The Internet Protocol Journal

December 2002 Volume 5, Number 4

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**

From The Editor

In December 1999 we published Part One of a two-part article on Internet Multicast. Some readers have asked "what happened to Part Two?" Finally, in this issue we are able to bring you the second article, "Internet Multicast Tomorrow." Multicast remains a technology with limited Internet-wide deployment, but numerous research activities are underway that may change this situation. Ian Brown, Jon Crowcroft, Mark Handley and Brad Cain provide an overview of current developments in multicast.

If all computer networking was a simple matter of "plug-and-play," I suppose this journal would not exist. Nevertheless, it is encouraging to see developments that aim to simplify configuration of network devices, particularly those that move around a lot. The Zeroconf working group of the *Internet Engineering Task Force* (IETF) has been developing standards for "configuration-free" networks. Edgar Danielyan explains the details in our second article.

We continue to receive numerous letters in response to our articles. Your feedback is very much appreciated, because it helps us develop material for future issues. Please keep your letters coming to **ipj@cisco.com**

The long-awaited online subscription system is now ready for deployment and you will be able to try it out in the very near future at **www.cisco.com/ipj**. With this system, you can update your mailing address as well as select delivery options, online notification of new issues and so on. As with any computer based system, I anticipate that we, with your help, will uncover a few bugs. Please report any problems you may encounter to **ipj@cisco.com**.

A new important resource is available from the *Internet Society* (ISOC). *The Internet Report* is a catalogue of IETF documents, including RFCs and Internet Drafts, that document the technology, protocols and operating procedures that form the Internet. The report includes RFCs, IETF Working Group drafts as well as individual drafts. The Internet Report is maintained by Geoff Huston. You can access the report online at **http://ietfreport.isoc.org/**

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

Internet Multicast Tomorrow

by Ian Brown, University College London,
Jon Crowcroft, University of Cambridge,
Mark Handley, ICIR,
Brad Cain, Storigen Systems

This article is part of a pair, the first of which looked at the state of play in IP multicast routing[0]. In this article, we look at the broader problems and future activities with multicast. We divide the areas into routing, addressing, transport, security, operations, and research.

There has been quite a bit of debate about the nature of compelling applications for multicast recently.[44] It is certainly the case that we do not completely understand the "market" for multicast—this is at least in part because multicast does not yet provide a complete set of functions for all the applications and services we might imagine. This is a typical "chicken and egg" situation, though: To put an extreme version of the argument, the application writers do not see any multicast deployed; the *Internet Service Providers* (ISPs) do not see any multicast applications; and the router vendors do not see any multicast service demand from ISPs. (The same problem afflicts IPv6, Integrated and possibly Differentiated Services, and mobile IP, of course.)

As we discussed in the part I of this article[0], this situation has been somewhat alleviated by streaming applications for audio and video from the classical content providers in the entertainment and news industries. And although we are still seeing some problems, we are also seeing broader interest and development.

The next section presents recent work on routing and addressing. After that we look at transport. Subsequently, we discuss security. Then we look at operations and management. Finally, we examine some of the research ideas that are available.

Routing and Addressing

The single biggest step recently in multicast routing and addressing has been the recognition that the demand for large-scale multicast is largely for one-to-many or single source. Combined with the ability to select sources at the receiver (as a means to prevent denial-of-service attacks) in the *Internet Group Management Protocol* (IGMP)v3, this has made a significant improvement to ISPs' willingness to deploy the service[42].

Source-Specific and Single-Source Multicast

The origins of the idea were thesis work at Stanford by Hugh Holbrook on Express multicast[43]. This is a specialized multicast architecture for one-to-many multicast groups. In this way, Express is a subset of the current multicast model in that it allows only a single sender to a multicast group. The advantages of Express are that certain aspects of multicast routing and addressing are easier solved by ignoring the many-to-many case. Many feel that the most likely large-scale applications of multicast are one-to-many, a fact that explains why Express is becoming popular as a short-term solution.

Express addresses are *channels* that are 64-bit addresses (that is, source address plus group address). Express sources transmit to a channel and advertise that channel. Receivers learn about these channels through advertisements or through other means (that is, URL) and initiate an Express join. Routers propagate these joins directly toward the source, building a source rooted multicast forwarding tree.

The Express model offers two primary benefits. First, Express simplifies the complexity of multicast routing. Secondly, Express simplifies the assignment of multicast addresses for IPv4. Because Express channels are 64 bits, a source can select any lower 32 bits (any group address) for its channel and not collide with another.

In order to implement Express with IPv4 multicast protocols, a special range of multicast addresses was defined. The 232/8 address has been allocated by the *Internet Assigned Numbers Authority* (IANA) for single-source multicast experimentation. In this range, an address has meaning only when "coupled" with a source address. Another way to explain it is that this address range is reserved for the lower 32-bit Express addresses. With this scheme, Express requires no modification to multicast data packets.

Express can be implemented with two protocols that have already been developed: IGMPv3[42] and *Protocol Independent Multicast Sparse Mode* (PIM-SM).

IGMPv3 extends IGMP to allow source-specific joins to a multicast address. This capability can be used to carry 64-bit (S,G) joins to a router. When a router receives the IGMPv3 join, it must be able to build the source-specific tree with a multicast routing protocol. PIM-SM, widely deployed in service provider networks, already possesses this capability. The combination of IGMPv3 and PIM-SM allows Express to be implemented without creating more protocols; this is one of the most powerful benefits of the Express model.

Interdomain Multicast

Currently there are four fairly widely deployed multicast routing protocols: *PIM Dense Mode* (PIM-DM), PIM-SM or *Source-Specific Multicast* (SSM), *Multicast OSPF* (MOSPF), and the *Distance Vector Multicast Routing Protocol* (DVMRP). Because of the different properties of these protocols, there are many difficulties in connecting heterogeneous routing domains together[38]. In general, most problems arise when connecting explicit join type protocols with flood-and-prune protocols. With service providers rolling out multicast using PIM-SM, connecting DVMRP and PIM-DM flood-and-prune is becoming common.

In order to connect two multicast routing domains, a *Multicast Border Router* (MBR) needs to exist between the two domains. This router must implement a shared forwarding cache architecture[39]. In this model, each multicast routing protocol running on a MBR submits its forwarding cache entries to a shared cache. This cache is the "bridge" between the trees in the different domains.

In order that the appropriate trees are created in each domain (on either side of a MBR), signaling must exist to bring sources from one domain to receivers in the other domain. This is part of the complication in connecting flood-and-prune protocol domains to explicit join protocol domains. In an explicit join protocol such as PIM-SM, joins are sent by edge routers to either a source or a *Rendezvous Point* when a host joins. A flood-and-prune protocol works quite differently, in a sense assuming that packets are desired; trees are pruned when edge routers receive new source packet but have no local listeners.

The signaling aspect of joining two domains can be accomplished with a variety of means. There are many options, but two stand out as providing the best methods of connecting domains. The first is to use *Domain Wide Reports* (DWRs)[36] in flood-and-prune domains. DWRs are similar to IGMP reports except that they are sent on a domain-wide basis. When a border router receives a DWR report, it can join a group on behalf of an entire domain. The second solution is to use the *Multicast Source Discovery Protocol* (MSDP)[37]. MSDP is currently used to send source lists between PIM-SM domains. It can also be used to connect domains by having the MBR also participate in MSDP. Sources can then be learned from an explicit join protocol domain; the MBR can then join the sources and flood them into attached flood-and-prune protocols domains.

Address Allocation

The schemes to provide dynamic distributed address allocation have not been successful to date. But with many multicast services being limited to either a single domain or a single source, the pressure is off. Instead, source-specific addresses are unique in any case. For many-to-many multicast (sometimes known as *Internet Standard Multicast* [ISM]), the problem has also been alleviated by the use of GLOP[61], which allocates sections of the address space by mapping Autonomous System numbers of a provider into Class D prefixes. This is potentially inefficient, but solves the contention, collision, revocation, or resolution problem that *Multicast Address Set Claim* (MASC) and *Multicast Address Allocation* (MALLOC)[60] attempt to do in a distributed dynamic manner.

In the longer term this address allocation, as well as scalable solutions to many-to-many multicast in the local domain and interdomain, await further development on bidirectional trees ["Bi-dir PIM" and the *Border Gateway Multicast Protocol* (BGMP)], which we discuss next. It is likely that these will need IPv6 to scale to serious usage.

Bidirectional PIM-SM

The PIM-SM multicast routing protocol builds both source and shared trees for the distribution of multicast packets. PIM-SM shared trees are rooted at special routers called Rendezvous Points and are unidirectional in nature. Shared tree traffic always flows from the Rendezvous Point down to the leaf routers. In some types of multicast applications, namely many-to-many type applications, a unidirectional tree may be inefficient.

Other multicast protocols such as *Core Based Trees* (CBT) and BGMP provide bidirectional shared trees. Bidirectional trees[40] do not have these inefficiencies in many-to-many applications. In a bidirectional tree, traffic from a source is forwarded directly onto the shared tree at the closest point; the traffic is then forwarded both "up" and "down" the tree to all receivers. This is in contrast to a unidirectional tree when the source packets are sent first to the Rendezvous Point (or root) and then down the tree. Recently, two proposals have been submitted that add bidirectional tree capabilities to PIM-SM[40].

