The Internet Protocol Journal

September 2011 Volume 14, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at:
www.cisco.com/ipj

ISSN 1944-1134

From The Editor

I recently attended a conference in Japan where the attendee network offered IPv6 service only. In the past, conferences such as the *Asia Pacific Regional Conference on Operational Technologies* (APRICOT) and meetings of the *Internet Engineering Task Force* (IETF) have conducted IPv6 experiments, but these have all been "opt-in" events. The conference in Japan was different: there was no IPv4 service available. Making this work involved a few manual configuration steps, but for the most part everything worked more or less the same as it did under IPv4. Some applications, including my instant message client and Skype did not work, and all connections to IPv4-only hosts needed to use *Fully Qualified Domain Names* (FQDNs) instead of IP addresses, but overall the experience gave me confidence that IPv6 is becoming a reality. As you might expect, this IPv6-only experiment also uncovered a number of bugs and incompatibilities that were duly reported to developers around the world.

Our first article is an overview of *TRansparent Interconnection of Lots of Links* (TRILL). TRILL uses Layer 3 routing techniques to create a large cloud of links that appear to IP nodes to be a single IP subnet. The protocol has been developed in the IETF and is currently being refined and enhanced in the TRILL working group. The article is by Radia Perlman and Donald Eastlake.

Developments in Internet technologies have lead to changes that go beyond the Internet itself. Not only is *Voice over IP* (VoIP) often used in place of traditional circuit-switched telephony, the telecommunication networks themselves are evolving to incorporate IP routers in place of traditional telephone switches. This evolution also applies to cellular telephone networks, specifically to what is known as *backhaul*—the transportation of voice and data from the cell sites to the mobile operators' core networks. Jeff Loughridge explains more in "The Case for IP Backhaul."

Once again I would like to remind you about the IPJ subscription renewal campaign. Each subscriber to this journal is issued a unique subscription ID that, coupled with an e-mail address, gives access to the subscription database by means of a "magic URL." If your subscription has expired or you have lost your subscription ID, changed e-mail, postal mail, or delivery preference, just send an e-mail to **ipj@cisco.com** with the updated information and we will take care of the rest.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Introduction to TRILL

by Radia Perlman, Intel Labs, and Donald Eastlake, Huawei Technologies

Transparent Interconnection of Lots of Links (TRILL)[1] is an *Internet Engineering Task Force* (IETF) protocol standard that uses Layer 3 routing techniques to create a large cloud of links that appear to IP nodes to be a single IP subnet. It allows a fairly large Layer 2 cloud to be created, with a flat address space, so that nodes can move within the cloud without changing their IP addresses, while using all the Layer 3 routing techniques that have evolved over the years, including shortest paths and multipathing. An early problem and applicability statement for TRILL can be found in [6]. Additionally, TRILL supports Layer 2 features such as *Virtual Local-Area Networks* (VLANs), the ability to autoconfigure (while allowing manual configuration if so desired), and multicast/broadcast with no additional protocol.

Additionally, TRILL is evolutionary in the sense that an existing Ethernet deployment, where the links are connected with bridges, can be converted into a TRILL cloud by replacing any subset of the bridges with devices implementing TRILL. Devices implementing TRILL are called *Routing Bridges,* or *RBridges*. As bridges are replaced, nothing changes for the IP nodes connected to the cloud except that the cloud becomes more stable and uses available bandwidth more effectively.

To understand why TRILL was needed, it is helpful to explore the history of Ethernet and IP.

Network protocols are usually described in terms of *layers*. The description usually quoted in textbooks is the *Open Systems Interconnection* (OSI) *Reference Model,* which describes seven protocol layers[4]. It is important to realize that the layers are useful primarily as a way to think about networking, but actual network protocols are far more complex. Layers get subdivided or combined, and often a technology usually thought of as belonging to a lower layer (for example, Layer 2) can be layered on top of a higher layer (for example, Layer 3). Most descriptions of network layers agree on the bottom four layers, and vary according to details such as whether syntax (for example, *Extensible Markup Language* [XML][7]), which would be a *Presentation Layer* in the OSI model, is a layer or not. Such descriptive choices do not affect how protocols are built, and luckily, for understanding of TRILL, the relevant layers to focus on are just the bottom three:

- Layer 1, *Physical Layer:* Physical, electrical, and optical specification for connectors, bit signaling, etc.
- Layer 2, *Data Link Layer:* The protocol that lets neighbor nodes on a link exchange packets
- Layer 3, *Network Layer:* The protocol that provides routing to create a path from a source node to a destination node

TRILL, as we will see, is a Layer 2 and ½ protocol: It glues links together so that IP nodes see the cloud as a single link. Therefore, TRILL is below Layer 3; but, it is above Layer 2 because it terminates traditional Ethernet clouds, just like IP routers would do.

It is definitely time to be confused. Why are there multiple links at Layer 2? Isn't that the job of Layer 3?

Evolution of Layer 2 from Point-to-Point Links to LANs

In the beginning (the 1970s or so for the purposes of this article), Layer 2 really was a direct link between neighbor nodes. Most links were point-to-point, and Layer 2 protocols primarily created *framing*—a way to signal the beginning and end of packets within the bit stream provided by Layer 1—and *checksums* on packets[11]. For links with high error rates, Layer 2 protocols such as *High-Level Data Link Control* (HDLC)[12] provided message numbering, acknowledgements, and retransmissions, so the Layer 2 protocol resembled, in some ways, a reliable protocol such as TCP. HDLC and other Layer 2 technologies sometimes provided an ability to have multiple nodes share a link in a master/slave manner, with one node controlling which node transmits through techniques such as polling.

Then the concept of *Local-Area Networks* (LANs) evolved, the most notable example being Ethernet. Ethernet technology enabled interconnection of (typically) hundreds of nodes on a single link in a peer-to-peer rather than master/slave relationship. Ethernet was based on *CSMA/CD,* where CS = *Carrier Sense* (listen before talking so you don't interrupt); MA = *Multiple Access;* and CD = *Collision Detect* (listen while you are talking to see if someone starts talking while you are so you are both interfering with each other). Interestingly, although IP had a 4-byte address and was the basis of addressing for the entire Internet, Ethernet had a larger 6-byte address, with aspirations for connecting only a small number of nodes in a fairly small region such as a single building.

The reason for the larger address space for Ethernet was to avoid the need to configure addresses when plugging nodes into a network. Instead, manufacturers of equipment would purchase blocks of Ethernet addresses and embed a unique address for each device in their hardware (the "MAC address"), and an Ethernet node would then be able to use that address in any Ethernet without fear of address collision.

Evolution of Ethernet to Spanning Tree

LANs came onto the scene with such fanfare that people came to believe that LAN technology was a replacement of traditional Layer 3 protocols such as IP. People built applications that were implemented directly on Layer 2 and had no Layer 3. This situation meant that the application would be limited by the artifacts of the Layer 2 technology, because a Layer 3 router cannot forward packets that do not contain the Layer 3 header implemented by the router.

In the case of the original Ethernet, it meant the application would work only within a maximum distance of perhaps a kilometer.

When people using technologies built directly on a LAN realized they wanted networks larger (in distance and total number of nodes) than the LAN technology allowed, the industry invented the concept of "bridges"—packet-forwarding devices that forwarded Layer 2 packets.

Forwarding Ethernet packets might seem easy because the Ethernet header looks similar to a Layer 3 header. It has a source and destination address, and the addresses are actually larger than IP addresses. But Ethernet was not designed to be forwarded. Most notably absent from the Ethernet header is a *hop count* (also sometimes referred to as a "time to live," or TTL) to detect and discard looping packets. But other features of a typical Layer 3 protocol were also missing in Ethernet, such as an address that reflects where a node is in the topology, node discovery protocols, and routing algorithms. These features were not in Ethernet because the intention of the Ethernet design was that it be a Layer 2 protocol, confined to operation on a single link.

The transparent bridge was invented as a mechanism to forward Ethernet packets emitted by end nodes that did not implement Layer 3. Ethernet at the time had a hard packet size limit, so bridges could not modify the packet in any way.

The transparent bridge design, which met those constraints, consisted of having bridges listen promiscuously, remember the source addresses seen on each port, and forward based on the learned location of the destination address. If the destination was unknown, the packet would be forwarded onto all ports except the one that it was received on.

This simple method worked only if there was only one path between any pair of nodes. So the concept was enhanced with a protocol known as the *Spanning Tree Algorithm*.[8] The physical topology could be an arbitrary mesh, but bridges, using the spanning-tree algorithm, would prune the topology into a loop-free (tree) topology on which data packets were forwarded. ("Spanning" means that packets can reach all the nodes.)

As Figure 1 shows, the spanning-tree concept is that an arbitrary topology could be built using Ethernet links (horizontal lines) and bridges (circles). Bridges running the spanning-tree algorithm determine a loop-free subset of the topology, and put some ports into standby (the ones that are shown in Figure 2 as dotted lines). Data packets flow on the ports that spanning tree determines should be active. This model does not yield optimal routes, as indicated in Figure 3, where packets between A and X go through the path of bridges 11, 7, 6, 2, 14, 4, and 3.

