Internet Multicast Routing and Transport Control Protocols

VICTOR O. K. LI, FELLOW, IEEE, AND ZAICHEN ZHANG, STUDENT MEMBER, IEEE

Contributed Paper

Multicasting is a mechanism to send data to multiple receivers in an efficient way. In this paper, we give a comprehensive survey on network and transport layer issues of Internet multicast. We begin with a brief introduction to the current Internet protocol multicast model—the "host group" model and the current Internet multicast architecture, then discuss in depth the following three research areas: 1) scalable multicast routing; 2) reliable multicast; and 3) multicast flow and congestion control. Our goal is to summarize the state of the art in Internet multicast and to stimulate further research in this area.

Keywords*—Congestion control, flow control, internet, multicast, reliable multicast, routing.*

NOMENCLATURE

ACK	Positive acknowledgment.
ADU	Application data unit.
AER	Active error recovery.
AFDP	Adaptive file distribution protocol.
AIMD	Additive increase/multiplicative decrease.
ALF	Application-level framing.
AMA	Aggregated multicast address.
ANTS	Active node transport system.
ARM	Active reliable multicast.
ARQ	Automatic repeat request.
BGP	Border gateway protocol.
BGMP	Border gateway multicast protocol.
CBT	Core-based tree.
CIDR	Classless interdomain routing.
CLM	Connectionless multicast.
DIS	Distributed interactive simulation.
DM	Domain manager.
DR	Designated receiver.
DSG	Destination-set grouping.
DSS	Destination-set splitting
DTRM	Deterministic timeouts for reliable multicast.
DTM	Dynamic-tunnel multicast.
DVMRP	Distance-vector multicast routing protocol.
ECMP	EXPRESS count management protocol.
ERS	Expanded ring search.
ESM	End system multicast.
EWMA	Exponentially weighted moving average.
EXPRESS	Explicitly requested single source.
FEC	Forward error correction.
FLICA	Filtered-loss indication-based congestion avoidance.
FS	Fair scheduler.
GC	Group controller.
GUM	Grand unified multicast.
IA	Intermediate agent.
ID	Identifier.
IGMP	Internet group management protocol.
IP	Internet protocol.
ISO/OSI	International Standardization Organization/Open System Interconnection.
ISP	Internet service provider.
LBRM	Log-based receiver-reliable multicast.
LGC	Local group concept.
LI	Loss indicator.
LIF	Loss-indication filter
LMS	Lightweight multicast service.
LSA	Link-state advertisement.
LTRC	Loss tolerant rate controller.
LVMR	Layered video multicast with retransmission.
LWS	Lightweight sessions.
MASC	Multicast address-set claim.
MBGP	Multiprotocol BGP4/multicast BGP.
MBone	Multicast backbone.
MDO6	Multiple destination option on IPv6.
MDP	Multicast dissemination protocol.

Manuscript received March 3, 2001; revised December 5, 2001. This work was supported in part by the Areas of Excellence Scheme established under the University Grants Committee of the Hong Kong Special Administrative Region, China (Project AoE/E-01/99).

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: {vli; zczhang}@eee.hku.hk).

Publisher Item Identifier S 0018-9219(02)02903-1.

MFT	Multicast forwarding table.
MIGP	Multicast interior gateway protocol.
MOSPF	Multicast OSPF.
MPLS	Multiprotocol label switching.
MSDP	Multicast source discovery protocol.
MTCP	Multicast TCP.
MTP	Multicast transport protocol.
NACK	Negative acknowledgment.
NAP	Network access point.
NAPP	Negative acknowledge with periodic polling.
NCA	Nominee-based congestion avoidance.
NSP	Network service provider.
OSPF	Open shortest path first.
PGM	Pragmatic general multicast.
PIM	Protocol-independent multicast.
PIM-DM	PIM dense mode.
PIM-SM	PIM sparse mode.
PLM	Packet-pair receiver-driven cumulative layered multicast.
QMTP	Quasi-reliable MTP.
QoS	Quality of service.
RA	Routing arbiter.
RALM	Router-assisted layered multicast.
RAMP	Reliable adaptive multicast protocol.
RBP	Reliable broadcast protocol.
RED	Random early drop.
REUNITE	Recursive unicast tree.
RGMP	Receiver-initiated group-membership protocol.
RLC	Receiver-driven layered congestion control.
RLM	Receiver-driven layered multicast.
RLMP	RLM with priorities.
RMANP	Reliable multicast active network protocol.
RMCM	Reliable multicast for core-based multicast trees.
RMP	Reliable multicast protocol.
RMTP	Reliable MTP.
RNP	Regional network provider.
RP	Rendezvous point.
RPF	Reverse path forwarding.
RSE code	Reed–Solomn erasure code.
RSVP	Resource reservation protocol.
RTP	Real-time transport protocol.
RTCP	RTP control protocol.
RTT	Round-trip time.
SA	Subnet agent (in LVMR) or sender's agent (in MTCP).
SGM	Small group multicast.
SM	Simple multicast.
SP	Synchronization point.
SR	Session relay (in EXPRESS).
SRM	Scalable reliable multicast.
SSM	Source-specific multicast.
TCP	Transmission control protocol.
TFMCC	TCP-friendly reliable multicast congestion control.
TMTP	Tree-based MTP.
TP	Turning point.
TRAM	Tree-based reliable multicast.
TTL	Time to live.
VBR	Variable bit rate.
XTP	Xpress transport protocol.