BGMP

BGMP[33] is a new inter-domain multicast routing protocol that addresses many of the scaling problems of earlier protocols. BGMP attempts to bring together many of the ideas of previous protocols and adds features that make it more service provider friendly. BGMP is designed to be a unified inter-domain multicast protocol in much the same way that the *Border Gateway Protocol* (BGP) is used for unicast routing.

BGMP is an inter-domain protocol in that it adopts particular design features of BGP familiar to providers. Two of these features follow: it uses TCP connections for the transfer of routing information and it has a state machine (with error notifications) similar to BGP.

In order to accommodate different applications and backward compatibility, BGMP can build three types of multicast trees, both unidirectional source and shared trees and bidirectional shared trees. Unidirectional trees are useful for single-source applications and for backward compatibility with other multicast routing protocols. Shared trees are useful for many-to-many applications (for example, multiplayer gaming, videoconferencing) and multicast forwarding state to scale for these types of applications.

One of the unique properties of BGMP is that its shared trees are rooted at an Autonomous System that is associated with the multicast group address of the tree. Having the root of the tree at the Autonomous System that is associated with the address is logical because there are likely members in that domain. Rooting the trees at an Autonomous System level also provides stability and inherent fault tolerance.

BGMP requires a way to discover which Autonomous Systems "own" which multicast addresses; this can be accomplished through the use of the MASC protocol or through globally assignable multicast addresses (for example, IPv6 multicast). The MASC protocol allocates temporary assignments from the IPv4 group D address space; it then distributes these assignments into *Multiprotocol BGP* (MBGP) so that BGMP will know which Autonomous System is associated with which group and, therefore, where to send join messages.

If globally assignable addresses are available, then BGMP can use any static address architecture for obtaining an Autonomous System from a multicast group address.

The combination of BGMP and a large multicast address space (for example, IPv6 address space) provide the best scaling for all types of multicast applications.

Transport and Congestion Control: Calling Down Traffic on a Site

Multicast is a multiplier. It gives an advantage to senders, but without their knowledge. Multicast (and its application level cousin, the CU-SeeMe reflector) can "attract" more traffic to a site than it can cope with on its Internet access link. (CU-SeeMe is a popular Macintosh- and PC-based Internet videoconferencing package that currently does not directly use IP multicast.) A user can do this by inadvertently joining a group for which there is a high-bandwidth sender, and then "going for a cup of tea." This problem will be averted through access control, or through mechanisms such as charging[58], which may result from the deployment of real-time traffic support.

The problem is seen as critical by ISPs who have a shared bottleneck in their access technology—this is the case for cable modem and in some cases for *Asymmetric Digital Subscriber Line* (ADSL), where a large number of fast lines converge on a slower interface to the backbone. Here, a single user may attract more traffic than this link can handle, without seeing a problem that he or she causes for other users (unicast or other multicast lower-capacity separate sessions using the same shared bottleneck). The use of IGMPv3 with authenticated join and con-figuration management would appear to be a possible solution to these woes. Alternatively, the use of TCP-friendly multicast congestion control (as envisaged for reliable multicast, but also as emerging in some *Real-Time Transport Protocol* (RTP)[4] applications), would also solve this problem.

Congestion Control

One of the critical areas to clarify is the role of congestion control in multicast transport protocols[1]. From an early stage, it was established that coexistence with TCP was a critical design goal for protocols that would operate in the wider Internet. Thus systems such as *TCP Friendly (Reliable) Multicast Congestion Control* (TFMCC)[8], *Pragmatic General Multicast Congestion Control* (PGMCC)[53], and receiver-driven congestion control[54] all extend the classic work by Raj Jain[15] and Van Jacobson[17] and subsequent evolution[16] on TCP congestion avoidance and control.

Recently, this line of thinking has even been extended back into the unicast world in the application of such control schemes to *User Datagram Protocol* (UDP)-like flows in the work on the *Datagram Congestion Control Protocol* (DCCP)[62], suitable for adaptive multimedia flows on RTP, for example.

Reliable Multicast

There is a clear requirement for some sort of analog to TCP for multicast applications that need a level of reliability. The *Internet Research Task Force's* (IRTF's) *Reliable Multicast Research Group* (RMRG) group[3] has developed numerous prototypical solutions to the problem, which turns out to be quite a large design space (not "one size fits all").

The IETF *Reliable Multicast Transport* (RMT) working group has now been chartered to develop single-source reliable multicast transport solutions that meet the current Internet constraints[1]. That group has developed a building block approach[12], which is based partly on abstracting components from existing work such as *Reliable Multicast Transport Protocol* (RMTP) II[18], *Receiver Driven Layered Congestion Control* (RLC)[7], *Multicast File Transfer Protocol* (MFTP)[28], *Pragmatic General Multicast* (PGM)[41], and many other protocols.

Some applications of RMT products are likely to be infrastructural rather than of direct use to the ISPs' customers—for example, distributing software to mirror sites seems to be one popular compelling use.

However, reliable multicast is sometimes regarded as something of an oxymoron. When people talk about "Reliable Multicast," they usually mean a single protocol at a single "layer" of a protocol stack, typically the transport layer (although we have seen people propose it in the network and even link [ATM!] layers too), that can act as any layered protocol can—to provide common functionality for applications (higher layers) that need it.

So what is wrong with that? Well, possibly three things (or more):

- *Fate sharing:* Fate sharing in unicast applications means that as long as there is a path that IP can find between two applications, then TCP can hang on to the connection as long as the parties like. However, if either party fails, the connection certainly fails.

  Fate sharing between multicast end points is a more subtle idea. Should "reliability" extend to supporting the connection fork recipients failing? Clearly this will be application specific (just as timing out on not getting liveliness out of a unicast connection is for TCP—we must permit per-recipient timeouts and failures).

- *Performance:* When A talks to B, the performance is limited by one path. Whatever can be done to improve the throughput (or delay bound) is done by IP (for example, load sharing the traffic over multiple paths). When A talks to B, C, D, E, or F, should the throughput or delay be that sustainable by the slowest or average?

- *Semantics:* As well as performance and failure modes, N-way reliable protocols can have different service models. We could support reliable one-to-n, reliable n-to-one, and reliable n-to-m.

Applications such as software distribution are cited as classic one-to-n requirements. Telemetry is given as an n-to-one reliable protocol. Shared whiteboards are cited as examples of n-to-m applications.

It is interesting to look at the reliability functions needed in these. The one-to-n and n-to-one protocols are effectively *simplex* bulk transfer applications. In other words, the service is one where reliability can be dealt with by "rounding up" the missing bits at the end of the transfer. Because this does not need to be especially timely, there is no need for this to be other than end to end, and application based. (Yes, we know telemetry could be time sensitive, but we are trying to illustrate major differences clearly for now.)

On the other hand, n-to-m processes such as whiteboards need timely recovery from outages. The implication is that the "service" is best done somewhat like the effect of having $n \times (m-1)/2$ TCP connections. If used in the WAN, the recovery may best be distributed, because requests for recovery will implode down the very links that are congested or error prone and cause the need for recovery.

Now there are different schemes for creating distributed recovery. If the application semantics are that operations (application data unit packets worth) are sequenced in a way that the application can index them, then any member of a multicast session can efficiently help any other member to recover (examples of this include Mark Handley's Network Text tool[16].) On the other hand, packet-based recovery can be done from data within the queues between network or transport and application, if they are kept at all members in much the same way as a sender in a unicast connection keeps a copy of all unacknowledged data.

The problem with this is that *because* it is multicast, we do not have a positive acknowledgement system. Therefore, there is no way to inform *all* end points when they can safely discard the data in the "retransmit" queue. Only the application really knows this!

Well, this is not to say that there is not an obvious toolkit for reliable multicast support—it would certainly be good to have RTP-style media timestamps (determined by the application, but filled in by the system). It would be good to have easy access to a timestamp-based receive queue so applications could use this to do all functions discussed previously. It might be advantageous to have virtual Token Ring, expanding ring search, token tree, and other toolkits to support retransmit "helper" selection.

Table 1 illustrates this in terms of where functions might be put to provide reliability (retransmit), sequencing, and performance (adaptive playout, say, versus end to end, versus hop-by-hop delay constraint).

Table 1: Reliable Multicast Semantics

	Recovery	Sequency	Dalliance
Network	not in our internet	ditto	int-serv
Transport	one–many	yes	adaptive
Application	many–many	operation semantics	adaptive

Router Assist for Reliable Multicast

As mentioned in previous sections, one of the difficulties in end-to-end multicast signaling is the "implosion" of signaling at a source from many receivers. This problem has been addressed in numerous ways, including the use of timers, the use of servers to aggregate signaling, and the use of router-assisted mechanisms. We now discuss three protocols that make use of router assistance in order to better scale end-to-end multicast protocols.

PGM[41] is a *negative acknowledgement* (NAK)-based router-assisted reliable multicast protocol. PGM uses routers to aggregate receiver-to-source signals (for example, the NAKs) as they flow toward the source. PGM router support also includes a subcasting ability whereby repairs will flow down only to receivers who have requested them.

Extending the ideas of router assist in PGM is the *Generic Multicast Transport Service* (GMTS). GMTS provides generic, fixed, simple services for any end-to-end multicast transport protocol. These services include such features as signal aggregation with predicates and sophisticated subcasting ability. GMTS was used as a basis for *Generic Router Assist* (GRA)[34], which is similar, IETF standards oriented, and a bit more streamlined.

