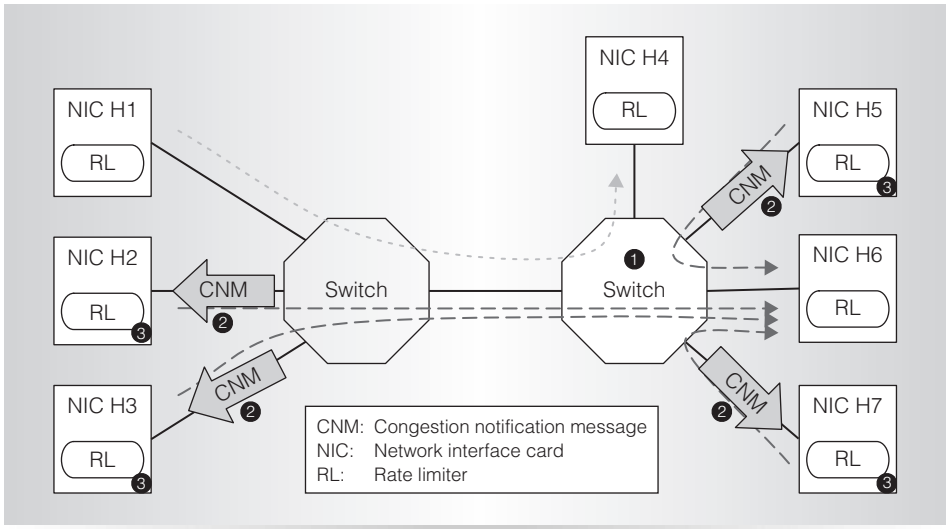

Figure 2. Quantized congestion notification protocol with the congestion point (1), congestion notification message (2), and rate limiter (3).

features for quantized congestion notification (QCN).[14]

The congestion-control mechanism provides information about the congestion in the subnet so that the rate limiter can take appropriate action at the ingress nodes that are contributing to congestion. It acts proactively to avoid oversubscription of subnet resources and reduce the growth of congestion trees, thereby improving overall network performance. In general, a congestion-control mechanism provides the following functions:

- *Congestion detection.* Network switches monitor egress queues to detect congestion.
- *Congestion notification.* The mechanism provides a protocol for notifying the ingress devices about congestion, including the details required to make a response. The notification can be sent in the forward direction as with TCP Explicit Congestion Notification and InfiniBand congestion control, or it can be sent in the backward direction toward the ingress as with the Internet Control Message Protocol's (ICMP) source quench and Ethernet QCN. In the latter case, the congestion notifications are generated for a specific ingress device and are sent as unicast frames addressed to the ingress device that is the congestion source.
- *Congestion response.* The mechanism defines the behavior at the ingress node when it receives a congestion notification. Based on the information in the notification, the ingress node might change the injection rate toward the congested destination using a rate limiter. The rate limiter controls the injection rate only for the congested destination; traffic to other destinations is unaffected.

The IEEE 802.1Qau standard specifies these three ingredients. It defines the congestion notification message (CNM) as flowing in a backward direction from the congestion point towards the ingress source. Ingress sources adapt the injection rate for the congested destinations as specified by the QCN algorithm.

Figure 2 shows the flow of CNMs from the congestion point toward the end nodes contributing to congestion. We call these *reaction points*. When the congestion point detects an egress queue level higher than the predefined threshold, it generates a CNM and sends it to the reaction points. The CNM contains the quantized feedback, F_b, which is based on probabilistic sampling at the congestion point to avoid generating too many CNMs.

The value of F_b is the quantization of the queue size excess and the rate excess at the