I. Introduction

A. Introduction to the Internet

The Internet is organized as a loose hierarchy, as illustrated in Fig. 1.

In the center of the hierarchy are primary NSPs, such as MCI WorldCom Inc., Sprint, and Internet II. NSPs are interconnected by high-speed links and provide Internet access to National ISPs and RNPs through NAPs. Attached to every NAP is an RA, which provides routing information for that NAP. Primary NSPs, the high-speed links between them and NAPs are often collectively called the Internet backbone. Routers[1] on the Internet backbone, called core routers, use BGP [1] to dynamically learn routing information and do not use default routing.[2] Local ISPs connect to Internet through National ISPs, RNPs, or at NAPs to an NSP directly and provide Internet service to their customers.

In the Internet, blocks of IP addresses are allocated to ISPs. An ISP then divides its allocation and assigns smaller blocks to its customers, which may be low level ISPs or individual customers. Hosts sharing a common part of the IP address (see the following introduction to IP addresses) are said to be in the same domain. In the Internet, domains are often organized hierarchically.

An IPv4[3] address usually contains a network ID and a host ID. A network ID is used to route a packet to its destination network and host ID is used to reach the destination host in that network. There are four classes of IP addresses: classes A, B, C, and D, as shown in Fig. 2.

Class A, B, and C addresses are used to identify hosts in the Internet and for unicast routing[4] and class D addresses are used for multicast routing. However, the granularity of the class-based division of the IP address space is too coarse to use IP addresses efficiently. For example, a company having 1000 hosts will ask for a block of class B addresses and leave most of them unused. With the rapid growth of the Internet, this inefficiency will quickly exhaust all IP addresses. To extend the lifetime of IPv4, CIDR [3] is proposed. It does not assign addresses according to class boundaries. Instead, an address in CIDR is associated with a network "prefix," which replaces the network ID in the traditional class-based scheme. An example CIDR address is 147.8.182.174/22, where "22" is the network prefix,

[1]A router switches a packet from an incoming link to an outgoing link on its way toward the destination.

[2]A router not on the Internet backbone usually has a default entry in its routing table, which is used to forward packets toward the backbone.

[3]In IPv4, the current version of IP, an IP address is a 32-bit number and is usually represented by four decimals, like 147.8.182.174. An IP address will be extended to 128 bits in the next generation of IP (IPv6 [2]).

[4]Unicast routing is the routing between a sender and a receiver.