Securing Multicast

Multicast security is more difficult than unicast security in several areas. The key exchange protocols used between unicast hosts do not scale to groups. Rekeying is required more often to maintain confidentiality as group membership changes. And the efficient authentication transforms used between two unicast hosts cannot protect traffic between mutually distrustful members of a group.

These problems are being worked on by the IETF *Multicast Security* (msec) and IRTF *Group Security* (gsec) working groups. Because of the wide range of application requirements in group communication, their work is based upon a building block approach similar to that of the RMT group.

The blocks being developed are data security transforms, group key management and group security association, and group policy management[49]. An application may use different blocks together to create a protocol that meets its specific requirements.

Data Security Transforms

A data security transforms block provides confidentiality and authentication services for data being transported between group members. Confidentiality is reasonably easy to provide using standard encryption algorithms. Authentication is more difficult, because the algorithms used in unicast protocols such as *IP Security* (IPSec) would not allow a group member to authenticate data as being from another specific group member. This is because the secret used to authenticate the traffic must be shared between all sending and receiving parties. Public-key signatures would solve this problem, but are an order of magnitude slower than symmetric authentication algorithms and hence especially unsuitable for real-time traffic and low-powered communications devices.

Instead, blocks such as the *Timed Efficient Stream Loss-tolerant Authentication Protocol* (TESLA)[55] are being developed that trade off small amounts of functionality (such as immediate rather than slightly delayed authentication) to retain the efficiency benefits of symmetric algorithms. TESLA senders use a hash chain of keys $k_{n...1}$ to sign data, where: $k_n = hash(k_{n-1})$

They release each key in the chain a short interval after the data the key has signed. As long as other group members received the data during that interval, they can be confident that the signature was made by the sender. If keys are lost during transmission, receivers can recompute any key earlier in the sequence simply by repeatedly applying the hash function used to any later key received. Finally, they can be sure that keys are coming from the sender because the first key in the sequence is digitally signed, while only the sender can know the later keys in the sequence (because by definition, a hash function must not be reversible).

Group Key Management and Group Security Association

To use data security transforms, group members need to possess the cryptographic keys necessary to encrypt or decrypt and sign or authenticate data. They also need to agree on parameters such as specific encryption algorithms. This building block allows this information to be shared between group members.

The Group Key Management architecture[47] provides a unified model for key management blocks. A central *Group Controller/Key Server* (GCKS) provides *Traffic Encrypting Keys* (TEKs) or *Key Encrypting Keys* (KEKs) to new group members after authenticating them with a unicast protocol. The GCKS may also delegate some of its functions to other entities, improving scalability.

In groups with simple security requirements, this may be the only communication required between a group member and GCKS. But if group changes need to be cryptographically enforced, further TEKs, encrypted using a KEK, may be provided to members by multicast or a more scalable protocol such as the *Logical Hierarchy of Keys* (LHK)[56] that does not require every rekey message to be sent to every group member. Alternatively, noninteractive mechanisms such as hash trees may be used to update keys[48]. Finally, group members may explicitly de-register with the GCKS using a one- or two-step message.

Three key management building blocks are being developed. The *Group Domain of Interpretation* (GDOI) builds on the *Internet Security Association Key Management Protocol* (ISAKMP)[52] to allow the creation and management of security associations for IPSec and other network or application layer protocols[46]. *Multimedia Internet Keying* (MIKEY) is targeted at real-time multimedia communications, particularly those using the Secure RTP, and can be tunneled over the *Session Initiation Protocol* (SIP)[45]. And a *Group Secure Association Key Management Protocol* (GSAKMP), along with a GSAKMP-Light profile, have also been developed[51].

Group Policy Management

The final building block defines policies such as which roles various entities may play in the group; who may hold group information such as cryptographic keys; the cryptographic algorithms used to protect group data; and proof that the creator of a given policy is authorized to do so. A group policy token is used to hold all of this information[50]. All or part of tokens can be made available to users in policy repositories or by using other out-of-band mechanisms.

Operational Deployment of Multicast

As mentioned previously, multicast seems to be difficult to deploy. One problem is that it has only recently moved from the research community (and typically implemented using tunnels) into the service community (running native IP multicast routing).

This means that debugging multicast sessions, applications, and routing is a common activity. However, because of the dynamic nature of multicast addresses and the anonymous nature of the multicast service model, debugging is somewhat more difficult than for the equivalent unicast case.

Fortunately, all current native multicast paths are at least computed from underlying unicast ones, and it is possible to use tools such as *mtrace* and *mrm* to query the underlying router system to try to figure out where things are going on. Of course, the relevant *Management Information Bases* (MIBs) need to be designed, but mere *Simple Network Management Protocol* (SNMP) access to the variables defined in these may not be enough.

Many multicast sessions are global, and not surprisingly, someone, somewhere, sometime in the session will have a problem. In a way, you only have to look at multicast as a way of sampling large pieces of the Internet at one time to see why it is difficult to understand. In fact, a research project called *Multicast-Based Inference of Network-Internal Characteristics* (MINC)[9, 57] is using that very observation to build tools of more general use.

MRM

One recent tool that has been developed to facilitate multicast monitoring and debugging is the *Multicast Reachability Monitor* (MRM)[32]. MRM consists of two parts; a MRM management station configures test senders and test receivers in multicast networks. A multicast test sender or test receiver is any server or router that supports the MRM protocol and can source or sink multicast traffic. MRM provides the ability to dynamically test particular multicast scenarios; this capability can be used for fault isolation and general monitoring of sessions.

MRM is typically used to configure MRM-capable routers as test senders and test receivers from a management station. Routers configured as test senders send multicast packets periodically to a configured multicast group at a configured rate. Routers configured as test receivers monitor traffic to a group and keep statistics that can be reported back via *RTP Control Protocol* (RTCP) packets. Test receivers can be configured to send RTCP reports when a given condition has been reached or when polled by a management station. Although the MRM protocol is simple itself, it provides powerful capabilities that can be used by future multicast debugging applications.

Research Ideas in Multicast Routing and Addressing

The seeming complexity exhibited by the full panoply of multicast protocols has led some people to develop doubts as to the eventual deployment of multicast. It is far too early to say whether these doubts are well founded. The slow pace of deployment is a symptom not just of this complexity, but also of the underlying complexity of handling growth and evolution of *any* type in such a large system as the Global Internet.

Having said that, it is worth mentioning four of the approaches that have been discussed in the Internet community recently:

- *Addressable Internet Multicast* (AIM), by Brian Levine, et al., attempts to provide explicit addressing of the multicast tree. The routers run a tree-walking algorithm to label all the branch points uniquely, and then make these labels available to end systems. This allows numerous interesting services or refinement of multicast services to be built. Of some particular interest would be the ability this service gives to end systems to do subcasting, which would be useful for some classes of reliable transport protocols.

- *Explicitly Requested Single-Source* (Express), by Hugh Holbrook et al., is aimed at optimizing multicast for a single source. The proposal includes additional features such as authentication and counting of receivers, which could be added to many other multicast protocols usefully. It is motivated by a perceived requirement from some ISPs for these additional features. Express makes use of an extended address (channel + group) to provide routing without global agreement on address assignment. A possible source of problem for AIM is the potential for unbounded growth in the size of identifiers for labeling subtree branch points.
- *Root Addressed Multicast Architecture* (RAMA), by Radia Perlman et al., is in some senses a generalization of Express type addressing, but it also requires bidirectional trees (CBT like, rather than current PIM-SM, although work on bidirectional PIM is under way too). The goal is to offer a single routing protocol for both intra- and interdomain. In fact, RAMA can be implemented by combining the address extensions proposed for Express, and two-level bidirectional PIM as an implementation of BGMP. RAMA and Express (and bidirectional PIM) require a mechanism for carrying additional information in multicast IP data packets.

  There are two critical problems for carrying this identifier that are difficult to solve in general: first, it takes new space in the IP packet, and this has to be accessed by both hosts and routers—that represents a deployment problem; secondly, in the general case, the extra field must be examined on the "fast path," in routers that have such a concept, and this takes valuable processing resources that may have to be taken away from some other forwarding task.
- *Connectionless Multicast* (CM) by Dirk Ooms, et al., is a proposal for small, very sparse groups to be implemented by carrying lists of IP unicast addresses in packets. The scheme is not simply a form of loose source routing, because it would make use of packet replication at appropriate branch points in the network. It may be well suited to IP telephony applications where a user starts with a unicast call, but then adds a third or fourth participant.
- The *L'Ecole Polytechnique Fédérale de Lausanne* (EPFL) work on *Distributed Core Multicast* (DCM) aims to address very large numbers of very small groups with mobile users, typical characteristics of mobile IP telephony users making conference or group calls.
- MIT has done some work on the use of wide-area "anycast" addresses for the core and Rendezvous Point. This results in a potential improvement in the availability of trees (and subtrees) for multicast delivery in the event of router or link outage. More importantly, it may be possible for a multicast group to survive network partitions (or lack of core reachability), a possibility that would make this an invaluable improvement to the service. It depends on the scalability of the wide-area anycast solution, which the MIT work shows is at least viable, and certainly worth more attention.