Figure 1: A Bridged Network

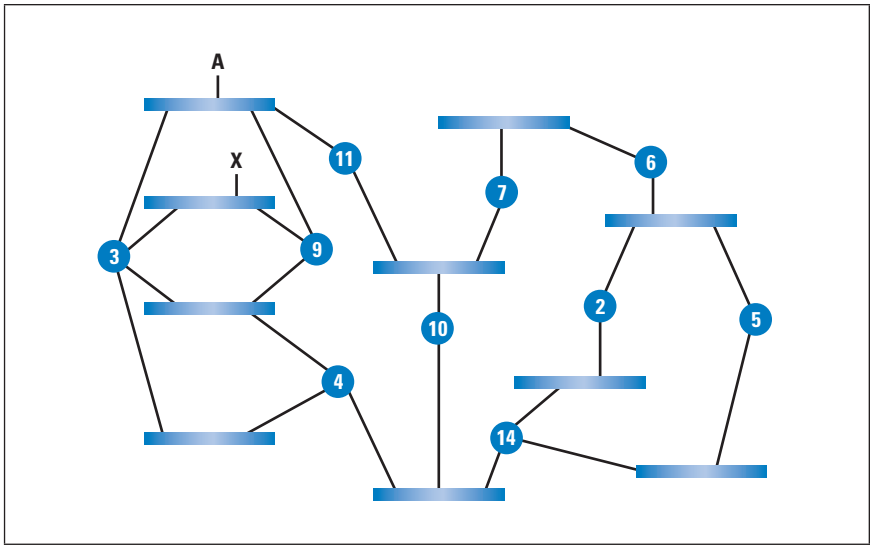


Figure 2: Bridged Network with Spanning Tree

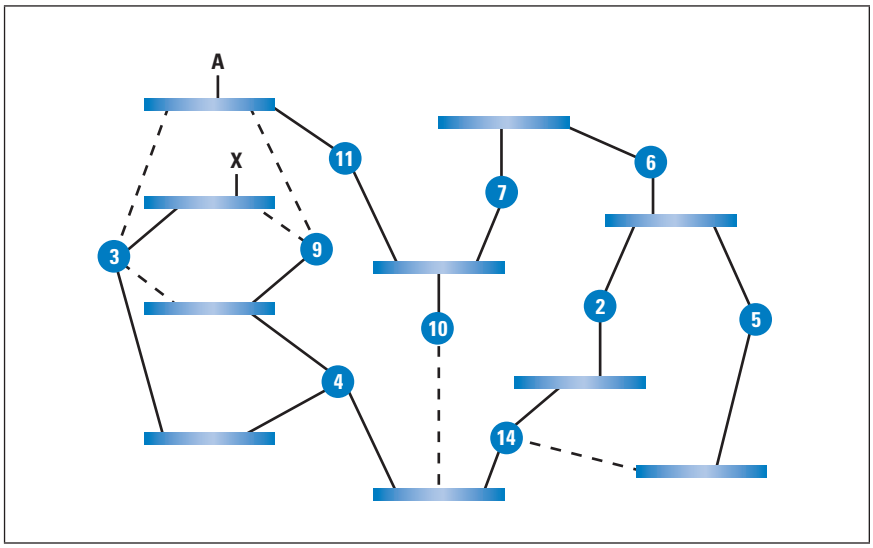
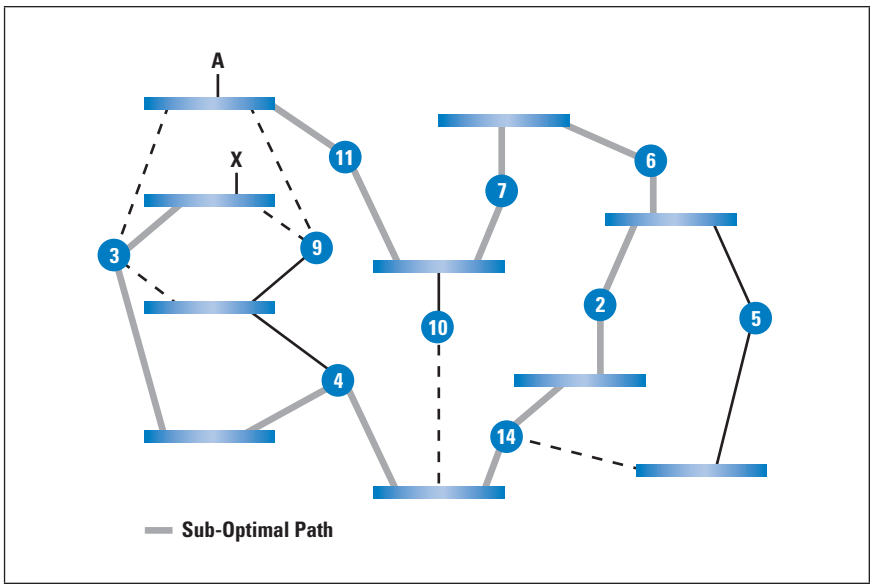


Figure 3: A Sub-Optimal Path



The spanning-tree algorithm is also inherently unstable. It requires bridges to be engineered to be able to examine every incoming packet at wire speed, to determine if the packet is a spanning-tree message, and if so, process it. The spanning-tree algorithm requires a bridge to forward unless there is a "more qualified" neighbor bridge on the link. Details of the spanning-tree algorithm, fascinating as they are, are beyond the scope of this article. If a bridge loses enough spanning-tree messages from its "more qualified" neighbor bridge because congestion overwhelms its ability to process incoming messages, the bridge will conclude that it does not have a more qualified neighbor, and therefore should start forwarding onto the link. This situation is extremely dangerous without a hop count, a field that would naturally be included in a protocol designed to be Layer 3 and forwardable.

The originally invented Ethernet, CSMA/CD, is pretty much nonexistent. Almost all Ethernet today consists of bridges connected with point-to-point links. The header still looks like Ethernet, but new fields have been added, such as VLANs discussed later in this article.

Characteristics of IP

Transparent bridging was necessitated by a quirk of history, in that applications were being built without Layer 3. But today, applications are almost universally built on top of IP. So why not replace all bridges with IP routers?

The reason is an idiosyncrasy of IP. In IP, routing is directed to a *link*, not a *node*. Each link has its own block of addresses. A node connected to multiple links will have multiple IP addresses, and if the node moves from one link to another, it must acquire a new IP address within the block for that link.

This property is not an inherent property of Layer 3, just a characteristic of IP. An alternative technology, proposed in 1992 as a replacement to IPv4, was *Connectionless-mode Network Protocol* (CLNP), an ISO packet format that had 20-byte addresses (actually, variable length). Its address, like IP, was hierarchical, routing to the longest matching address prefix in the forwarding table that matched the destination address. But in IP, the bottom level of routing was to a single link. In CLNP, the bottom level of routing consisted of routing to a cloud known as an "area," that included lots of links (typically hundreds). Within the area, end nodes announced themselves and routers routed directly to the end node. An end node could move within an area without changing its Layer 3 address. Routers within an area would not need to be configured.

In contrast, with IP, a block of IP addresses needs to be carved up to assign a unique block to each link, IP routers need to be configured with the address block for each of their ports, and nodes that move from one link to another have to change their Layer 3 addresses. Therefore, it is still popular to create large bridged Ethernets, because a bridged set of links looks to IP like a single link.

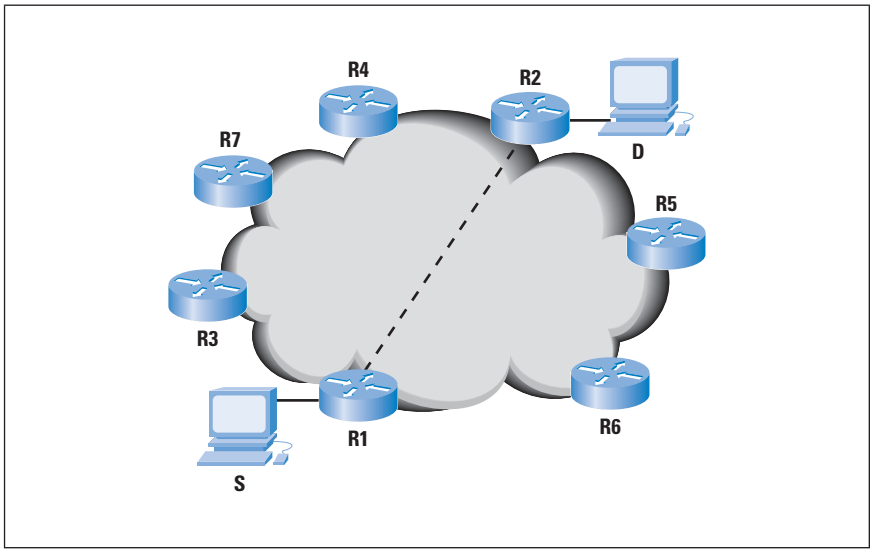
TRILL: Best of Both Worlds

TRILL allows the ease of configuration of Ethernet while benefitting from the routing techniques provided at Layer 3. It also coexists with existing bridges; it is not necessary to replace all the bridges in an Ethernet, but the more bridges replaced by RBridges, the better the bandwidth usage and the more stable the cloud becomes (because the spanning trees get smaller and smaller, and ultimately disappear if all bridges are replaced by RBridges).

Figure 4 shows the basic concepts in TRILL handling a unicast packet where the location of the destination is known:

- RBridges run a link state routing protocol, which gives each of them knowledge of the topology consisting of all the RBridges and all the links between RBridges. Using this protocol, each RBridge calculates shortest paths from itself to each other RBridge, as well as trees for delivering multidestination traffic.
- When an RBridge, R1, receives an Ethernet frame from an end node S, addressed to Ethernet destination D, R1 encapsulates the frame in a TRILL header, addressing the packet to the RBridge R2, to which D is attached. The TRILL header contains an "ingress RBridge" field (R1), an "egress RBridge" field (R2), and a hop count.
- When R2 receives the encapsulated packet, R2 removes the TRILL header and forwards the Ethernet packet on to D.