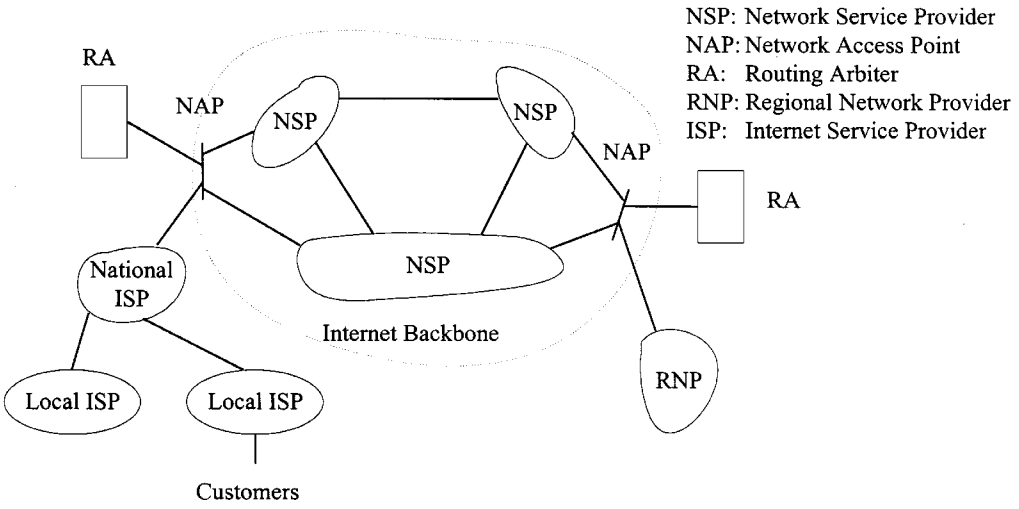

Fig. 1. Internet hierarchy.

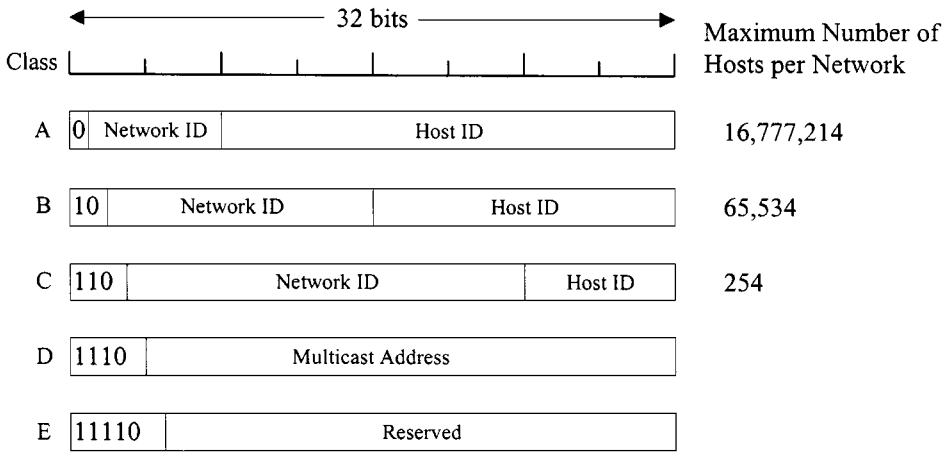

Fig. 2. IPv4 addresses.

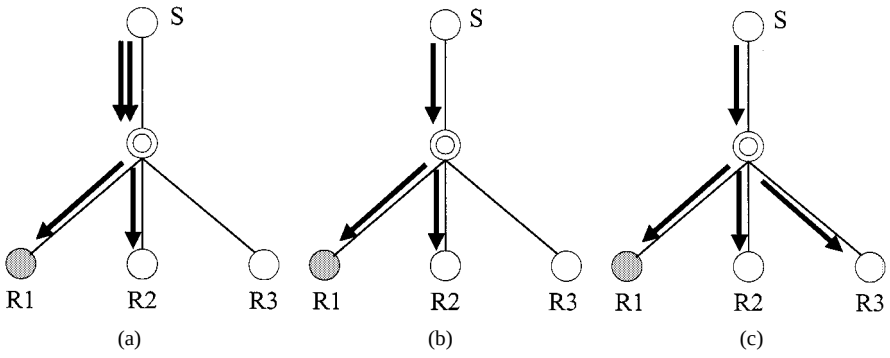

Fig. 3. Illustration of (a) unicast, (b) multicast, and (c) broadcast.

indicating that the first 22 bits in the address represent a network and the remaining bits indicate a specific host. Currently, network prefix in CIDR ranges from 13 to 27, providing more flexibility to fit various requirements in IP address allocations. CIDR also enables "route aggregation," i.e., a single high-level route entry in the routing table can represent many lower level routes. This helps reduce routing table size and shorten routing time at routers.

B. Introduction to Multicasting

Multicasting refers to sending datagrams to a subset of destinations in the network. Ideally, in multicast, the sender only needs to send every datagram once and there is at most one copy of the datagram on every physical link. Compared with broadcast, only relevant routers and hosts take part in the transmission and reception of multicast datagrams. The concept is illustrated in Fig. 3.