- *Yet Another Multicast* (YAM) routing protocol[30] was devised by Ken Carlberg of SAIC to address the possibility of forming different multicast trees based on some QoS metric—the idea is that IGMP is modified to provide a "one-to-many" join, and a receiver sends this with required performance parameters. Routers receiving the request over links that can provide this service respond. The receiver (sender of the one-to-many IGMP) selects the one to then commit the join to.
- *Quality of Service Sensitive Multicast Internet protoCol* (QoSMIC) is a development from YAM by Faloutsos[29] at Toronto, and slightly modifies the tree-building exercise.
- When multicast and *Multiprotocol Label Switching* (MPLS) are mentioned together, there is both confusion and surprise. MPLS can be used with multicast in two very different ways. The first method is by building multicast trees over MPLS traffic-engineered paths. Some multicast routing protocols already make use of unicast forwarding information for the construction of multicast trees. Using multicast traffic-engineered paths is simply an extension of this concept—with one caveat. Some multicast routing protocols use *Reverse Path Forwarding* (RPF) checks on incoming packets to prevent looping; this is accomplished by checking to see if the incoming interface is the "closest" to the source. With MPLS traffic engineering, RPF checks are difficult. A solution has not been presented at this time that addresses this problem.

  The second method for using multicast with MPLS is through the use of point-to-multipoint virtual circuits in much the same way as ATM point-to-multipoint virtual circuits. These are useful in cases where receivers are statically configured to a multicast address or multicast traffic is always to be delivered to a destination. Mapping dynamic memberships into a multipoint circuit has proven difficult, for example, with ATM. There are currently several Internet drafts that propose various solutions for MPLS and multicast[31].
- Several groups have been working on end system-only multicast schemes, probably most notably Carnegie-Mellon University[59].

Summary and Conclusions

In this article, we have looked at some of the newer ideas in the research and development community in the area of multicast. There is still a lot to be done to close the loop between network services, transport, and applications, but present research indicates that we will eventually achieve this goal.

References

[0] M. Handley and J. Crowcroft, "Internet Multicast Today," *The Internet Protocol Journal,* Vol. 2, No. 4, December 1999.

[1] A. Mankin, A. Romanow, S. Bradner, and V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols," RFC 2357, June 1998.

[2] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," Proceedings of SIGCOMM '98, September 1998.

[3] Reliable Multicast Research Group: **http://www.east.isi.edu/RMRG/**

[4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, January 1996.

[5] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, Scalable Reliable Multicast (SRM)," Proceedings of ACM SIGCOMM '95.

[6] M. Handley and J. Crowcroft, "Network Text Editor (NTE): A scalable shared text editor for the Mbone," Proceedings of ACM SIGCOMM '97, September 1997.

[7] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer," Proceedings of INFOCOM '98.

[8] M. Handley, S. Floyd, and B. Whetten, "Strawman specification for TCP friendly (reliable) multicast congestion control (TFMCC)," work in progress.

[9] S. R. Caceres, N. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation," Proceedings of IEEE Infocom '99, March 1999.

[10] S. J. Cowley, "Of Timing, Turn-taking, and Conversations," *Journal of Psycholinguistic Research,* 1998, Vol. 27, No. 5, pp. 541–571.

[11] Jonathan Rosenberg and Henning Schulzrinne, "Timer Reconsideration for Enhanced RTP Scalability," Proceedings of the Conference on Computer Communications (IEEE Infocom), March/April 1998.

[12] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer," RFC 3048, January 2001.

[13] Handley, M. et al., "Rate Adjustment Protocol," Proceedings of Infocom 1999.

[14] Kouvelas, I. et al., "Self Organising Transcoders," Proceedings of NOSSDAV 1998.

[15] D-M. Chiu and R. Jain, "Analysis of the Increase and Decrease Algorithm hms for Congestion Avoidance," *Computer Networks and ISDN Systems,* Vol. 17, pp. 1–14, 1989.

[16] S. Floyd and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control," Technical report,
`ftp://ftp.ee.lbl.gov/papers/collapse.ps`

[17] V. Jacobson, "Congestion Avoidance and Control," Proceedings of ACM SIGCOMM '88, August 1988, pp. 314–329.

[18] J. C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," Proceedings of IEEE INFOCOM '96, March 1996, pp. 1414–1424.

[19] M. Mathis, J. Semke, J. Mahdavi, and T. Ott,"The Macroscopic Behaviour of the TCP Congestion Avoidance Algorithm," ACM *Computer Communication Review,* Vol. 27 No. 3, July 1997.

[20] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," Proceedings of SIGCOMM '96, August 1996, pp. 1–14.

[21] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modelling TCP Throughput: A Simple Model and Its Empirical Validation," Proceedings of SIGCOMM '98, September 1998.

[22] L. Rizzo and L. Vicisano, "A Reliable Multicast Data Distribution Protocol Based on Software FEC Techniques," The Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS '97), June 1997.

[23] Dan Rubenstein, Jim Kurose, and Don Towsley, "The Impact of Multicast Layering on Network Fairness," Proceedings of ACM SIGCOMM '99, August 1999.

[24] N. Shacham, "Multipoint Communication by Hierarchically Encoded Data," Proceedings of IEEE Infocom '92, 1992, pp. 2107–2114.

[25] Chris Greenhalgh, Steve Benford, Adrian Bullock, Nico Kuijpers, and Kurt Donkers, "Predicting Network Traffic for Collaborative Virtual Environments," *Computer Networks and ISDN Systems,* Vol. 30, 1998, pp. 1677–1685.

[26] Steve Deering, "Host Extensions for IP Multicasting," RFC 1112, August 1989.

[27] S. Deering, C. Partridge, and D. Waitzman, "Distance Vector Multicast Routing Protocol," RFC 1075, November 1988.

[28] Ken Miller, "Multicast File Transfer Protocol," White Paper, Starburst Technologies.

[29] Michalis Faloutsos, Anindo Banerjea, and Rajesh Pankaj, "QoSMIC: Quality of Service Sensitive Multicast Internet ProtoCol," ACM *Computer Communication Review,* Vol. 28, pp. 144–153, September 1998.

[30] K. Carlberg and J. Crowcroft, "Building Shared Trees Using a One-To-Many Joining Mechanism," ACM *Computer Communication Review,* Vol. 27, pp. 5–11, January 1997.

[31] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, and F. Ansari, "Framework for IP Multicast in MPLS," work in progress.

[32] K. Almeroth, K. Sarac, and L. Wei, "Supporting Multicast Management Using the Multicast Reachability Monitor (MRM) Protocol," UCSB CS Technical Report, May 2000.

[33] D. Thaler, D. Estrin, D. Meyer, et al., "Border Gateway Multicast Protocol (BGMP)," Proceedings of ACM SIGCOMM '98, 1998.

[34] B. Cain, T. Speakman, and D. Towsley, "Generic Router Assist Building Block," work in progress.

[35] B. Cain and D. Towsley, "Generic Multicast Transport Services (GMTS)," Proceedings of Networking 2000, Paris, France, May 2000.

[36] B. Fenner, "Domain Wide Multicast Group Membership Reports," work in progress.

[37] D. Farinacci et al., "Multicast Source Discovery Protocol," Internet Draft, January 2000, work in progress.

[38] B. Cain, "Connecting Multicast Domains," Internet Draft, work in progress, October 1999.

[39] D. Thaler, "Interoperability Rules for Multicast Routing Protocols," RFC 2715, October 1999.

[40] D. Estrin and D. Farrinacci, "Bi-directional Shared Trees in PIM-SM," work in progress.

[41] T. Speakman et al., "PGM Reliable Transport Protocol Specification," RFC 3208, December 2001.

[42] B. Cain, S. Deering, and A. Thyagarajan, "Internet Group Key Management Protocol, Version 3," work in progress.

[43] H. Holbrook and D. Cheriton, "IP Multicast Channels: Express Support for Large-scale Single-source Applications," Proceedings of SIGCOMM '99, September 1999.

[44] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," IEEE Network Magazine, Special Issue on Multicasting, January/February 2000.

[45] J. Arkko, E. Carrera, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: Multimedia Internet KEYing," Internet Draft, work in progress, February 2002.

[46] M. Baugher, T. Hardjano, H. Harney, and B. Weis, "The Group Domain of Interpretation," Internet Draft, work in progress, February 2002.

[47] M. Baugher, R. Canetti, L. Dondeti, and F. Lindholm, "Group Key Management Architecture," Internet Draft, work in progress, February 2002.

[48] B. Briscoe, "MARKS: Zero Side Effect Multicast Key Management Using Arbitrarily Revealed Key Sequences," Proceedings of Networked Group Communication, November 1999.

[49] T. Hardjano, R. Canetti, M. Baugher, and P. Dinsmore, "Secure IP Multicast: Problem Areas, Framework, and Building Blocks," Internet Draft, work in progress, September 2000.

[50] T. Hardjano, H. Harney, P. McDaniel, A. Colgrove, and P. Dilmore, "Group Security Policy Token," Internet Draft, work in progress, November 2001.

[51] H. Harney, A. Schuett, and A. Colegrove, "GSAKMP Light," Internet Draft, work in progress, July 2001.

[52] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, November 1998.

[53] Luigi Rizzo, "pgmcc: A TCP-friendly Single-Rate Multicast Congestion Control Scheme," Proceedings of ACM SIGCOMM '2000, August 2000.

[54] Luby et al., "Wave and Equation Based Rate Control Using Multicast Round Trip Time," Proceedings of ACM SIGCOMM '2002, September 2002.

[55] A. Perrig, R. Canetti, B. Briscoe, D. Tygar, and D. Song, "TESLA: Multicast Source Authentication Transform," Internet Draft, work in progress, November 2000.

[56] D. M. Wallner, E. Harder, and R. C. Agee, "Key Management for Multicast: Issues and Architectures," RFC 2627, September 1998.