Figure 4: RBridging



What the TRILL header looks like, how R1 knows that R2 is the correct "egress RBridge," and some of the concepts in the link state protocol *Intermediate System-to-Intermediate System* (IS-IS) are described in the next section. We also explain how TRILL handles multidestination frames, VLANs, and IP Multicast.

The TRILL Header

The main fields in the TRILL header are: ingress RBridge nickname (16 bits), egress RBridge nickname (16 bits), hop count (6 bits), and a multidestination flag bit (1 bit). A typical Layer 3 header would contain a source, a destination, and a hop count. So TRILL is basically an encapsulation header with flat 16-bit addresses. How RBridges obtain "nicknames" is described later in this article.

This header is very simple for core RBridges to forward, compared with either an IP or an Ethernet header. The destination field is just 16 bits, so it can be a simple table lookup to find the entry in the output port, as opposed to the Ethernet 6-byte destination, which typically requires content-addressable memory or hashing, or the longest prefix matching of IP.

Learning End-Node Locations

How does R1 know that R2 is the correct egress RBridge for some destination D? The default mechanism is learning the correspondence between (ingress RBridge, source MAC address) when the egress RBridge decapsulates a packet. If R1 does not know where the destination MAC is located, R1 encapsulates the packet in a TRILL header with the multidestination flag set, indicating that it should be transmitted through a tree to all the RBridges.

An additional mechanism, which is optional, is known as *End-Station Address Distribution Information* (ESADI). ESADI allows R1 to announce some or all of the end nodes that are attached to R1. Both announcing to and listening to ESADI are optional. This mechanism has advantages over flooding and learning from data packets:

- ESADI packets can have cryptographic protection.
- R1 might have a more definite reason to know that S is attached to R1 than simply seeing a packet with the S address in the header. For instance, R1 might have been configured to lock down a port to the S MAC address. Or there might be a cryptographically protected enrollment protocol when S attaches to R1.
- R1 might be able to have tighter timers on verifying the location of local end nodes; for instance, if they are IP nodes, R1 might be able to ping them.

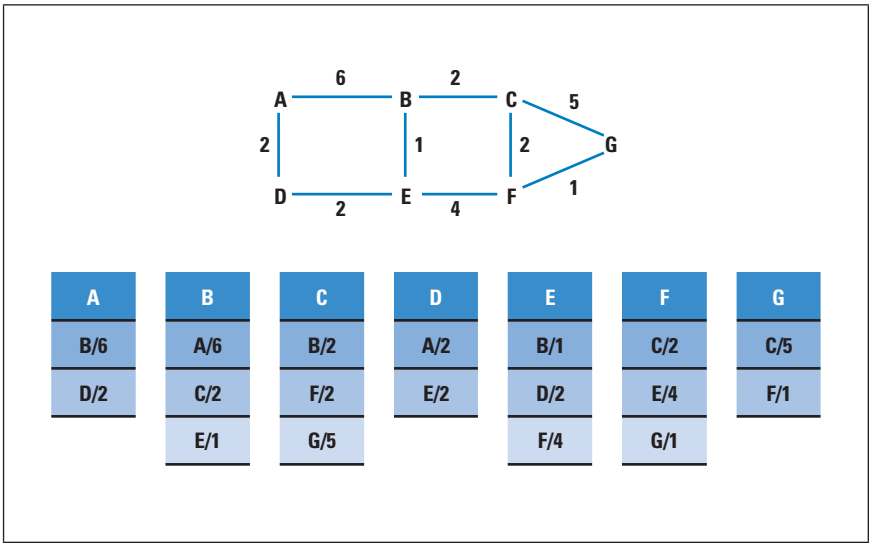
It is also possible to have a directory that lists not only (RBridge nickname, {set of attached end-node MAC addresses}) but also {(end-node IP address, end-node MAC address)} pairs. The first RBridge, or a *hypervisor,* or the end-node process itself, might query the directory about the destination, and encapsulate packets, rather than flooding, and thus also be able to bypass the IPv4 *Address Resolution Protocol* (ARP) and the *IPv6 Neighbor Discovery* (ND) protocols.

Link State Protocols

A *link state* protocol is a routing protocol in which each router R determines who its neighbors are, and broadcasts (to the other routers) a packet, known as a *Link State Packet* (LSP), that consists of information such as "I am R," and "My neighbor routers are X (with a link cost of c1), Y (cost c2), and Z (cost c3)." The commonly deployed link state protocols are *Intermediate System-to-Intermediate System* (IS-IS)[2][9] and *Open Shortest Path First* (OSPF)[10]. IS-IS, designed in the 1980s to route DECnet, was adopted by the *International Organization for Standardization* (ISO). IS-IS can route IP traffic and is used by many *Internet Service Providers* (ISPs) to route IP. IS-IS was a natural choice for TRILL because its encoding easily allows additional fields, and IS-IS runs directly on Layer 2, so that it can autoconfigure, whereas OSPF runs on top of IP and requires all the routers to have IP addresses.

Figure 5 shows a small network (at the top), consisting of 7 routers. In the bottom half of the figure, the LSP database is shown; all the routers have the same LSP database because they all receive and store the most recently generated LSP from each other router. The LSP database gives all the information necessary to compute paths. It also gives enough information for all the routers to calculate the same tree, without needing a separate spanning-tree algorithm. As we will see, TRILL requires a tree (at least one tree) for distribution of multidestination packets.

Figure 5: Router Network and Link State



Acquiring Nicknames

Given that the most recently generated link state packet of each RBridge is broadcast to, and stored by, each other RBridge, it is possible to spread other information through the link state packets, such as a protocol for acquiring a unique nickname. Each RBridge chooses a nickname at random, avoiding nicknames already acquired by other RBridges (as discovered by examining the LSP database).

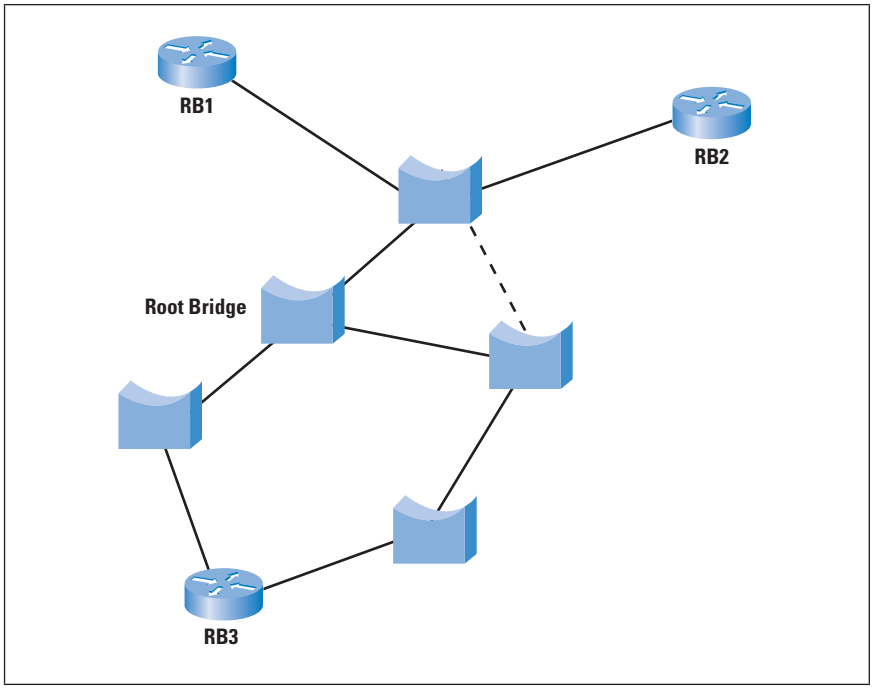
If two RBridges choose the same nickname, there is a tie-breaker, based on configured priority and 6-byte system ID. One of the RBridges gets to keep the nickname and the other RBridge has to choose another nickname that appears not to be in use.

It is possible to configure RBridges with nicknames, in which case a configured nickname takes priority over one that was randomly chosen. And in the case of misconfiguration, where two RBridges have been configured with the same nickname, again, ID and priority choose a winner, and the other one has to choose a different nickname.

Mixing RBridges with Bridges

TRILL is designed so that any subset of bridges in an Ethernet can be replaced by RBridges. A set of links connected by bridges will be perceived by RBridges as a single shared link connecting the RBridges on that link. The bridges inside that link will behave as ordinary bridges, forming a spanning tree and forwarding packets along that tree. Figure 6 illustrates an Ethernet connected by several bridges, with one port (indicated by the dashed line) selected by the spanning tree as being in backup.

Figure 6: RBridges Connected by Bridged LAN



The RBridges RB1, RB2, and RB3 perceive the link as in Figure 7, a single shared link, on which RB3 has two ports.

Introducing RBridges into a bridged Ethernet partitions the spanning trees into smaller spanning trees. RBridges operate on a topology consisting of the RBridges themselves, connected with "links" that are either bridged Ethernets or point-to-point links.