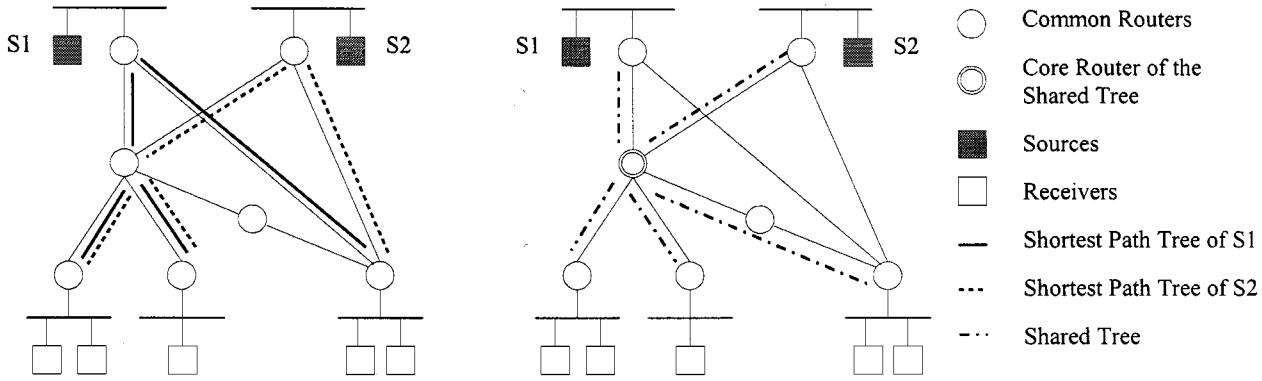

Fig. 4. Shortest path tree and shared tree. (a) Shortest path trees in multicast routing. (b) Shared tree in multicast routing.

Suppose we want to send a message from S to R1 and R2. In unicast, a copy of the message is sent to the two receivers separately and duplicate copies will appear on some physical links. In multicast, only one transmission is made by the sender S. At each intermediate node, copies are made and sent, as required, to outgoing links. At most, one copy is required on each physical link. In broadcast, copies are made and sent to each outgoing link at each intermediate node. As a result, even nodes which do not require a copy, such as R3 in Fig. 3, will get the message.

Nowadays, many emerging applications in the Internet require point/multipoint to multipoint delivery, such as audio/video conferencing, web cache updating, one-to-many file distribution, distance learning, and Internet games. Internet multicast is crucial to the development of the Internet due to its ability of delivering point/multipoint to multipoint data in an efficient and scalable way. A protocol is called scalable if it works efficiently even as the size of the network increases.

The first Internet multicast paradigm, the "host group" model [4], is proposed in the late 1980s. Since 1992, Internet multicast has been tested and implemented on the MBone [5]. However, multicast is far from being fully developed. There are many open issues that require further investigation. In this paper, we summarize Internet multicast research in several major areas, including scalable multicast routing, reliable multicast and multicast flow and congestion control. We will not only introduce proposed schemes and protocols, but also discuss the reasoning and design philosophy behind them. Our goal is to summarize existing work and promote further research.

1) "Host Group" Model: The "host group" model of multicast is proposed in 1989 [4]. In this model, the hosts participating in the same multicast session form a host group identified by a single class D IP address. A host may join and leave the group at any time and may belong to more than one group at a time. To send datagrams to a group, a host need not know the membership of the group, or be a member of the group. Data delivery in the "host group" model is best effort. Senders multicast to and receivers receive from their local links and it is the "multicast routers" that have the responsibility of delivering the multicast datagrams.

2) Current Internet Multicast Architecture: The current Internet multicast architecture is largely evolved from the "host group" model. It consists of the group management protocols, the IP multicast routing protocols, and multicast transport protocols.

The group management protocols are used for group member hosts to report their group information to the multicast routers on the subnet. IGMP [4], [6], [7] is the group management protocol currently used in Internet multicast. However, new protocols are still emerging, such as the RGMP [8].

Multicast routing protocols on the Internet deal with the problem of efficiently transmitting multicast datagrams from the source subnetwork(s) to the destination subnetworks. A natural routing structure for multicasting is a tree. The proposed multicast routing protocols differ in how the multicast trees are constructed and what IP unicast routing algorithms are used when constructing the trees. Currently, there are mainly two kinds of multicast trees: source-based shortest path tree and shared tree, as illustrated in Fig. 4. As will be explained later, the core router serves as the root of the shared tree.