[57] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley, "Multicast-Based Inference of Network-Internal Delay Distributions," **http://www.cs.umass.edu/pub/Lopr99TR9955.ps.Z**

[58] T. Henderson and S. Bhatti, "Protocol Independant Multicast Pricing," Proceedings of NOSSDAV 2001.

[59] Yang-hua Chu, Sanjay G. Rao, and Hui Zhang, "A Case for End System Multicast," Proceedings of ACM SIGMETRICS, June 2000, pp. 1–12.

[60] Multicast Address Allocation Working Group, **http://www.icir.org/malloc/**

[61] D. Meyer and P. Lothberg, "GLOP Addressing in 233/8," RFC 3180, September 2001.

[62] **http://www.icir.org/dccp/**

IAN BROWN holds a BSc from The University of Newcastle upon Tyne and a PhD from University College London. His research has focused on network security and active networking. He is a member of the ACM, IEEE, and is a contributor to the Internet Engineering Task Force, particularly in the area of authorized emergency communications. He has also worked extensively on the social implications of technology, and is a trustee of Privacy International and advisory board member of the Foundation for Information Policy Research. His e-mail address is:
I.Brown@cs.ucl.ac.uk

BRAD CAIN is a Senior Consulting Engineer at Storigen Systems, where he contributes to product development in the areas of networking and storage technology. Prior to joining Storigen, Cain was chief scientist at Cereva Networks, where he worked on system architecture and new product development. Cain also worked at Mirror Image Internet, one of the first commercial Content Delivery Networks (CDNs), where he helped architect their content distribution system. Cain is a contributor in the IETF and IRTF in the areas of IP multicast, IP routing, MPLS, and content networking. He has published numerous papers in the areas of routing and multicast and has more than 40 patents pending in the areas of multicast, security, routing, and router architecture. Cain holds a masters and bachelors in electrical engineering from the University of Delaware.
E-mail: **Brad.Cain@storigen.com**

JON CROWCROFT is the Marconi Professor of Networked Systems at the University of Cambridge. Prior to that he was professor of networked systems at University College London (UCL) in the Computer Science Department. He is a member of the ACM, a Fellow of the British Computer Society, a Fellow of the IEE, and a Fellow of the Royal Academy of Engineering, as well as a senior member of the IEEE. He is a member of the IAB, and was general chair for ACM SIGCOMM from 1995 to 1999. He is on the editorial team for the ACM/IEEE *Transactions on Networks and Computer Communications,* as well as on the program committee for ACM SIGCOMM and IEEE Infocomm. He has published five books—the latest is *Linux TCP/IP Implementation,* published by Wiley in 2001.
E-mail: **Jon.Crowcroft@cl.cam.ac.uk**

MARK HANDLEY received his BSc in Computer Science with Electronic Engineering from University College London in 1988 and his PhD from UCL in 1997. For his PhD he studied multicast-based multimedia conferencing systems, and was technical director of the European Union funded "MICE" and "MERCI" multimedia conferencing projects. After two years working for the University of Southern California's Information Sciences Institute (ISI), he moved to Berkeley to join the new ICSI Center for Internet Research (formerly known as ACIRI). Most of his work is in the areas of scalable multimedia conferencing systems, reliable multicast protocols, multicast routing and address allocation, and network simulation and visualization. He is co-chair of the IRTF Reliable Multicast Research Group, and he previously chaired the IETF Multiparty Multimedia Session Control working group. E-mail: **mjh@icir.org**

Zero Configuration Networking

by Edgar Danielyan, Danielyan Consulting

Zero configuration networking may sound like an oxymoron to many who spend most of their time setting up and mending networks. But don't decide on a career change yet—although zero configuration networks exist and work, they don't work always and everywhere. In this article I describe the current state of the affairs in zero configuration IP networking, introduce Zeroconf, the suite of zero configuration IP protocols, and tell what they do and how they work. This article is only a brief introduction to zero configuration networking and Zeroconf, so if you are really interested in all the details, refer to the sources listed in the References section at the end of this article.

The best introduction to Zeroconf is the one from the *Zeroconf Working Group* of the *Internet Engineering Task Force* (IETF)[1]:

> "The goal of the Zero Configuration Networking (Zeroconf) is to enable networking in the absence of configuration and administration. Zero configuration networking is required for environments where administration is impractical or impossible, such as in the home or small office, embedded systems 'plugged together' as in an automobile, or to allow impromptu networks as between the devices of strangers on a train."

Essentially, to reduce network configuration to zero (or near zero) in *Internet Protocol* (IP) networks, it is necessary, inter alia, to:

- Distribute IP addresses (without a *Dynamic Host Configuration Protocol* [DHCP] server),
- Provide name resolution (without a *Domain Name System* [DNS] server),
- Find and list services (without a directory service), and
- Distribute multicast IP addresses, if necessary (without a multicast server).

These and other requirements are defined in an Internet Draft titled "Requirements for Automatic Configuration of IP Hosts" by Aidan Williams[2]. This document does not define Zeroconf protocols themselves but instead spells out the requirements that should be met to achieve effective and useful zero configuration IP networking. One of the most important requirements for any Zeroconf protocol is that it should not interfere with other protocols and it must be able to exist on the same network with other non-Zeroconf protocols and devices. Another requirement is "no less" security—Zeroconf protocols should not be less secure than existing non-Zeroconf protocols—more on this later. Although IPv6 addresses some of the requirements of zero configuration networking (such as automatic allocation of link-local addresses), other requirements have yet to be met for both IPv4 and IPv6.

Zeroconf IETF Working Group

The Zeroconf Working Group of the IETF is chaired by Erik Guttman of Sun Microsystems and Stuart Cheshire from Apple Computer, with Thomas Narten (IBM) and Erik Nordmark (Sun) serving as area directors. It was chartered in September 1999 and had its first meeting at the 46th IETF in Washington, D.C., in November 1999. Those interested in the work of Zeroconf WG may find the mailing list archive of the working group at:
`http://www.merit.edu/mail.archives/zeroconf/`

Where and When to Use Zeroconf

For a correct understanding of the applicability and usefulness of Zeroconf it is necessary to keep in mind that it is a *link-local* technology. Link-local addressing and naming are meaningful only in a particular network; link-local addresses and names are not global and are not unique globally. In this case it means that Zeroconf is intended for use in small wired or wireless local-area networks in situations and places where zero configuration is necessary. It is appropriate to use Zeroconf in such networks when there is no possibility (or it is inappropriate) to set up a working IP network using the traditional technologies such as DNS and DHCP. Zeroconf is not appropriate and should not be used in many cases, for example in:

- Medium or large networks
- Networks where a high degree of security and control is required
- Large public access networks
- Networks with low bandwidth and high latency (such as some wireless networks)

When inappropriately used, Zeroconf may bring more problems and headaches than it solves. In contrast, examples of correct and appropriate use would include:

- Home and small office networks
- Ad hoc networks at meetings and conferences (especially wireless networks)
- Two devices needing to spontaneously share or exchange information

Likewise, Zeroconf advantages from one viewpoint may become annoying problems from another. Consider, for instance, the automatic distribution and configuration of link-local IP addresses. For a home network user this is a blessing—no longer do you have to spend time creating an addressing scheme and setting the IP addresses and netmasks on devices that should just work. But for an enterprise network (especially an incorrectly configured one), sudden appearance of nodes with (yet) unfamiliar and strange (this is not your regular `10.*` or `192.168.*`) IP addresses may result in more than surprise and added workload for the network administrator.

Continuing in this manner, Multicast DNS (mDNS) that ends the misery of having to remember and type **`ftp 10.20.30.1`** every time you need to transfer files from or to your PC named Bobo and replaces it with just **`ftp bobo`** may result in strange behavior on some networks. The bottom line? Zeroconf is not a one-size-fits-all solution; it wasn't designed to be one, and will not work as one.

Zeroconf and Security

Security should occupy an important place in the minds of all networking professionals, so an introduction to zero configuration networking would be incomplete without a mention of its security position. Security goals of Zeroconf are defined in section 4, Security Considerations, of "Requirements for Automatic Configuration of IP Hosts"[2]:

> "Zeroconf protocols are intended to operate in a local scope, in networks containing one or more IP subnets, and potentially in parallel with standard configured network protocols. Application protocols running on networks employing zeroconf protocols will be subject to the same sets of security issues identified for standard configured networks. Examples are: denial of service due to the unauthenticated nature of IPv4 ARP and lack of confidentiality unless IPSec-ESP, TLS, or similar is used. However, networks employing zeroconf protocols do have different security characteristics, and the subsequent sections attempt to draw out some of the implications.
>
> Security schemes usually rely on some sort of configuration. Security mechanisms for zeroconf network protocols should be designed in keeping with the spirit of zeroconf, thus making it easy for the user to exchange keys, set policy, etc. It is preferable that a single security mechanism be employed that will allow simple configuration of all the various security parameters that may be required. Generally speaking, security mechanisms in IETF protocols are mandatory to implement. A particular implementation might permit a network administrator to turn off a particular security mechanism operationally. However, implementations should be "secure out of the box" and have a safe default configuration.
>
> Zeroconf protocols MUST NOT be any less secure than related current IETF-Standard protocols. This consideration overrides the goal of allowing systems to obtain configuration automatically. Security threats to be considered iclude both active attacks (e.g. denial of service) and passive attacks (e.g. eavesdropping). Protocols that require confidentiality and/or integrity should include integrated confidentiality and/or integrity mechanisms or should specify the use of existing standards-track security mechanisms (e.g. TLS (RFC 2246), ESP (RFC 1827), AH (RFC 2402) appropriate to the threat."

