The Internet Protocol Journal - Volume 5, Number 4

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. PIMSM, 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-andprune 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 heir knowledge. Multicast (and its application level cousin, the CUSeeMe 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-ton 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 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. *n*&215; (*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).

	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-tosource 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