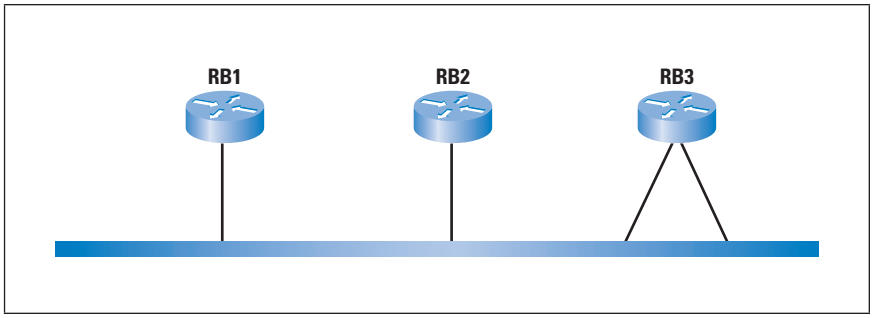


Figure 7: Figure 6 as Perceived by RBridges: a Single Shared Link Where RB3 Has 2 Ports onto the Same Link

Link Types and the Hop-by-hop Header

In addition to the TRILL header, when RBridge R1 is forwarding a TRILL-encapsulated frame to neighbor RBridge R2, there is an additional header that is specific to the type of link connecting R1 and R2. Although TRILL carries Ethernet inside, a link between two or more RBridges could be an arbitrary type of link; for example, besides Ethernet, it could be a *Point-to-Point Protocol* (PPP) link[13], an IP or *IP Security* (IPsec) tunnel, *Multiprotocol Label Switching* (MPLS) path, etc.

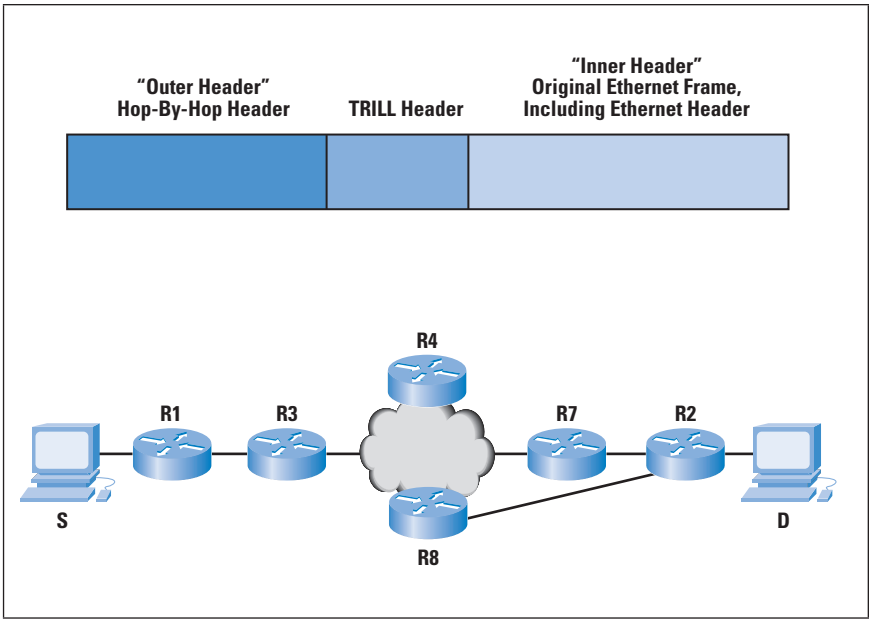
If the link is an Ethernet link, the "outer" header is an Ethernet header. If it is a PPP link, the outer header is a PPP header. The outer Ethernet header (on an Ethernet link) serves two purposes:

- If there are bridges on the link, they will perceive the packet as a normal Ethernet packet, and forward it through the spanning tree. The learning tables of the bridges on the link will see only the addresses of the RBridges on that link.
- It allows R1, when forwarding onto a link with multiple neighbors (say R2 and R3), to specify which of R2 or R3 is chosen by R1 to forward the packet by unicasting the packet to the chosen next-hop RBridge. For example, it could be that both R2 and R3 are equal costs to the destination, so R1 would need to specify which of them should forward the packet. Otherwise, both might forward the packet, and the packet would be duplicated.

So, as illustrated in Figure 8, a TRILL-encapsulated packet might have three headers:

- The outer header, or hop-by-hop header, which is stripped off at each hop, is specific to the type of link connecting neighbor RBridges, and, when forwarded between R1 and R2, it specifies R1 as source and R2 as destination
- The TRILL header, which similarly to a Layer 3 header remains in place as the packet travels from the first RBridge to the last RBridge, specifying the first RBridge (the one that encapsulated the packet with a TRILL header) as the ingress RBridge, and the last RBridge (the one that will decapsulate the packet) as the egress RBridge
- The inner Ethernet header, which specifies the communicating end-node pair as source and destination

Figure 8: TRILL Packet Headers



Again referring to Figure 8, assume S transmits an Ethernet packet to D. In the inner Ethernet header, Source = S, Destination = D.

R1 encapsulates it with a TRILL header, where ingress RBridge = R1 and egress RBridge = R2. R1 forwards it to R3, putting on a link header appropriate to the link. If the link is an Ethernet link, the outer Ethernet header will indicate S = R1, D = R3. When R3 forwards to R7, R3 leaves the TRILL header as is (other than decrementing the hop count), strips the outer header, and puts in a new outer header indicating S = R3, D = R7. Likewise, R7 forwards to R2. If it is a PPP link, there is no source or destination. When R2 forwards to D, R2 strips off the TRILL header and D sees the Ethernet packet exactly as transmitted by S.

VLANs

Ethernet has a concept known as a *Virtual LAN* (VLAN), which partitions communities of end nodes sharing the same infrastructure (links and bridges), such that end nodes in the same set can talk directly to each other (using Ethernet), whereas those in different VLANs have to communicate through a router. IP nodes, although generally unaware of Ethernet VLAN tags, perceive different VLANs to be different IP subnets.

Typically, a bridge is configured with a VLAN for each port, and the bridge adds a tag to the Ethernet header that indicates which VLAN the packet belongs to. A bridge with a port that is configured to be VLAN x will deliver only packets tagged as VLAN x to that port, and will usually strip the VLAN tag before forwarding.

The original Ethernet did not have a VLAN concept. In today's Ethernet standard, each packet must be associated with a VLAN. A bridge might be configured with a default VLAN for a port, meaning that if no VLAN tag is in the packet, the bridge will treat it as if it is that default VLAN. A bridge B might be configured in various ways that make VLANs more complex:

- B might be configured to drop a set of VLANs rather than forward them onto a particular port, even though the port is a transit port.
- B might be configured to modify the VLAN tag to a different value when forwarding from one port to another.
- B might be configured to remove the VLAN tag when forwarding onto a particular port.

Appointed Forwarders

If there are multiple RBridges on the same link, together with end nodes, it is important that only one of them encapsulate a packet from an end node. As illustrated in Figure 9, if both R1 and R2 were to encapsulate a unicast packet from S, two copies would be delivered to the destination. However, if S were to transmit a multidestination packet (such as a multicast, or an unknown destination), then the copy that R1 encapsulates would be forwarded through the campus, received by R2 (which likely would not know that the packet originated on its port to R1), and R2 would decapsulate it. Then R1 would see a native packet from S, exactly as the first copy, and again encapsulate it and send it into the campus.

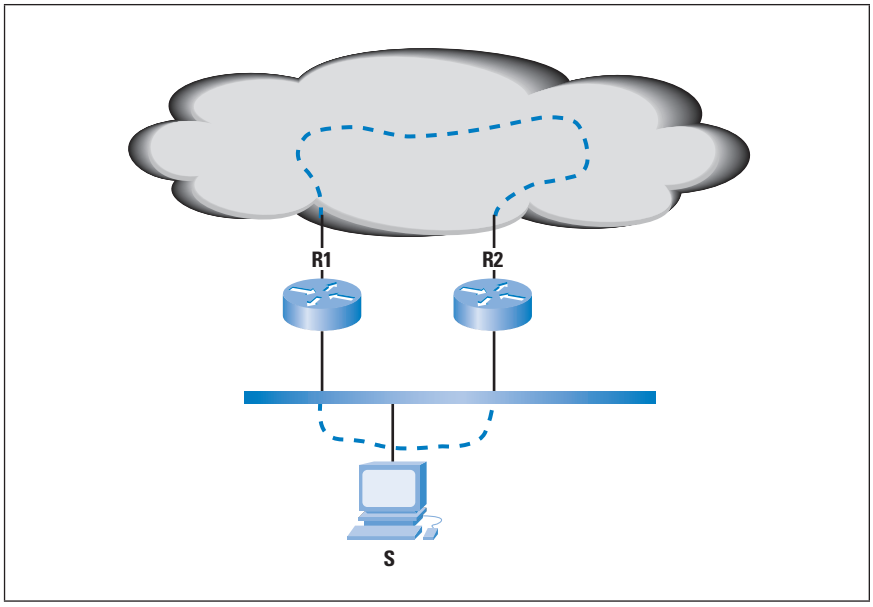


Figure 9: Link with Multiple RBridges. Note: No Hop Count Protection on Native Frame.

The hop count in the TRILL header would not solve this loop, because the hop count does not exist while the packet is not encapsulated with a TRILL header.

IS-IS has an election protocol in which one of the RBridges is elected as the *Designated RBridge* (DRB). In order to allow load-splitting the task of encapsulating and decapsulating traffic, the DRB may delegate the job of encapsulation/decapsulation based on VLAN. In other words, if R1 is DRB, R1 can delegate to R2 the task of encapsulating/decapsulating traffic for a set of VLANs, say VLANs x, y, and z, and delegate to R3 a different set of VLANs, and R1 might handle the rest.

Implications of VLANs on TRILL

TRILL treats VLANs strictly as a way of partitioning the end nodes, in contrast with IEEE, which allows bridges to drop transit traffic based on VLAN. Consequently, an Ethernet link connecting TRILL RBridges R1 and R2 might be able to deliver packets tagged with VLAN x, but not deliver packets tagged with VLAN y.

It is important, as shown in Figure 9, that all the RBridges on a link know about each other; otherwise they might both encapsulate a packet.