DVMRP [9], PIM-DM [10], and MOSPF [11] use shortest path trees, while PIM-SM [12], CBT [13], [14], and BGMP [15] use shared trees. The shared tree in PIM-SM can be switched to a shortest path tree when needed.

The trees established by a multicast routing protocol are usually reflected on the MFTs in the on-tree routers. A common MFT is indexed by group IDs and for each group ID, there is a set of outgoing interfaces (oifs) and, optionally, a set of incoming interfaces (iifs). The group ID includes the (source, group) pair [usually written as (S, G)] in shortest path trees and only group address [usually written as (*,G)] in shared trees. If an incoming multicast packet matches a group ID in the MFT, the iif is checked to see whether it comes from the correct interface (protocols using bidirectional trees like CBT and BGMP do not perform this checking). If it checks, the packet is forwarded to all the oifs in the oif list of this MFT entry; otherwise, it will be discarded.

The DVMRP protocol incorporates the distance vector algorithm to provide routing information. Based on this infor-

mation, each multicast router checks whether a packet is received from the interface used by the router to send packets to the sender. If so, the packet is forwarded according to the oif list of the corresponding (S, G) entry; otherwise, it is discarded. This is called RPF. The (S, G) entry is set when the first packet sent from sender S to group G is received, with the oif list including all the interfaces except the incoming one. Some of the oifs will be pruned by prune messages sent from downstream multicast routers that do not use this router as an upstream router to the sender or do not wish to receive data of group G. The pruned interfaces are marked as "pruned" and will be restored after a certain time-out period. Therefore, downstream routers need to send prune messages periodically to keep an interface "pruned." This is called "flood and prune." A downstream router can also send a "graft" message to cancel a "prune" state immediately. PIM-DM is very similar to DVMRP. The main difference between them is that PIM-DM does not depend on a certain underlying unicast routing protocol.

The MOSPF protocol is the multicast extension of the link-state routing protocol OSPF version 2 [16]. By flooding a new LSA, called group membership LSA, each router in the domain has complete knowledge of the network and membership information. When the first datagram of a group arrives, each router builds the shortest path tree rooted at the sender of the datagram and caches the tree for future usage.

While DVMRP and MOSPF build a shortest path tree for each source in each group and are based on a specific unicast routing algorithm to provide routing information, CBT and PIM-SM use shared tree(s) for each multicast group and can operate with any unicast routing protocol. In CBT, each group has a core router serving as the root of the shared tree. Senders send datagrams toward the core and receivers receive them from the shared tree. PIM-SM works in a similar way, but the core router is now called the RP. There are three main differences between CBT and PIM-SM: 1) the shared trees built in CBT are bidirectional, while in PIM-SM they are unidirectional; 2) PIM-SM trees are "soft state," maintained by periodical "join" messages, while CBT trees are "hard state" and an explicit tear down message is needed to delete a state; and 3) if the traffic volume exceeds a certain threshold, in PIM-SM a router can switch from the shared tree to a shortest path tree.

In DVMRP and PIM-DM, data is broadcasted to flood the network initially and each multicast router needs to send prune messages to stop receiving data that it does not want. In MOSPF, each multicast router needs to gather group membership information of local links and flood group membership LSAs in the network. Therefore, the above protocols, which we shall call dense mode protocols, are more suitable for regions where group members are densely distributed. On the other hand, CBT and PIM-SM are sparse mode protocols designed for sparse regions. Only multicast routers with local group members or needed for transmission will join the shared tree of the group. All other multicast routers will be unaware of the group. In this sense, CBT and PIM-SM have better scalability than dense mode protocols. However, CBT and PIM-SM still need to flood the core/RP information to all multicast routers, so that they can join a core/RP for a certain multicast group.

For these reasons, the above protocols cannot be used directly for Internet wide interdomain multicast. To perform interdomain multicast, several solutions are proposed and developed. One of them is the MBGP [17]/PIM-SM/ MSDP [18] scheme, which, although easy to implement, is only considered an interim solution due to its lack of scalability. Other efforts include the BGMP/ MASC solution, SM [19], EXPRESS multicast [20], etc. They will be introduced in this paper later.