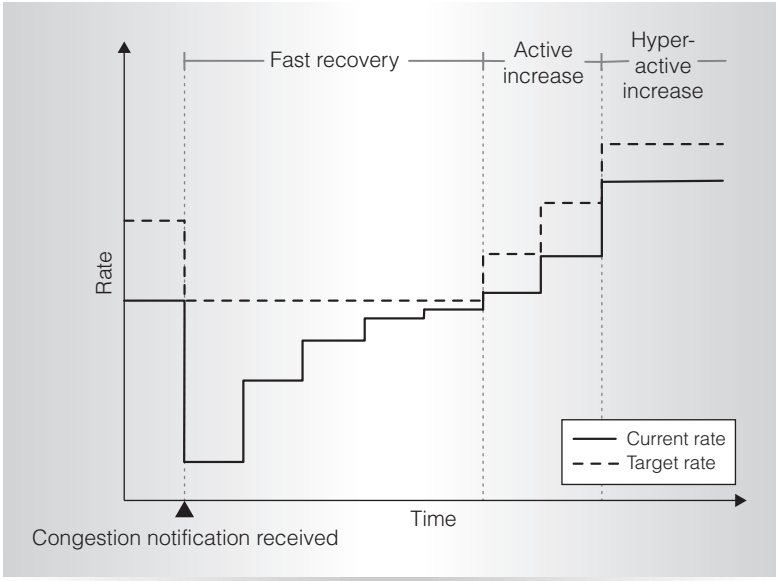

Figure 3. The process of rate decrease and rate increase at the reaction point. After a rate decrease, a new target rate is set and the rate limiter will slowly try to increase the sending rate to match the target rate. The rate increase happens in the three stages (fast recovery, active increase, and hyperactive increase), where each stage increases the rate faster than the previous one.

current congestion point. When F_b is negative, the queue is oversubscribed and there is a certain probability that the congestion point sends a CNM to the reaction points identified by the source address in the frame that triggered the congestion. When F_b is positive, there is no congestion and the congestion point does not generate a CNM.

When a reaction point receives a CNM, it installs a rate limiter to control the rate of traffic toward the congested destination. QCN defines an additive increase, multiplicative decrease algorithm loosely similar to Binary Increase Congestion control TCP,[15] where the rate limiter performs a multiplicative decrease upon receiving a CNM from a congestion point and then increases the rate autonomously in absence of feedback. Thus, the sources contributing to congestion are forced to reduce the amount of traffic injected toward the congestion point, reducing the negative impact on victim flows.

The rate limiter continuously reduces the transmission rate at the reaction points as long as it receives CNMs. In the absence of CNMs, it continuously tries to recover its transmission rate. Figure 3 shows the various stages of rate recovery in the QCN reaction point algorithm. In the figure, the target rate is the transmission rate for the reaction points just before the reception of the last CNM. After a reduction, the target rate is set as the new goal for the current rate that the reaction point is currently transmitting at. The rate increase happens in three stages: fast recovery, active increase, and hyperactive increase, where each stage increases the rate faster than the previous one. Using multiple stages helps the reaction point algorithm quickly search for and stabilize at the correct rate for a given reaction points and appropriately share the bandwidth available at the congestion point.

Ethernet has two apparent improvements over the InfiniBand congestion-control mechanism.

- It uses backward congestion notification rather than forward congestion notification, reducing the time between the detection of congestion and notification of congestion at the contributing source.
- The notifications contain information about the congestion's severity, allowing the contributor to adapt its rate reduction accordingly.

Datacenter Bridging Exchange protocol

The DCBX protocol is specified as part of the 802.1Qaz Enhanced Transmission Selection standard and is used for automatic exchange and configuration of DCB equipment. Its main responsibility is to configure link parameters related to DCB. It has three main features:

- a protocol for exchanging DCB parameters between peers;
- a write operation to set received DCB parameters; and
- a resolution engine for conflicting settings.

The DCBX protocol covers functionality for peer discovery, information exchange between peers, detection of invalid configurations, auto negotiation, and peer configuration. The DCBX protocol is based on the

Link-Layer Discovery Protocol and supports three main capabilities:

- *Distribution of information*. It supports the passing of information used for DCB applications or for peer configuration. For example, a switch can pass the number of traffic classes it supports, and the end point can use this information to organize its buffers.
- *Symmetric parameter passing*. With symmetric parameter passing, the goal is that both sides of link use the same parameters. For example, for priority-based flow control, both the send and receive sides of the link are configured with the same number of priorities.
- *Asymmetric parameter passing*. Asymmetrical parameter passing is used when certain configuration parameters differ at the send and receive sides of the link. For example, for ETS bandwidth configuration, the upstream and downstream bandwidth might differ.

The write operation sets the received parameters at the receiver side. DCBX combines this operation with a conflict-resolution mechanism that discovers mismatched or invalid configurations based on the symmetric or asymmetric parameter passing. The protocol currently covers the configuration of the parameters specified in 802.1Qbb, 802.1Qaz, and 802.1Qau.

Shortest path bridging

The routing algorithm describes how we compute the path a packet travels from source to destination, and has a major impact on both latency and throughput. Designing a routing algorithm that minimizes latency, maximizes throughput, and is deadlock free when used with lossless flow control, is difficult—not only because these goals often conflict, but also because the target technology's available features constrain the design. A good general-purpose routing algorithm should be topology agnostic, deadlock free, and use all links available in the underlying topology. This is not the case for the routing algorithm used in Ethernet.