The IS-IS election is done through Hello messages, whereby RBridges announce themselves. Unfortunately, possible configuration of bridges, whether intentional or by mistake, can partition a link for traffic marked as VLAN y, but have the link be connected for traffic marked as VLAN x. This situation complicates the IS-IS election. When transmitting a Hello message onto an Ethernet link, an RBridge R1 must assign it to a VLAN. If R1 chooses VLAN y, its neighbor R2 might not see the Hello message. And then, unaware that there were multiple RBridges on the link, both R1 and R2 might encapsulate a VLAN x packet.

TRILL handles this situation by having the DRB (by default) transmit Hello messages on all the VLANs for which it is enabled on the port. The DRB chooses a VLAN, say VLAN A, for inter-RBridge communication on the link, and informs the other RBridges on the link that they should use VLAN A. The other RBridges transmit IS-IS messages (including Hello messages and LSPs) and encapsulated TRILL packets, putting VLAN A in the outer header. The VLAN tag in the inner header is the one that represents the community that the end node belongs to. The VLAN tag in the outer header is only for the purpose of traversing an Ethernet hop between RBridges.

Additionally, (by default), an RBridge that is Appointed Forwarder for a VLAN, transmits Hello messages on that VLAN.

If it is known that there are no bridges, the RBridges (including the DRB) can be configured to send Hello messages only on the single VLAN specified by the DRB.

Modified Hello Protocol

IS-IS has an election protocol in which routers (or RBridges in the case of TRILL) send Hello messages. Not only does the Hello message transmitted by R1 announce R1 to its neighbors, but the R1 Hello message contains a list of neighbors that R1 has heard Hello messages from. R2 will not consider R1 to be a neighbor unless R2 sees itself listed in the Hello messages of R1, indicating connectivity is two-way. When choosing a DRB, R2 ignores any routers for which connectivity to R2 is not two-way. Therefore, if there were a shared link with strange connectivity properties, the routers on the link might partition into cliques, each with its own DRB, each clique representing a separate link to the rest of the routers.

A surprising aspect of the use of IS-IS for TRILL was that the Hello protocol had to be modified slightly. In Layer 3 IS-IS, Hello messages are padded to the maximum size, because a possible hardware failure mode was that a link between R1 and R2 might be able to transmit small packets, but not large packets. In Layer 3, the IS-IS assumption was that R1 and R2 would rather not see that they were potential neighbors than use a flaky link. In IS-IS, LSP packets can be fragmented only by the source R1. All routers agree upon the maximum size of an LSP fragment that is guaranteed to be able to traverse all the links. Links that cannot forward packets of that size are not reported in the topology, and indeed, in Layer 3 IS-IS, would not even be discovered in the topology, because the Hello message (padded to that size) would not be seen by the neighbor router.

But with TRILL, it is important that only a single RBridge be elected DRB, because the DRB determines which RBridge will encapsulate/decapsulate packets for each VLAN. One of the first implementations of TRILL wound up forming a loop, where two RBridges, R1 and R2, both performed encapsulation/decapsulation. This situation resulted because neighbors R1 and R2 did not see each other's Hello messages, because the R1 Hello, padded to classic Ethernet maximum size by R1, became too large to forward when a VLAN tag was added, so did not reach R2.

To ensure that only a single RBridge on a link would be elected DRB, TRILL modified the Hello protocol as follows:

- Limit the size of Hello messages and do not pad them (in order to remove artificial impediments to receipt by neighbors).
- Elect a DRB based solely on priority (not two-way connectivity as in Layer 3 IS-IS). In other words, defer to a higher-priority RBridge R1 even if R1 does not list you as a neighbor.
- Have a separate mechanism for probing, using packets of different sizes, to see what size packets can be forwarded on the link.

In addition to solving the multiple-DRB problem, this design enables TRILL to discover which links can handle jumbo-grams, so that paths can be engineered that can forward jumbo-grams.

If the link between R1 and R2 is not acceptable because it cannot handle the assumed LSP fragment size, or because connectivity is not two-way, the link is not reported in LSPs. The capability of a link to handle larger sizes can be reported in LSPs.

There was enough confusion about this minor change to the Hello protocol, and skepticism that the Hello mechanism, which has worked correctly for Layer 3 for decades, would need to be modified for TRILL, that an additional RFC was written [3] to specifically explain the TRILL Hello mechanism.

Multidestination Frames

Multiple Trees

The original design for TRILL had the RBridges compute a single, shared tree, based on the LSP database, and all multidestination traffic was forwarded along that tree. But, to be able to load-split the use of links for multidestination traffic, a facility for using multiple trees was added early in the development of the TRILL standard.

In TRILL, the RBridge with the highest priority to be a TREE root announces to the other RBridges (through its LSP) how many trees, and which trees, should be calculated. A tree is calculated as a tree of shortest paths from a given Root, with a deterministic tie-breaker so that all RBridges calculate the same tree. The Root can be an RBridge or a pseudonode. In some cases, a Root is particularly well-situated in the topology such that its tree forms good paths for all pairs of nodes, but it is desirable to have multiple different trees, choosing different tie-breaker links, calculated from the same Root. TRILL accomplishes this setup by having that Root acquire multiple nicknames, one for each tree, and using the tree number in the tie-breaker algorithm, so that although all the trees from that Root will still be shortest-path trees, different links will be chosen in the different trees.

When R1 encapsulates a multidestination frame, R1 sets the "multi-destination" flag and specifies the tree Root nickname in the "egress RBridge" field in the TRILL header.

Filtering

A multidestination frame will be tagged with a VLAN (in the inner header). The frame need not be delivered to all RBridges—just those that are connected to a port with end nodes in that VLAN. So RBridges announce, in their LSPs, which VLANs they are attached to, where "attached to," means that they are acting as Appointed Forwarder.

Additionally, TRILL provides for filtering based on Layer 2 MAC addresses derived from IP Multicast groups. RBridges announce the set of such MAC addresses they wish to receive. The first RBridge that accepts an IP Multicast control message, such as *Internet Group Management Protocol* (IGMP), snoops on it [5] and learns what multicast listeners or multicast router is attached. This snooping is used so R1 can report in its LSP the IP Multicast groups it wishes to receive (or all groups if a multicast router is attached).

One other refinement to multidestination is the *Reverse Path Forwarding* (RPF) check. To safeguard against loops, when R is calculating which subset of its ports belong to a particular tree, R also calculates, for each port, the set of ingress RBridges whose traffic on that tree should arrive on that port.

So, the processing of a multidestination frame received by R, with TRILL header indicating Ingress = R1 and Egress/tree Root = R2, is as follows:

- If the port on which R receives the packet is not included in the tree "R2," discard the packet.
- If the port on which R receives the packet is in tree R2 but R1 is not listed in the RPF information for that port for tree R2, discard the packet.
- For each other port in R2, if the specified VLAN is reachable through that port and the IP Multicast address is requested by an RBridge along the path through that port, forward the packet on that port.

IS-IS Pseudonodes

If there is a link with N RBridges, rather than modeling the link as having on the order of N^2 links to be reported in LSPs, IS-IS has the DRB model the link as a pseudonode. The DRB gives the pseudonode a name, and the RBridges on the link report connectivity just to the pseudonode. The DRB generates an LSP on behalf of itself, reporting connectivity to the pseudonode, but additionally generates an LSP on behalf of the pseudonode, reporting connectivity to all the RBridges on the link. This portion of IS-IS is as designed from the beginning (from its origin as Phase V DECnet routing).

When IS-IS was originally designed, Ethernets tended to be very large shared links. But today, most Ethernets are simply point-to-point links (unless there are bridges making them appear to be shared links). So it would be wasteful for RBridges to always create a pseudonode for each Ethernet link. In Layer 3 it is not as unreasonable to always treat an Ethernet as a large shared link because an "Ethernet" link, as perceived by Layer 3, is likely to be a large collection of point-to-point links glued together with either bridges or RBridges.

But RBridges are likely to often see Ethernet links with just a single neighbor, especially in a topology with no bridges. So TRILL has the ability for the DRB to specify to its neighbor RBridges whether to report the link as a pseudonode or to report connectivity to all the RBridge neighbors as separate links. By default, the DRB R sets a flag known as the "bypass pseudonode" flag in its Hello message on the link, unless at some point since R rebooted R has seen two simultaneous neighbor RBridges on that link. With this mechanism, true point-to-point Ethernet links will be reported as a link between R1 and R2 rather than a pseudonode P, with links R1–P, R2–P, and P–R1 and P–R2 reported.

TRILL Implementations

TRILL is being widely implemented. TRILL fast-path hardware is included in chips available from all major merchant silicon manufacturers. A successful interoperability test was held at the University of New Hampshire *InterOperability Laboratory* in late 2010, and TRILL products are announced and shipping.

Future Potential TRILL Enhancements

Here are just three enhancements to TRILL being considered:

- Data centers require more VLANs than can be specified in 12 bits with a single VLAN tag. A TRILL extension to optionally include the ability to encode 24 bits of VLAN-like labeling in TRILL data frames is being considered.
- By optionally giving a pseudonode a nickname and having the appointed forwarder use that nickname in the ingress RBridge field, if the appointed forwarder changes, the end-node learning cache of distant RBridges will still be correct.
- A proposal is being made allowing IS-IS to be hierarchical in a TRILL campus. IS-IS hierarchy partitions the LSP database so that any single RBridge LSP database will be smaller, its path computation will be less computation-intensive, and it will lower the amount of LSP traffic. In particular, it shields the effects of a link that is cycling quickly from most of the campus, because only the RBridges in the region with the link will see reports of the state of that link.