Although this document does not address each and every aspect of security issues with Zeroconf, it sets requirements for Zeroconf protocols. As is the case with traditional IPv4 and IPv6, use of such techniques as *IP Security Architecture* (IPSec) or *Transport Layer Security* (TLS) may be appropriate in some cases. However, the nonstatic (or one may say nondurable) nature of both IP addresses and names in Zeroconf environment may pose a problem for IPSec and TLS deployment.

Dynamic Configuration of IPv4 Link-Local Addresses

Generally speaking, the first requirement that should be fulfilled before any useful IP communication can occur are the IP addresses of sender and recipient. The IP addresses are usually either assigned and set manually or provided by some other means such as DHCP or the *Point-to-Point Protocol* (PPP). However, neither of these is possible in zero configuration networks. Therefore, an automatic mechanism for dynamic configuration of IP addresses without any manual intervention or dependence on third-party service (that is, DHCP) is necessary. This mechanism already exists in IPv6 but not in IPv4. In "Dynamic Configuration of IPv4 Link-Local Addresses"[3], Stuart Cheshire, Bernard Aboba, and Erik Guttman describe a method that may be used in IPv4 networks to automatically assign IPv4 addresses valid for local communication on a particular interface. A special network **`169.254/16`** is reserved with the *Internet Assigned Numbers Authority* (IANA) for this purpose. It is necessary to highlight that **`169.254/16`** addresses are reserved for link-local use only. The document also addresses such issues as support for multiple addresses and multiple interfaces, continuous address conflict detection, effects of joining previously not interconnected networks, and other considerations.

IPv4 Address Conflict Detection

Address conflicts in IP networks are annoying problems that (needlessly) take time and effort to detect and rectify, so a separate document on address conflict detection was deemed necessary. "IPv4 Address Conflict Detection"[4] by Stuart Cheshire presents two things: first, a way to prevent this unfortunate situation of conflicting IP addresses from happening, and second, a way to detect address conflicts if they do happen even after all the precautions. Both of these are accomplished using the *Address Resolution Protocol* (ARP). Interestingly, in the Security Considerations section of the document the author states:

> "The ARP protocol [RFC 826] is insecure. A malicious host may send fraudulent ARP packets on the network, interfering with the correct operation of other hosts. For example, it is easy for a host to answer all ARP requests with responses giving its own hardware address, thereby claiming ownership of every address on the network.
>
> This specification makes this existing ARP vulnerability no worse, and in some ways makes it better: Instead of failing silently with no indication why, hosts implementing this specification are required to either attempt to reconfigure automatically, or if not that, at least inform the human user of what is happening."

Although some may argue about the question of whether or not it is effective, appropriate, and useful to "inform the human user" in this case, this solution nevertheless follows the principle of at least not worsening the current security situation of an existing protocol.

Zeroconf Multicast Address Allocation Protocol

The *Zeroconf Multicast Address Allocation Protocol* (ZMAAP) defined in[5] specifies a method for peer-to-peer allocation of *multicast addresses without a multicast* (MADCAP) server in small zero configuration networks. The word "small" is important here because ZMAAP is not scalable beyond small networks (and is not designed to be).

Multicast DNS

"Performing DNS queries via IP Multicast"[6] by Stuart Cheshire suggests some very useful ideas on how to use mDNS with maximum benefit and minimum hassle in zero configuration networks. In my opinion, the best thing about this proposal is that it does not require any changes to the DNS protocol (messages, resource record types, etc.) itself. Instead it concentrates on the use of multicast for name resolution in environments where no DNS servers exist (and where one would not reasonably expect them to). The goal is to have a working name resolution service without name servers. The document proposes to use **`local.arpa`** (although the exact choice of this special domain is not the goal of this document) as the link-local domain (like the **`169.254/16`** network for dynamic allocation of IPv4 link-local addresses described earlier in this article). For reverse address resolution, **`254.169.in-addr.arpa`** is also link-local. The multicast address **`224.0.0.251`** that is used for mDNS queries is registered by the IANA for this purpose. No delegation is performed within mDNS domain **`local.arpa`**. There is also no *Start of Authority* (SOA) record for the mDNS domain because of the nature of zero configuration networks where it is intended to be used—in particular, there is no mailbox responsible for the zone. Likewise, zone transfers are not applicable with mDNS zones. To summarize, any local link has its own local and private **`local.arpa`** and **`254 254.169.in-addr.arpa`** zones, which have only link-local significance in the particular Zeroconf network.

DNS Service Discovery

Like the multicast DNS solution described previously, the *DNS Service Discovery* (DNS-SD)[7] does not require any changes to the existing DNS protocol; thus it is completely compatible with the existing DNS server and client software.

What DNS-SD proposes is a naming scheme for DNS *Resource Records* (RRs) to allow for service discovery using the existing DNS—either the traditional or multicast DNS described in the previous paragraph. DNS-SD uses the SRV and PTR resource records to provide the required functionality. To cite from [7]:

"Service discovery requires a central aggregation server. DNS already has one: It's called a DNS server.

Service discovery requires a service registration protocol. DNS already has one: It's called DNS Dynamic Update.

Service discovery requires a security model. DNS already has one: It's called DNSSEC.

Service discovery requires a query protocol. DNS already has one: It's called DNS."

It is necessary to note that DNS-SD is compatible with mDNS and vice versa, but neither requires the other one to function. However, it is practical to use mDNS for service discovery (using DNS-SD) to have a single protocol and interface and not have to implement another protocol just for service discovery.

Industry Support

Any new technology needs industry support to succeed, and Zeroconf is no exception. Several major vendors have announced plans to support or already support Zeroconf in their products, including Apple, Epson, Hewlett-Packard, Lexmark, Philips, Canon, Xerox, Sybase, and World-Book. One can expect that more companies will Zeroconf-enable their products as the technology itself matures and hopefully becomes standardized and widespread.

Rendezvous

Rendezvous is Apple Computer's implementation of Zeroconf in its Darwin 6 and Mac OS X 10.2 ("Jaguar") operating systems. Apple has stated its full support for the Zeroconf and intent to completely replace the aging AppleTalk with Zeroconf-enabled Macs, without sacrificing the ease of use and transparency to end users provided by AppleTalk networks. A good example of Zeroconf's use in OS X would be the iChat instant messaging (IM) client, which comes with the Version 10.2 of Mac OS X. It works not only with *AOL Instant Messenger* (AIM) and Mac networks but may also be used between Zeroconf-enabled Macs in a Zeroconf network.

Coupled with Apple's implementation of IEEE 802.11b ("WiFi") in ad hoc mode, it permits a wireless zero configuration network that just works without any configuration or additional hardware or software.

Apple has also made the source code for the mDNS Responder, a part of Rendezvous implementing mDNS, freely available through the Darwin Open Source Project. Mac OS X software developers are encouraged to use Zeroconf, and there are documentation and application examples to facilitate this. More information about Rendezvous and Zeroconf on Macs is available from Apple's Web sites[9].

Summary

With computers and computer networks becoming more and more complex and sophisticated, some people (including the author of this article) believe that care should be taken by those in the know not to create more problems than we solve using these computers and networks. Yes, we want more features—but we also need to remember that most users of these features do not have doctorates in computer science and (surprise, surprise) don't even wish to. Zero configuration networking would probably help in this regard, minimizing and even eliminating in some cases the need to configure and administer small networks. Let me conclude by quoting once more from the Zeroconf Working Group:

> "It is important to understand that the purpose of Zeroconf is not solely to make current personal computer networking easier to use, though this is certainly a useful benefit. The long-term goal of Zeroconf is to enable the creation of entirely new kinds of networked products, products that today would simply not be commercially viable because of the inconvenience and support costs involved in setting up, configuring, and maintaining a network to allow them to operate."

References

[1] Zeroconf Working Group, Internet Engineering Task Force (IETF): **http://www.ietf.org/html.charters/zeroconf-charter.html**

[2] Aidan Williams, "Requirements for Automatic Configuration of IP Hosts," **draft-ietf-zeroconf-reqts-12.txt**

[3] Stuart Cheshire, Bernard Aboba, and Erik Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses," **draft-ietf-zeroconf-ipv4-linklocal-07.txt**

[4] Stuart Cheshire, "IPv4 Address Conflict Detection," **draft-cheshire-ipv4-acd-02.txt**

[5] Octavian Catrina, Dave Thaler, Bernard Aboba, and Erik Guttman, "Zeroconf Multicast Address Allocation Protocol (ZMAAP)," **draft-ietf-zeroconf-zmaap-02.txt**

[6] Stuart Cheshire, "Performing DNS Queries via IP Multicast," **draft-cheshire-dnsext-multicastdns-00.txt**

[7] Stuart Cheshire, "Discovering Named Instances of Abstract Services Using DNS," **draft-cheshire-dnsext-nias-00.txt**

[8] Zeroconf: **http://www.zeroconf.org**

[9] Rendezvous: **http://developer.apple.com/macosx/rendezvous/** **http://www.apple.com/macosx/jaguar/rendezvous.html**

[10] Erik Guttman, "Autoconfiguration for IP Networking: Enabling Local Communication," *IEEE Internet Computing,* June 2001.