Although multicast routing protocols provide best effort delivery of multicast datagrams on the Internet, many multicast applications have requirements beyond this. Therefore, various multicast transport protocols are proposed on top of the multicast routing protocols to meet the needs of different applications. In [21], they are classified according to the kind of applications they support. "General purpose" protocols, such as RBP [22], MTP [23], and XTP [24], are designed to provide a general solution to the group communication problem. They represent the earlier stage of the multicast transport protocol deployment. It is realized later that a single generic protocol cannot meet the requirements of all multicast applications and, thus, most recent protocols are designed with some specific applications in mind. Some protocols are designed for multipoint interactive applications, such as RTP/RTCP [25] and SRM [26], while others support data dissemination services, such as the MDP [27] and the AFDP [28].

Multicast transport protocols serve two major functions, namely, providing reliability and performing flow and congestion control. There are different definitions of reliability for different multicast applications. For example, total reliability is more suitable for reliable bulk-data transfer such as file distribution, while semireliability and time-bounded reliability are designed for loss-tolerant real-time applications such as video conferencing. We will discuss them in detail in Section III. Multicast flow and congestion control, which is discussed in Section IV, is crucial to the development of multicast in today's best effort Internet. Extensive research has been done in recent years in this area and new protocols and schemes have emerged. However, the problem of multicast flow and congestion control is far from being solved. The major challenges in both reliable multicast and multicast flow and congestion control are due to the requirements to provide scalability and to deal with heterogeneity in a multicast group.

II. Scalable Multicast Routing

Scalability considerations on Internet multicast routing include: 1) a protocol should scale well for large groups which have members scattered over large areas and 2) a protocol should deal with a large number of concurrent groups. The first aspect is dealt with by sparse mode multicast routing protocols, interdomain multicast routing protocols, and "channel" multicast service models such as EXPRESS [20] and SM [19]. The second aspect of scalability can

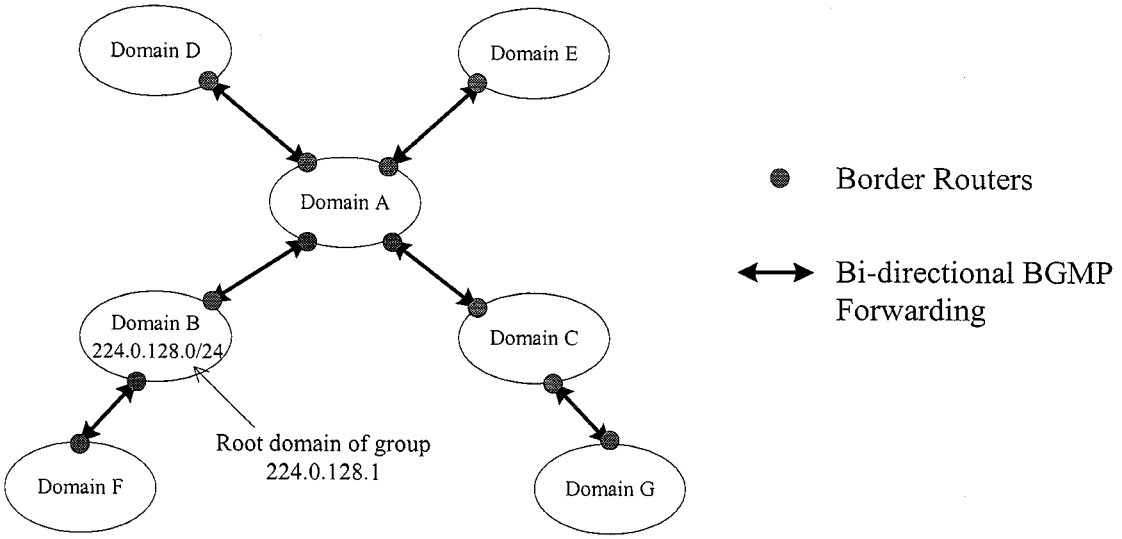

Fig. 5. Illustration of BGMP. 224.0.128.0/24 represents the address range from 224.0.128.0 to 224.0.128.255.

be achieved by multicast address aggregation or by using unicast approaches to eliminate part or all of the multicast states in the routers.

A. Scalable Multicast Routing for Large Groups

Sparse mode routing protocols such as PIM-SM and CBT are proposed to support multicast groups with sparsely distributed members. In these protocols, only relevant hosts and routers are aware of a certain group. However, there is still a need to advertise the RP or core information all over the network. Therefore, these protocols do not directly satisfy the needs of Internet-wide multicast for which several interdomain multicast routing protocols are designed. At the same time, other efforts try to use another service model, the "channel" model, instead of the traditional "host group" model to deliver data to large groups, such as EXPRESS and SM.