Summary

The TRILL standard creates a cloud with a flat Ethernet address, so that nodes can move around within the cloud and not need to change their IP address. Although nodes attached to the cloud perceive the cloud as an Ethernet while the packet is traversing the cloud, it is encapsulated with a TRILL header, which like a Layer 3 technology, contains a source (ingress RBridge), destination (egress RBridge), and hop count. The addresses in the TRILL header are 16 bits, enabling a TRILL campus to support 64,000 RBridges. Transit RBridges do not learn about location of end nodes—only the existence of, and path to—other RBridges.

TRILL can use all the Layer 3 techniques, including shortest paths, *Equal Cost Multipath* (ECMP), and traffic engineering. It also supports VLANs and multicast. TRILL can calculate multiple trees, so that multidestination traffic can be split across links. Multi-destination frames can be filtered based on VLAN and IP (v4 or v6) Multicast groups.

TRILL is compatible with existing Ethernet bridges (switches), so a bridged Ethernet can be gradually upgraded by replacing any subset of the bridges with RBridges. The more that are upgraded, the better the bandwidth usage, and the more stable the network becomes.

References

[1] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification," RFC 6325, July 2011.

[2] "Information technology—Telecommunications and information exchange between systems—Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)," ISO/IEC 10589:2002.

[3] Eastlake 3rd, D., Perlman, R., Ghanwani, A., Dutt, D., and V. Manral, "Routing Bridges (RBridges): Adjacency," RFC 6327, July 2011.

[4] ITU-T, "X.200: Information technology—Open Systems Interconnection—Basic Reference Model: The basic model," July 1994.

[5] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches," RFC 4541, May 2006.

[6] Touch, J. and R. Perlman, "Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement," RFC 5556, May 2009.

[7] W3C, "XML Base (Second Edition)," W3C Recommendation 28 January 2009,
`http://www.w3.org/TR/2009/REC-xmlbase-20090128/`

[8] Perlman, R., "A Protocol for Distributed Computation of a Spanning Tree in an Extended LAN," *9th Data Communications Symposium,* Vancouver, 1985.

[9] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments," RFC 1195, December 1990.

[10] Moy, J., "OSPF Version 2," RFC 2328, April 1998.

[11] Simpson, W., "The Point-to-Point Protocol (PPP)," RFC 1661, July 1994.

[12] **http://www.interfacebus.com/HDLC_Protocol_Description.html**

[13] Carlson, J. and Eastlake 3rd, D., "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol," RFC 6361, August 2011.

RADIA PERLMAN is a Fellow at Intel Labs, working on the design of various network routing and security protocols. She is the inventor of the Spanning Tree Algorithm, the designer of IS-IS, and the original concept for TRILL. She is the author of the textbook *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*. She is an IEEE Fellow and holds a Ph.D. from MIT.
E-mail: **radiaperlman@gmail.com**

DONALD EASTLAKE 3rd is Co-Chair of the IETF TRILL Working Group and a voting member of IEEE 802.1. He is the author of 56 IETF RFCs and a Principal Engineer with Huawei Technologies working on advanced network product research. Previously, he was a Principal Engineer at Cisco Systems and before that a Distinguished Member of Technical Staff at Motorola Laboratories, working on network protocols, security, and mesh networking.
E-mail: **d3e3e3@gmail.com**

The Case for IP Backhaul

by Jeff Loughridge, Brooks Consulting LLC

In any hierarchical network, designers must specify how the access layer delivers traffic to the core. In *Mobile Network Operator* (MNO) networks, the transport of voice and data from the cell sites to the wireless MNOs' core networks is called *backhaul. Time Division Multiplexing* (TDM) backhaul has dominated backhaul deployments since the inception of wireless communication. Leasing the backhaul access of multiple T1s/E1s for every cell site becomes prohibitively expensive in terms of operating expenses, particularly for providers that do not own the last mile. Today's 3G/4G cellular technologies have spurred a major change in the backhaul network: the transition from TDM to packet backhaul.

Ethernet is the most widespread packet-based backhaul technology. While this service is a vast cost and scale improvement over TDM backhaul, carrier Ethernet is a stepping stone in the evolution of backhaul networks. Expect MNOs to move to true IP backhaul networks to meet the scalability needs of their expanding networks. In this article, we will explain mobile backhaul evolution, shortcomings in carrier Ethernet backhaul, and how evolving service requirements will motivate cell site backhaul vendors to add IP-awareness to their networks.

Legacy Backhaul

Cellular systems were initially designed to carry only voice traffic. Since transporting digitized voice was a mature and well-understood technology, there was no need to take a divergent path for the backhaul of voice traffic in early cellular systems. Using TDM had obvious advantages among those being:

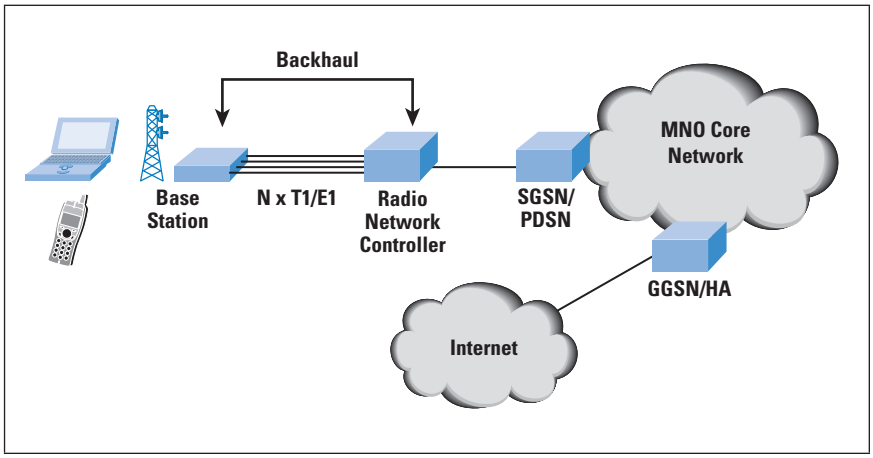
- Use of the same equipment used in wireline voice transmission
- Technical staffs' familiarity with TDM concepts and troubleshooting
- Ability to use existing *Operations, Administration, Maintenance, and Provisioning* (OAM&P) systems
- Ubiquity of the T1/E1 service

The initial work to offer data service on cellular systems naturally focused on adding data transmission to the existing voice infrastructure. Standards such as *Global System for Mobile Communications* (GSM) and *Interim Standard 95* (IS-95) took similar approaches in borrowing TDM time slots for data. The data services of the 1990s were very slow, even when compared to consumer modems of the time. Standards developed in the late 1990s and deployed in the early 2000s (*Enhanced Data rates for GSM Evolution* [EDGE] and *CDMA2000*) improved data transfer speeds.

TDM was clearly entrenched as a foundational technology for data communication in cellular networks going into the early 3G technology deployments (*Universal Mobile Telecommunications System* [UMTS] and *Evolution Data Optimized* [EV-DO]).

Figure 1 depicts the backhaul portion of the MNO network and how it fits into the broader architecture.

Figure 1: The Backhaul Network in the MNO Architecture



As data traffic usage for 3G networks grew, shortcomings of TDM backhaul began to materialize. The two prominent areas were bandwidth and cost. Cell sites with TDM access are typically equipped with multiple T1/E1s. With faster radio interfaces, the backhaul became the bottleneck in the network. Some smartphones became consumers of multi-megabyte data rates. User experiences were poor on some wireless networks as a result of a dearth of bandwidth in the backhaul segment. Continuing to increase the number of TDM lines or increase their capacity was not a viable option since the growth increments were too small and the operating expenses were too high.

The second limitation of TDM in 3G networks is cost. Although the cost of T1/E1s decreased considerably over the years, the costs piled up given the number of cell sites and number of T1/E1s per site. This figure became the highest contributor to the cost of the backhaul network. The MNOs that owned the last mile were at a distinct competitive advantage compared with the carriers who had to pay another party (often in a minimally competitive marketplace) for TDM access. For MNOs to continue their incredible traffic growth rates, a new access model was needed.

Carrier Ethernet Adoption

Ethernet quickly emerged as the most popular backhaul technology to replace TDM access infrastructure (other providers moved forward with microwave access with varying levels of success). The various iterations of Ethernet from 1970s to 2000s had trumped other LAN technologies in the market, and at the turn of the century gigabit Ethernet leveraged its success in the LAN to become popular in the WAN. The technology had several major advantages:

- *Large drop in cost per bit:* Ethernet would allow providers to drastically alter their access cost model by supplanting the aging and costly TDM infrastructure. With the price that consumers were willing to pay per month of data service staying relatively stagnant, this adjustment to the cost model was critical.
- *Ethernet can be carried over more underlying technologies: Synchronous Optical Networking/Synchronous Digital Hierarchy* (SONET/SDH), *Generic Framing Procedure* (GFP), *Dense Wavelength Division Multiplexing* (DWDM), and *Multiprotocol Label Switching* (MPLS) are a few examples. A key benefit Ethernet's ability to operate over these technologies was that many providers could consolidate their wireless access with their existing and speedier wireline access networks.
- *Ethernet interfaces ubiquitous and inexpensive:* Ethernet won the battle for LAN dominance. The technology was not restricted to traditional personal computers and servers—printers, phones, game consoles, *Digital Video Recorders* (DVRs), and home media center hubs are some examples of other equipment that often included Ethernet interfaces. This ubiquity in the business and consumer spaces results in a diverse supplier set and economies of scale for the vendors and suppliers.
- *Ease of bandwidth upgrade:* TDM circuits have an implementation time measured in months. This slow turn-around time for upgrades is a poor fit for an environment in which data usages is increasing at fast rates. Ethernet is much different. An increase in bandwidth to a network end-point will not require a change in equipment unless moving between the established tiers of 10, 100, 1000 Mb/s. Since the Ethernet service vendor likely uses a "policer" to keep customers within the purchased bandwidth level, a change in software configuration is usually all that is required to upgrade bandwidth. Another advantage is that bandwidth can be upgraded in granular increments. With the right back-end systems, an upgrade will take a matter of minutes. For companies looking to increase the velocity of service deployment, the ability to quickly move to high speeds is very favorable.