The default routing algorithm for Ethernet switches is the Multiple Spanning Tree Protocol defined in IEEE Standard 802.1D.[16] MSTP guarantees connectivity while preventing loops in the network by using an automatic configuration phase in which any topology is turned into a tree within a given VLAN. The problem with reducing any topology into a tree is that we end up with a lot of unused links and wasted resources, unless we use multiple VLANs with a separate tree for each. For LANs this is not a severe problem, but in high-performance data centers we need efficient topologies and want to use every link available to achieve the best performance possible. In this setting, the MSTP is no longer an efficient solution and a new and improved solution should be sought if Ethernet is to remain competitive.

The IEEE 802.1aq Shortest Path Bridging (SPB) project is working on a proposal that solves this problem.[17] Rather than use a single spanning tree for each switch, it lets each switch become the root of its own minimal spanning tree. Each switch will then have a shortest path to all other destinations, and by careful merging of symmetric spanning trees, all-to-all shortest-path routing is possible. In SPB, a modified version of the Intermediate System-Intermediate System (IS-IS) link state routing protocol[18,19] performs topology discovery and each switch uses the result of the discovery process to calculate the shortest path trees. SPB has several benefits over MSTP and similar solutions. It makes shortest-path routing a reality with only minor changes to the forwarding hardware in current Ethernet switches, and it supports multiple equal cost shortest-path trees when the physical topology allows it. Furthermore, it limits the negative impact of link and switch faults because it requires only local reconfiguration when using IS-IS. In addition, it is interoperable with legacy equipment not supporting SPB.

A similar approach to deliver shortest-path routing in Ethernet is the IETF Transparent Interconnection of Lots of Links (TRILL) working group.[20] The TRILL working group is designing a topology-agnostic shortest-path routing algorithm that is Ethernet compliant and based on the same link-state routing protocol technology as SPB. One important difference

between TRILL and SPB is that TRILL is switch adaptive while SPB is host adaptive when multiple shortest paths are used. In other words, with SPB the source can choose between multiple deterministic paths, while with TRILL the path is selected on a hop-by-hop basis as a packet traverses the network. The latter approach is less predictable and harder to control from a traffic-management perspective because the path taken between a given <source, destination> might differ on a packet-by-packet basis.

None of these proposals, however, handles the deadlock problem when used with lossless flow control. Therefore, traffic that needs lossless flow control must be separated out on a special priority, and this priority must be carefully managed by VLANs or the physical topology to avoid deadlocks. This might limit the use of lossless flow control due to increased complexity in network configuration and reduced choice of topologies. This contrasts with InfiniBand, where several deadlock-free routing algorithms address various topologies. Several academic proposals solve the deadlock problem for Ethernet,[7,21,22] but adopting these proposals requires a larger degree of change than the approach described here.

The emerging Ethernet DCB enhancements aim to close the gap to other high-performance networking technologies and make Ethernet the de facto standard in datacenter computing. Based on the suggested improvements in flow control, service differentiation, congestion control, and routing, the technical issues seem to be addressed in a good manner (perhaps with the exception of lossless flow control and deadlock-free routing, which do not seem fully resolved). The extent to which DCB can raise Ethernet's performance remains to be seen and depends on the DCB standard's actual implementation in upcoming Ethernet devices. MICRO