EDGAR DANIELYAN is a self-employed consultant, author, and editor specialising in UNIX, networking, and information security. In previous life he has been a cofounder of a national ISP and manager of a country TLD. He is currently working on his next book (*WLAN Security*) which is due to be published in 2003. His previous book, *Solaris 8 Security,* was published by New Riders Publishing in 2001. He is also a member of IEEE, IEEE Standards Association, IEEE Computer Society, ACM, USENIX, and the SAGE. He is online at **http://www.danielyan.com** and can be reached by e-mail at **edd@danielyan.com**

Book Reviews

Ruling the Root

Ruling the Root: Internet Governance and the Taming of Cyberspace, by Milton L. Mueller, ISBN 0-262-13412-8, The MIT Press, 2002, **http://mitpress.mit.edu**

> "WASHINGTON, Apr. 1 /Governance Newswire/ — The organizations that create street names, assign addresses, and assign telephone numbers have issued a joint announcement: Henceforth any conversation not conducted in Bahasa Malayu will result in termination of the relevant address or telephone number assignment."

The above bit of fiction is not pure silliness. Fear of equivalent, Internet-related excesses is the essence of Milton Mueller's book, *Ruling the Root*. The Syracuse University professor believes that administration of Internet addresses and domain names provides a fulcrum for overall Internet governance. He says they create a "political economy" vulnerable to serious abuse. Domain name administration is equated with control over Internet content, because, "a domain name record [is] very much like an Internet driver's license" as if it provides permission to use the Net, and even authorizes the locations one may visit.

Organization

The book covers both IP address and domain name administration. The material on IP addresses is thin, perhaps because it is a well-managed area without significant controversy. This is in marked contrast to the recent history of debate on *Domain Name System* (DNS) oversight. So it might have been instructive to see a comparison between the two administrative models, beyond simply noting that domain names can be interesting.

Discussion covers Internet technology, the history and politics of DNS and IP administrative management structure, and the intellectual property aspects of name assignment conflicts. Mueller suggests a three-layer hierarchy: technical, economic, and policy. What is missing from this "architecture" and from the entire book is any concern for the pragmatic details of administration and operation of these global, mission-critical services. Yet such tasks are difficult to perform well, as Network Solutions repeatedly demonstrated over the years, by losing registrations and corrupting critical data files; and the effects of problems are large.

When *Star Trek's* Captain Picard commands, "make it so," we know that he fully appreciates the challenges in implementing his directive. However, for *Ruling the Root*, policy development is not concerned with the operational complexities.

Not surprisingly, the book often demonstrates a misunderstanding of constraints inherent in DNS technology, although the tutorial on basic Internet technology is adequate, in spite of making the common error about the "T" in TCP/IP.[1]

Differing Opinions

Other reviewers of the book have called it well written, insightful, and nuanced. Indeed the discussion of history that is fully documented and involves simple, clear, objective facts is quite good. The rest of the time Mueller presents biased and unfounded descriptions of Internet governance, motives, and decisions, while failing to distinguish between what is fact and what is his opinion.

Ruling the Root sees adversaries, conspiracies, and threats, and permits no balancing sense of diverse collaboration, constructive criticism, or productive compromise. The technical community is somewhat less suspect, but is deprecated with the usual cliche about its naivete. So Mueller misses the essential point that techies designed, built, operated, and grew this robust, survivable, equitable system for global operations and service governance.

Professor Mueller's treatment of the dominant DNS registry, *Network Solutions* (NSI), now VeriSign, is curiously superficial and soft. NSI benefited spectacularly from the National Science Foundation's decision to permit charging for domain names, and from the policies and delays in the formation of the *Internet Corporation for Assigned Names and Numbers* (ICANN), as well as ICANN's distraction away from its intended registry oversight function and toward abstract debates about Internet governance. Yet the book does not consider NSI's role in ICANN-related political processes.

Mueller fails to understand the history of the organization that managed the DNS from its inception, the *Internet Assigned Numbers Authority* (IANA) and Jon Postel's role in running it. IANA is incorrectly represented as a simple operations arm of the U.S. Government. The grass-roots basis for its real legitimacy is missed. Its policy role is missed. Its collaborative processes are denied. For example, Mueller tells us that the description of IANA in RFC 1083, published in 1988 meant, "a new world was being defined by the RFC." In reality it was simply documenting established practice, as is typical for operations RFCs.

Validation

Mueller's substantiation of his analyses is also problematic. The book must be read with careful attention to the actual authority of each source. Goals and agendas are often misstated. For example, he characterizes the pre-ICANN *International Forum for the White Paper* (IFWP) as "the real arena for arriving at a decision [about the details of the new organization]." Its actual goal was simply to be a forum for discussion. Discussion, not decision-making.[2]

The book claims that the pre-ICANN *International Ad Hoc Committee* (IAHC) was formed "to develop and implement a blueprint for a global governance structure for the domain name system." In fact, the IAHC was formed for "specifying and implementing policies and procedures relating to iTLDs (international top-level domains, now called 'generic' TLDs, or gTLDs)."[3] He claims, "They had asserted that the root was theirs to dispose of." To the contrary, the IAHC was explicitly subordinate to IANA, and had nothing at all to do with management of the DNS root or any non-gTLD part of the DNS. Interestingly, the endnote Mueller offers as substantiation disproves his characterization.

Ruling the Root is loaded with endnotes—27 pages of small print. However, even the formal citations are problematic. Note #55 cites a newspaper article as a primary source, as if it were definitive proof the person discussed in the article held a specific opinion. Mueller's Note #45 claims to substantiate that, "Postel himself... admitted...it is unclear who actually controls the name space." Yet the note is for *Internet Architecture Board* (IAB) minutes. Attributing it to Postel was a fabrication.

Back-room, deal-making, conspiracy explanations are offered without substantiation. Of changes to *Internet Engineering Task Force* (IETF) management, Mueller states: "The most important reason the IETF didn't institute voting was that Jon Postel and several other senior figures vowed that they would refuse to run for office." Postel never made such a vow, and the process to effect these IETF changes did not experience any such attempts at influence. Of Postel's instructing some root servers to retrieve copies of the DNS root from a non-NSI master, Mueller claims that Postel was "apparently concerned about the direction U.S. policy was taking."

No substantiation is offered, because the claim is false. Postel and others were concerned about NSI's reaction to its own loss of control. The switch was intended to see what it would take to move NSI out of the hierarchy. These are not small matters of nuance. They show a pattern of misrepresentation.

The Author

Professor Mueller's credibility would have been aided by disclosing his own affiliations. The only ICANN constituency (the Non Commercial Domain Name Holders Constituency) claiming to represent the non-commercial world focuses on the civil society concerns that dominate the public debate about ICANN. Professor Mueller's discussion of the group is quite thin and does not disclose the fact that he held a dominant management position in it. In his criticism of dispute-resolution activities, he neglects to mention that he is a paid arbitration panelist.

An important book should be read because it has factual detail and thoughtful insight. *Ruling the Root* is, instead, important because it so thoroughly embodies the difficulties that have emerged in discussing Internet policy. Because so many people take *Ruling the Root* seriously, it should be read. However, the serious problems of the book encourage borrowing it, rather than buying a copy. Based on the pattern noted in this review, a thorough audit of those problems would be appropriate for the relevant Syracuse University academic ethics committee.

—*Dave Crocker*[4], *Brandenburg InternetWorking*
dcrocker@brandenburg.com

References

[1] The "T" stands for transmission, not transport or transfer.

[2] **http://web.archive.org/web/19981206105122/**
http://www.ifwp.org/

[3] **http://www.iahc.org/iahc-charter.html**

[4] Factual claims in the review that do not have citations are based on the reviewer's direct experience. Dave Crocker wrote the first Internet standard for domain name syntax (RFC 822). He also was the IETF area director for initial work on DNS security. More recently he was one of Jon Postel's appointees to the IAHC. He naively thought that its work should be conducted in the manner that had been typical for Internet administration. So the last few years of charged, global politicization have been an education. He must also note that he was once Jon Postel's officemate.

High-Speed Networks and Internets

High-Speed Networks and Internets: Performance and Quality of Service, 2nd ed., by William Stallings, ISBN 0-13-032221-0, Prentice Hall, 2002. **http://www.prenhall.com/stallings**

This thoroughly updated classic covers topics of traffic engineering, queuing, and traffic modeling. The book gives a complete look around the protocols of the next generation: *Resource Reservation Protocol* (RSVP), *Multiprotocol Label Switching* (MPLS), and *Real-Time Transport Protocol* (RTP). It gives the keys to understand the way Frame Relay, TCP, and ATM react to congestion and flow control. The book also deals with new trends and standards that will lead the telecommunications industry in the following years. A very useful book, from the same author of traditional titles such as: Data Communications, Cryptography, Computer Architecture, and many more.

Organization

High-Speed Networks is divided into seven parts. The first one discusses the basic background needed to understand the rest of the book. Following the introduction, the second chapter goes on with the classical: the *Open System Interconnection* (OSI) model and the TCP/IP suite.

Part II explains packet-switching technologies in detail. The forth chapter explains the architecture of Frame Relay, and the next one focuses on ATM, including its operation and the adaptation layers. Chapter 6 works on high speed LANs, covering Fast Ethernet and Gigabit Ethernet, with the different media supported by each.