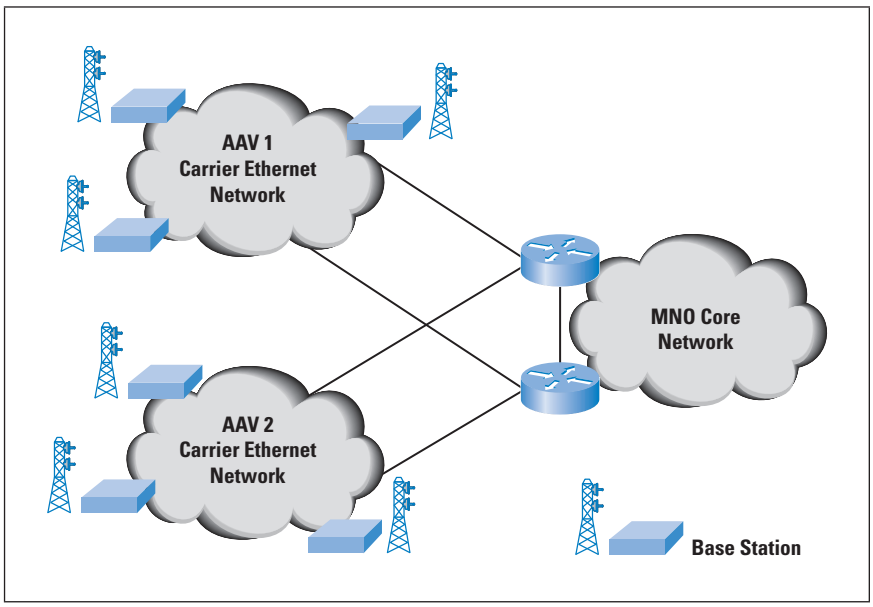
Established in 2001, the *Metro Ethernet Forum* (MEF) played a critical role in the acceptance of carrier Ethernet by wireless and wireline providers. The MEF is not a standards organization like the *Internet Engineering Task Force* (IETF). Instead, the MEF builds upon the work of standards bodies to establish common terminology, service requirements, and network interface requirements. The MEF created an architecture framework along with measurement and testing specifications. Although the MEF did not eliminate wireless providers' concerns about packet backhaul—particularly in the areas of jitter, delay, and packet delivery, the forum did increase the comfort level associated with metro Ethernet services. The MEF's E-LINE service definition established a connection-oriented path, a concept much more pleasing to traditional telcos than the perceived "anything goes" nature of packet switched networks. For more detail on the MEF's service definitions, see [0].

By the second half of the 2000s, many wireless providers were planning the deployment of Ethernet-based backhaul for new *High Speed Packet Access* (HSPA), *Worldwide Interoperability for Micro-wave Access* (WiMAX), and *Long-term Evolution* (LTE). In making this radical change, the providers often had to consider protecting existing revenue streams from voice and data (providers electing to move forward with greenfield deployments were at a luxury). Pseudowire technologies enabled the carriage of TDM traffic over IP/Ethernet networks, thus preserving investment in existing infrastructure.

Rather than build carrier Ethernet infrastructure, the MNOs that were not facilities-based (or had limited last mile footprints) purchased services from other parties, known as *Alternate Access Vendors* (AAV) in telco parlance In the United States, the *Local Exchange Carriers* (LECs) and cable companies were well positioned for this business. MNOs often used multiple AAVs in a given market to cover the cell site footprint. Getting fiber to cell sites outside of major metropolitan areas was not always possible, which led some MNOs to use hybrid backhaul solutions that included microwave and TDM inverse muxing in addition to carrier Ethernet.

Figure 2 illustrates how MNOs rely on AAVs to cover their cell site footprint in a given market.

Figure 2: Alternative Access Vendors



The adoption of carrier Ethernet services by MNOs was not without challenges. Mobility gear such as *Radio Network Controllers* (RNC), base stations, and *Home Location Registers* (HLR) historically relied on T1/E1 interfaces for connection to the network. Telecom vendors had to implement Ethernet interfaces along with IP stacks. The providers had to completely revamp provisioning, service monitoring, performance monitoring, and service assurance systems and processes. Consider the following example.

For years, operations groups at telcos counted on near-immediate notification with an alarm indication signal in the *Time Division Multiple Access* (TDMA) frame. TDMA frames arrive every 125 µsec (8,000 times a second). Packet-switched networks do not share the synchronous nature of TDM and do not have OAM fields in framing bits. The operators now had to rely on nascent specifications such as Y.1731 and 802.1ag for service monitoring.

Timing and synchronization—necessities in mobile networks—are gleaned from the physical layer in TDM networks. Asynchronous networks such as Ethernet/IP do not have an inherent mechanism for timing and synchronization. Keeping a single T1/E1 at the cell site is one method to ensure timing and synchronization in a carrier Ethernet scenario; however, the use of upper layer protocols is more appropriate, particularly for new builds that have no legacy TDM circuits. *Synchronous Ethernet* (SyncE), *Precision Time Protocol* (PTP, also known as IEEE 1588v2), and *Network Time Protocol version 4* (NTPv4) were deployed in backhaul networks to provide timing and synchronization. Note that SyncE transports timing information over the physical layer much like the TDM timing model, while PTP and NTP use IP for transport and are not dependent on an Ethernet physical layer.

The learning and flooding aspects of all Ethernet networks present inherent scaling challenges for very large networks. Spanning tree and its derivatives are commonly used to address these issues at low and medium scale. For larger networks that provide service to multiple customers, the service must scale in terms of its ability to offer service to multiple entities and in terms of the many switches required for an expansive footprint. Many protocols have arisen to solve one or both of these challenges. Examples are *Virtual Private LAN Service* (VPLS), *Multiprotocol Label Switching–Transport Profile* (MPLS-TP), and *Provider Backbone Bridging–Traffic Engineering* (PBB-TE). Being relatively new technologies, these can and do present challenges for operations groups. The breakages can occur in ways that are very difficult for the Carrier Ethernet provider and wireless provider to jointly troubleshoot.

The Next Step – IP Backhaul

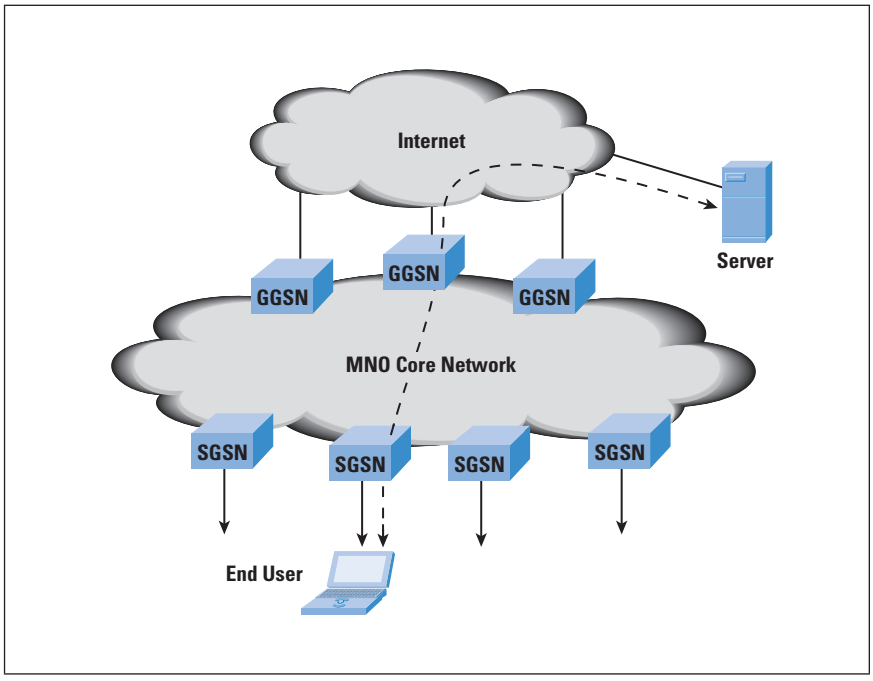
The phrase "all-IP" is frequently used to describe the most recent wireless technologies such as HSPA+, WiMAX, and LTE. This is applicable as the majority of network elements, including the handsets, are IP enabled. The existence of large-sized carrier Ethernet networks in the network architecture undermines the IP-centric argument. IP has superior scaling properties over Layer 2 networks. The footprint and number of nodes for carrier Ethernet networks continues to expand rapidly as the MNOs deploy 3G and 4G networks. The author sees evidence that protocols used to overcome Ethernet scalabilities issues will become increasingly complex and push MSOs and AAVs toward Layer 3-centric backhaul networks.

Before delving into the drivers of IP backhaul, let's examine a typical data traffic flow for today's wireless networks. We'll use the 3GPP's *GSM Packet Radio System* (GPRS) as this is the most common in world-wide deployments. Data flows are very centralized in this architecture. Macro-level mobility is controlled by two types of *GPRS Support Nodes* (GSN): *Gateway GPRS Support Nodes* (GGSN) and *Serving GPRS Support Nodes* (SGSN). GGSNs are typically deployed within the mobile core network at locations with Internet access. This is often at centralized mobile switching centers. SGSNs can be deployed closer to the network edge and multiple SGSNs can be served by a single GGSN.