References

1. InfiniBand Trade Assoc., *InfiniBand Architecture Specification,* 1.2.1 ed., 2007.
2. R. Weiss, ''Sun Oracle Database Machine and Exadata Storage Server,'' whitepaper, Oracle, 2010.
3. IBM Corp., ''Transparent Application Scaling with IBM DB2 PureScale,'' whitepaper, IBM, 2009.
4. *IEEE Standards 802.3-2002 LAN/MAN CSMA/CD Access Method,* IEEE, 2002.
5. H. Barrass et al., ''Definition for New Pause Function,'' IEEE 802.1 Working Group presentation, http://ieee802.org/1/files/public/docs2007/new-cm-barrass-pause-proposal.pdf.
6. S.-A. Reinemo et al., ''An Overview of QoS Capabilities in InfiniBand, Advanced Switching Interconnect, and Ethernet,'' *IEEE Comm.,* vol. 44, no. 7, 2006, pp. 32-38.
7. S.-A. Reinemo and T. Skeie, ''Effective Shortest Path Routing for Gigabit Ethernet,'' *Proc. 2007 IEEE Int'l Conf. Comm.* (ICC 07), IEEE CS Press, 2007, pp. 6419-6424.
8. S.-A. Reinemo, T. Skeie, and O. Lysne, ''Applying the Diffserv Model in Cut-through Networks,'' *Proc. 2003 Int'l Conf. Parallel and Distributed Processing Techniques and Applications,* Computer Science Research, Education, and Applications (CSREA) Press, 2003, pp. 1089-1095.
9. *IEEE Standard 802.1Q-2003 Virtual Bridge Local Area Networks,* IEEE, 2003.
10. M.K. Wadekar et al., Priority Groups (Traffic Differentiation over Converged Link), IEEE 802.1 Working Group presentation; http://ieee802.org/1/files/public/docs2007/new-wadekar-priority-groups-1107-v1.pdf.
11. W.J. Dally and B. Towles, *Principles and Practices of Interconnection Networks,* Morgan Kaufmann, 2004.
12. M. Shreedhar and G. Varghese, ''Efficient Fair Queueing Using Deficit Round Robin, *ACM SIGCOMM Computer Comm. Rev.,* vol. 25, no. 4, Oct. 1995, pp. 231-242.
13. E.G. Gran et al., ''First Experiences with Congestion Control in InfiniBand Hardware,'' *Proc. 24th IEEE Int'l Parallel & Distributed Processing Symp.* (IPDPS 10), IEEE CS Press, 2010, pp. 1-12.
14. *IEEE Standard 802.1Qau-2010 IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 13: Congestion Notification,* IEEE, 2010.
15. L. Xu, K. Harfoush, and I. Rhee, ''Binary Increase Congestion Control for Fast, Long Distance Networks,'' *Proc. 23rd Ann. Joint Conf. IEEE Computer and Comm. Societies*

(INFOCOM 2004), vol. 4, IEEE Press, 2004, pp. 2514-2524.
16. *ANSI/IEEE Std 802.1D Media Access Control (MAC) Bridges,* IEEE, 1998.
17. *IEEE Standard 802.1aq—Shortest Path Bridging Draft 2.1,* 2009, http://www.ieee802.org/1/pages/802.1aq.html.
18. D. Oran, *OSI IS-IS Intra-domain Routing Protocol,* IETF RFC 1142, Feb. 1990; http://www.rfc-editor.org/rfc/rfc1142.txt.
19. P. Ashwood-Smith et al., ''Shortest Path Bridging and Backbone Bridging with IS-IS,'' IETF Internet draft, work in progress, June 2009.
20. J. Touch and R. Perlman, *Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement,* IETF RFC 5556, May 2009; http://www.rfc-editor.org/rfc/rfc5556.txt.
21. F. De Pellegrini et al., ''Scalable Cycle-Breaking Algorithms for Gigabit Ethernet Backbones,'' *Proc. 23rd Ann. Joint Conf. IEEE Computer and Comm. Societies* (INFOCOM 04), vol. 4, IEEE Press, 2004, pp. 2175-2184.
22. O. Lysne et al., ''Layered Routing in Irregular Networks,'' *IEEE Trans. Parallel and Distributed Systems,* vol. 17, no. 1, 2006, pp. 51-65.

Sven-Arne Reinemo is a postdoctoral researcher at Simula Research Laboratory. His current research interests are routing, fault tolerance, and quality of service in interconnection networks. Reinemo has a PhD in computer science from the University of Oslo. He is a member of IEEE.

Tor Skeie is a professor at the Simula Research Laboratory and the University of Oslo. His research interests include scalability, effective routing, fault tolerance, and quality of service in switched network topologies. Other research interests include the road to deterministic Ethernet end-to-end and how precise time synchronization can be achieved across switched Ethernet. Skeie has a PhD in computer science from the University of Oslo.

Manoj K. Wadekar is a Fellow and senior director of technology at QLogic Corporation. His research interests include standardizing enhancements to Ethernet for the data center in the IEEE and in cross-industry initiatives. Wadekar has a master's degree in electrical engineering from the Indian Institute of Technology. He is a member of IEEE and a voting member of the IEEE 802.1 working group.

Direct questions and comments to Sven-Arne Reinemo, Simula Research Laboratory, Martin Linges vei 17, Fornebu, N-1325 Lysaker, Norway; svenar@simula.no.