The third part is one of the most important; chapter 7 presents an overview of probability and stochastic processes. Although it is a brief one, it is useful to make revision of some concepts. The next chapter works on queuing analysis, introducing the basic elements of a queuing model. It explains the topics with plenty of examples: M/M/1, multiserver queues, and networks of queues, presenting all the formulas. Chapter 9 is dedicated to self-similar traffic. As recent studies indicate, traffic on high speed networks does not have the characteristics needed for the queuing theory. It introduces and explains the concept of self-similarity. Then the author applies this concept to data traffic analysis and examines performance implications. Based on papers on this subject, Stallings explains this new approach to traffic modeling not analyzed before.

The forth part focuses on another main topic: congestion and traffic management. Chapter 10 explains the effects of congestion and the different ways to control and avoid it. In the following chapter the author discusses control mechanisms at the link level. He examines different ways used by protocols to handle flow control: *Stop and Wait, Sliding Window,* and *Go back N-ARQ*. An analysis of the performance gained by using *Automatic Repeat Request* (ARQ) techniques follows.

These chapters give a detailed description of the different ways that communications can be handled. Chapter 12 focuses on transport-level traffic management. It explains TCP flow control in detail, including the retransmission strategy. The way TCP avoids congestion is discussed thoroughly. The next chapter continues with congestion control in ATM networks. The framework for traffic control is explained in detail, with sections dedicated to *Available-Bit-Rate* (ABR) and *Guaranteed-Frame-Rate* (GFR) traffic management.

The next part of the book is about Internet routing. Chapter 14 presents the algorithms used to compute the minimum path, and introduces some elementary concepts in graph theory. Later the author concentrates on Interior routing protocols, analyzing the *Routing Information Protocol* (RIP) and *Open Shortest Path First* (OSPF), the most important ones. Next the book discusses exterior routing protocols and multicast. The author describes in a simple way these addressing schemes and the related protocols.

The following section is dedicated to *Quality of Service* (QoS) in IP networks. The first chapter discusses integrated services, with coverage of queuing disciplines such as *Weighted Fair Queuing* (WFQ). A review of the Differentiated Services architecture follows.

After discussing the concepts, the author examines the protocols that support QoS: RSVP, MPLS, and RTP. He explains the philosophy behind each protocol, its characteristics, and its implementation.

In the final part of the book, the author changes the subject to compression. In Chapter 19 he presents an overview of information theory, discussing typical areas such as entropy. The next chapter continues with loss-less compression, facsimile compression, and others. It discusses the Lempel-Ziv algorithm used in PKZIP. The final chapter reviews lossy compression, explaining the discrete cosine transform, a key component of the *Joint Photographics Expert Group* (JPEG) and *Motion Picture Experts Group* (MPEG) standards.

Two very interesting appendices end the book: one for Internet standards and the standardization process and the other one dedicated to sockets, containing source code. Although the book is not dedicated to programming, the inclusion of TCP sockets can be useful to understand its implementation.

A book worth reading

We are facing an essential book for networking professionals, designers, and engineers. It covers unusual topics such as self-similar traffic and data compression. It is the basement for the design of any high speed network. As Internet traffic continues to grow, the optimization of network resources becomes a critical topic. Also, more and more voice traffic is carried over packet networks, congestion being one of its worst enemies. The time-sensitive traffic needs attention, and this book provides the tools to manage it.

In addition to its solid coverage of topics, the book has plenty of bibliography and many links to the principal sites for each chapter. With no doubt this is a very useful book, from the well-known technical author William Stallings.

—Rodrigo J. Plaza, Iplan Networks, Argentina
rplaza@iplan.com.ar

Would You Like to Review a Book for IPJ?

We receive numerous books on computer networking from all the major publishers. If you've got a specific book you are interested in reviewing, please contact us and we will make sure a copy is mailed to you. The book is yours to keep if you send us a review. We accept reviews of new titles, as well as some of the "networking classics." Contact us at **ipj@cisco.com** for more information.

Letters to the Editor

ENUM

Ole,

As the co-chair of the ENUM work group in the IETF, I was delighted with Geoff Huston's article. (*The Internet Protocol Journal*, Volume 5, No. 2, June 2002, page 13).

I would like to point out and clarify several other issues raised by the Letters to the Editor published in the subsequent issue.

First, as a practical matter though the North American Numbering Plan uses a single country code "1," there will not be a single administration of ENUM within "1." The agreements between the IAB and the ITU on the administration of **e164.arpa** clearly indicate that these resources will be administered on a nation-state basis.
www.iab.org/DOCUMENTS/enum-pr.html
www.iab.org/DOCUMENTS/sg2-liaison-e164-sep-02.html

The United States, Canada, Bermuda, and the 18 countries of the NANP will be free to administer their numbering resources as they so choose through the use of 1 + NPA (area codes) zones within the root of **e164.arpa**.

Dr. Deleuze writes, "E.164 numbers are really telephone addresses. They are tied to telephone network topology and are surely not user friendly. There are no user-friendly names in the telephone system."

In fact, this is not exactly correct either. Since the advent of Number Portability by several national telephone administrations, including the United States, telephone numbers are no longer tied to the underlying network or routing structure of the PSTN. Actual routing of phone calls in the United States is done on Local Routing Numbers for all landline calls and, beginning in November of 2003, for wireless calls as well.

Phone numbers even now are essentially names, much like domain names in the Internet. In the United States, phone numbers can be taken or "ported" to any wireline service provider within proscribed geographic boundaries, in 2003 between wireless service providers and from wireline to wireless providers as well.

I partially take issue with Dr. Deleuze's thought that telephone numbers are not "user-friendly." Phone numbers are readily identifiable, easy to use, and are not tied to culture or language, problems we have not yet solved with domain names.

—*Richard Shockey, NeuStar Inc.*
rich.shockey@NeuStar.com

Visitor Networks

Dear Editor,

The September 2002 issue of IPJ featured a very interesting, comprehensive article on visitor networks. One aspect I found not mentioned, however, is the danger of users in such scenarios falling victim to fake visitor gateways. In public wireless hot spots, as they are increasingly being setup at numerous locations these days, attackers could employ their own mobile WLAN device to direct visitors trying to log on to the hot spot to their own fake login page, enabling them to easily collect their login details such as credit card information. Using encryption does not help here as long as the gateway does not need to authenticate itself to the customer's mobile device. The average user should not have a chance to realize whether he or she is connected to a legitimate or a fake login page—if he or she is aware of that potential danger at all. Given the fact that all such an attack would need, apart from readily available equipment such as a portable computer with a WLAN card, is some small piece of appropriate software and that it would be quite difficult to detect,that kind of threat unfortunately should be quite realistic in such environments.

—Dr. Georg Schwarz
Detecon International GmbH, Berlin, Germany
Georg.Schwarz@detecon.com

The author responds:

This is a good point that was not discussed in the article. There are actually at least three cases that visitors need to worry about. The first is, as you mentioned, that the service provider is not who they say they are. This can be dealt with by using SSL certificates assuming the visitor is conscious of the URL that he/she is being directed to and knows that it belongs to the real service provider. If the visitor has no idea who is a reasonable service provider, this is a different class of problem, very similar to what has happened with public telephones that accept standard calling and credit cards—someone makes a call, receives the service but then gets charged an outrageous rate. The third case is a man-in-the-middle attack or passive snooping where someone with a laptop as you describe is able to grab traffic and gather passwords.

Some basic advice to visitors is for services that require subscription, although possibly inconvenient, never subscribe on a potentially compromised connection. That way, only the service provider-assigned username and password is compromised, instead of more sensitive personal information related to the account. Connections using 802.1x authentication with EAP-TLS provide mutual authentication and are in the long run, a better solution than redirection of web pages. No matter what kind of security one has, inevitably there will be legally legitimate providers that will take advantage of visitors and in that case it's just "buyer beware."

—Dory Leifer
leifer@del.com

Wireless Security

Hi Ole,

Again, I found the latest issue of IPJ quite enlightening and useful. However, I do have one comment regarding the article by Greg Scholz on "An Architecture for Securing Wireless Networks." Although the use of source IP addresses to provide policy group membership on the firewall works in most cases, some client OSs and some IPSec VPN boxes allow the source address (even if it is the endpoint address of the tunnel, not the "real" address of the host) to be changed,provided the source address of the enciphered traffic does not change. This would allow users to change the policy group they belong to. A better solution is to use a VPN box that can associate groups of IPSec tunnels to VLANs. Then the firewall could be configured to allow policy group membership based on VLANs. This takes all determination of policy group membership off the client host and places it in the domain of trust of the VPN and firewall boxes.

—Chris Liljenstolpe
Cable and Wireless
chris@cw.net

Fragments

Upcoming Events

The IETF will meet in San Francisco, California, USA March 16–21, 2003. The IETF will also meet in Vienna, Austria, July 13–18, 2003 and in Minneapolis, Minnesota November 9–14, 2003.
See **http://www.ietf.org/meetings**

The next APRICOT (*Asia and Pacific Regional Internet Conference on Operational Technologies*) will be held in Taipei, Taiwan, February 19–28. See **http://www.apricot2003.net/**

The *Internet Corporation for Assigned Names and Numbers* (ICANN) will meet in Rio de Janeiro, Brazil, March 23–27, 2003, in Montreal, Canada, June 22–26, 2003, and in Carthage, Tunisia, December 1–5, 2003. See **http://www.icann.org**

The Internet Protocol Journal

Ole J. Jacobsen, Editor and Publisher

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

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
Cisco Systems, Inc.