The GGSN is the mobility anchor, much like the home agent in wireless networks that use Mobile IP. The SGSN is akin to the foreign agent in Mobile IP. GPRS network tunnel traffic between SGSN and GGSN using an IP-in-IP tunneling protocol called *Generic Tunneling Protocol* (GTP). Although GTP has several purposes in the GPRS core network, our focus will be on its tunneling of packets between SGSN and GGSN (called the *Gn* interface). The movement of the subscriber to a region served by another SGSN will trigger a macro-mobility event. A new GTP tunnel is formed using the original GGSN for session continuity [2].

Since all traffic from the *Mobile Subscriber* (MS) must traverse the GGSN as the mobility anchor, the traffic flow from the MS follows a very predictable path to a centralized location. Note that there is not a 1:1 relationship between SGSNs and GGSNs. As mentioned earlier, typical deployment of GGSNs is very centralized. Figure 3 depicts the flow.

Figure 3: Data flow in a GPRS Network



Although technologies like LTE are touted as flat IP networks, this only holds true from a *Radio Access Network* (RAN) perspective. What if a subscriber wants to communicate with another subscriber in the same building or local machine-to-machine traffic is highly sensitive to latency? The packets will be sent to the mobility anchor, perhaps hundreds of kilometers away. Routing decisions can be made in the RAN and core network; however, the decision is restricted since traffic must traverse the predefined tunnel endpoints.

Wireless networks will gradually decentralize and distribute mobility management. In 3G networks, some providers have been extending the core network closer to the subscriber as mobile gateways (GSNs and their equivalents in non-3GPP networks) become more cost-competitive. By deploying mobile gateways at what were previously aggregation *Points Of Presence* (POPs) and buying Internet connectivity at these locations, Internet-bound traffic exits the network quickly, consuming fewer resources for the provider. Other signs of this shift are evident in LTE and WiMAX. LTE's S1-flex interface allows the RAN to be connected to multiple core networks. The WiMAX reference model separates the *Network Access Provider* (NAP) and *Network Service Provider* (NSP). The NAP, which provides radio access functionality, can connect to multiple NSPs for Internet connectivity.

To fully realize the benefits of an IP-centric backhaul, steps must be taken to go beyond simply distributing mobility management. New solutions are needed to eliminate mobility anchoring via tunneling. Vendors, providers, and universities have already started to examine how to dispose of tunneling in the mobile environment [2].

The IP-centric backhaul network has many advantages over the carrier Ethernet networks that enable many of today's packet backhaul networks. Various advantages benefit the wireless providers, the IP backhaul provider, or both. These advantages are most prevalent when the MSOs have a highly distributed mobility management architecture.

- *Backhaul Offload:* Today's mobile elements at the cell tower have no ability to influence routing decisions; there is only one path to the core network. Adding egress points to the cell site or backhaul network reduces the distance and amount of traffic that must be backhauled. To accomplish the addition of egress points in a carrier Ethernet network, connection-oriented mechanisms such as Ethernet Virtual Circuits would require that the MSO and AAV modify multiple network elements' configurations. Offloading traffic with an IP network is substantially more simple and scalable. Offloading packets from the backhaul will represent a huge savings in access costs. The base station could be capable of hot potato routing traffic directly to an ISP instead of backhauling commodity Internet traffic to the MSO, where the costs of equipment, power, and software licenses quickly accumulate.

- *Multicast:* The reliance on tunneling as described earlier in this piece severely restricts the usefulness of multicast in current wireless networks. Distributing the mobility elements controlling the tunneling closer to the subscriber will mitigate these effects as would the elimination of mobility anchoring via tunneling techniques. The implementation of a true flat IP network would extend multicast capability into the RAN and position both MNOs and IP backhaul providers to realize the efficiency gains of multicast.
- *Localized Content and Peering:* With localized egress points, local content could be reached directly rather than traversing the core network. This would position wireless providers to peer with other providers at the local or regional level, a benefit that would be substantial for wireless providers operating in countries with non-meshy Internet infrastructure and expensive wide-area communications lines. In addition, caches could be implemented much closer to the subscriber to improve the user experience for video and other content types.
- *Machine-to-Machine (M2M) and Peer-to-Peer (PtP):* When the communication is device to device in close geographic proximity, the traversal of the core network only adds latency, complexity, and cost. A distributed mobility management architecture and IP backhaul network engender an optimized path for M2M and PtP. The mobility anchor point could be placed at the cell tower or local aggregation point, providing a much improved communication path for subscribers and machines connected to the wireless network.
- *Uptime and Reliability:* Wireless providers have experienced challenges with carrier Ethernet service. Some of these problems can be chalked up to the relative newness of using carrier Ethernet for cell site backhaul. One has to wonder though, what experience exists in the industry for maintaining giant Layer 2 networks? The number of mobile devices will expand exponentially, triggering the deployment of thousands of new cell sites, microcells, and picocells. The author is less than confident that any underlying technology that enables carrier Ethernet will scale to the necessary degree while maintaining the uptime and reliability that users expect from their data service.

For large IP networks, the industry has over fifteen years' experience in designing, engineering, and operating IP networking carrying traffic at staggering capacities. The staff expertise, software maturity, and systems support exists today to maintain sizable IP networks. There are established best practices for Tier 1 ISPs that help ensure long uptime, speedy convergence upon failure, and sound network design.

Delivering an IP Backhaul Service

IP backhaul offerings could be delivered in a variety of ways. The simplest design for IP backhaul providers would be a shared IP transport network that commingles traffic between customers.

The wireless providers could then use protocols such as *Layer 2 Tunneling Protocol version 3* (L2TPv3) to build an MPLS/VPN-like overlay to provide logical separation and address overlap prevention. The preferred approach for MNOs would likely be a Layer 3 VPN service from the AAV, thereby offloading much of the routing complexity from the MNO.

An IP backhaul service must be capable of routing IPv6 packets, as the useful lifetime of an IPv4-only service is limited. MNOs cannot obtain new IPv4 addresses to number the base stations, and using RFC 1918 space is not a scalable approach. Using IPv6-only to address mobility equipment at cell sites (and equivalent radio interfaces) is the preferred method for overcoming the scarcity of IPv4 addresses.

The shift from carrier Ethernet to IP backhaul should not be a monumental one for many carrier Ethernet providers. The heavy lifting of installing fiber and deploying a packet switched infrastructure has already been accomplished. In addition, carriers that implement carrier Ethernet with protocols like VPLS already have an infrastructure that is ready for IP. The most challenging aspect of the transition will be the work needed to prepare OAM&P systems for an IP service. Of course, this may vary based on carrier Ethernet implementation and systems.

Conclusion

Carrier Ethernet service for cell site backhaul is a vast scale and cost improvement over TDM backhaul and has been extremely successful. OSI Layer 3 IP networks have superior scaling properties that will replace Layer 2 backhaul networks of today. Advances in wireless networking systems, the proliferation of new devices, and the development of new mobility services will be best served with a truly IP-centric backhaul network.

References

[0] Santitoro, Ralph, "Metro Ethernet Services—A Technical Overview," 2003, **http://metroethernetforum.org/metro-ethernet-services.pdf**

[1] M. Grayson, K. Shatzkamer, and S. Wainner, *IP Design for Mobile Networks,* Cisco Press, 2009.

[2] *Distributed Mobility Management in Future Wireless Networks* (DiMoWiNe), **http://conference.researchbib.com/print.php?category=event&id=10232&uid=6**

JEFF LOUGHRIDGE is the principal consultant and owner of Brooks Consulting LLC, a firm that specializes in Tier 1 ISP best practices and the design, engineering, and operations of large-scale wireline and wireless IP/MPLS networks. Prior to founding Brooks Consulting, Jeff spent over ten years supporting Sprint's global IP network in both technical and managerial capacities. He earned a bachelor's degree in computer science from Duke University and an MBA from the University of Phoenix—Northern Virginia campus.
E-mail: **jeffl@brooksconsulting-llc.com**

Fragments

Global INET 2012

To help mark its 20-year-anniversary, the *Internet Society* (ISOC) is hosting a global forum that will bring together visionaries and thought leaders from around the world to focus on issues that will impact the future of the Internet.

The *Global INET 2012,* which is scheduled to take place in Geneva, Switzerland from April 22–24, will feature high-powered speakers, thought-provoking panel discussions, and interactive workshops to develop a vision for the explosive growth of the Internet over the next 20 years.

Thought leaders from across the Internet community will collaborate on topics critical to the global Internet's future, including privacy, net neutrality, IPv6, security, digital content and innovation, and human rights and freedom of expression.

Since its beginnings in 1992, ISOC has been dedicated to helping keep the Internet open, accessible, and defined by users—regardless of where they live, what they do, their abilities, or who they are.

Registration for Global INET 2012 is scheduled to begin in October 2011.

For more information:

[1] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, Stephen Wolff, "A Brief History of the Internet," December 2003, also published in ACM's *Computer Communication Review,* Volume 39, Number 5, October 2009.
http://www.isoc.org/internet/history/brief.shtml
http://www.sigcomm.org/ccr/papers/2009/October/1629607.1629613

[2] "The Internet Society's Principles and Goals,"
http://www.isoc.org/isoc/mission/principles/

[3] **http://www.isoc.org/isoc/conferences/inet/12/gva.shtml**

IPv6 Week

IPv6 Week will be a coordinated test of the new Internet Protocol, held February 6–12, 2012. Websites, content providers, Internet Services Providers, Network Service Providers, as well as end users are invited to participate. This is a Brazilian initiative, but anyone can participate.

For more information visit: **http://www.ipv6.br/IPV6/WeekIPv6**

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, troubleshooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ contains standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at `ole@cisco.com`

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com