1) Interdomain Multicast Routing: To perform Internet-wide multicast and to deal with the scalability problems, interdomain multicast routing proposals usually adopt a hierarchical architecture in which intradomain multicast routing protocols, such as DVMRP and PIM, are used in each domain and interdomain multicast routing protocols are used for transmitting multicast data between domains.

Note that although we only consider the scalability issue, there are many other issues such as policy and stability that need to be considered in interdomain multicast routing.

a) MBGP/PIM-SM/MSDP solution: As mentioned above, MBGP/PIM-SM/MSDP is the near-term solution for the interdomain multicast routing problem. It includes three protocols: MBGP, PIM-SM, and MSDP.

MBGP extends BGP4 messages so that routes corresponding to different protocols can be implemented. However, it does not carry multicast group information. The next hop information provided by MBGP is used in PIM-SM to construct a multicast tree connecting multiple domains. In each domain, there is an RP and the interdomain PIM-SM shared tree has multiple RPs. The MSDP is used to disseminate source information of one domain to other domains, so that receivers in other domains can receive data multicast by the source and switch to a shortest path tree when needed. MSDP peers exchange messages using TCP connections and RPF-flooding.

MBGP/PIM-SM/MSDP is easy to implement because it is based largely on existing protocols. However, it has difficulties dealing with dynamic groups and does not scale well due to the flooding of MSDP messages.

b) BGMP/MASC solution: The BGMP was previously known as GUM. As illustrated in Fig. 5, BGMP builds interdomain bidirectional shared trees rooted at a single domain. In each domain, any multicast routing protocols can be used for intradomain routing and they are called MIGPs. Besides the bidirectional shared trees, source-specific branches are also used in BGMP primarily to avoid data encapsulation. The root domain of a group's shared tree has the multicast address range that covers the group's address and it is often the group initiator's domain. The choice of the root domain has a great impact on the performance. For example, it is important to avoid the "third party dependency" problem in which the delivery of a multicast session depends on a third domain that contains neither senders nor receivers and whose only function is to provide an RP.

In BGMP, since each domain needs to have a range of multicast addresses to be used by groups rooted in the domain, a hierarchical multicast address allocation scheme is required. MASC, based on the structure of the interdomain topology, uses a "claim-collide" mechanism to hierarchically allocate addresses among domains [15]. The MASC address allocation scheme enjoys the advantages of simplicity, relatively high address space utilization, policy support, and robustness. With the decoupling of intra- and interdomain multicast address allocations, addresses can be internally allocated very quickly.

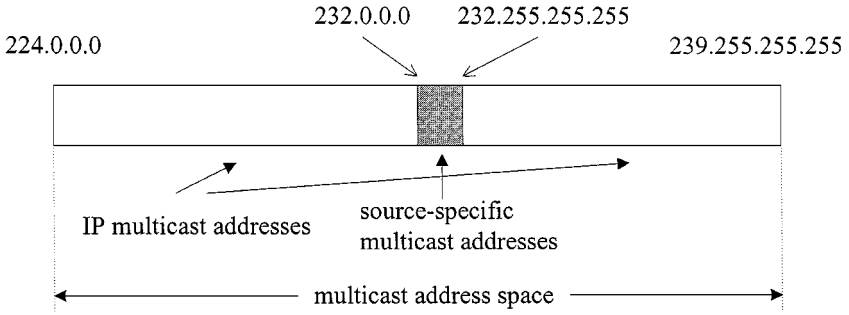

Fig. 6. SSM addresses.

2) Source-Specific Multicast and EXPRESS: SSM [29], [30] is a different service model from the "host group" model defined in [4]. It is designed for single (or almost single[5]) source multicast sessions and may be used for both intra- and interdomain multicast routing due to its scalability to large number of group members. Recently, a new working group, SSM, is created in IETF for this multicast service model.

In SSM, a multicast session is identified by both a source IP address S and an SSM destination address G and the (S, G) pair is referred to as a "channel." The multicast address G is restricted in the reserved multicast address space 232/8 (232.0.0.0 to 232.255.255.255), as shown in Fig. 6. Unlike the "host group" model, two channels (S1, G) and (S2, G) are usually used for different multicast sessions. Receiving sockets/hosts subscribe to a certain channel to receive traffic from it.

In addition, only a single source S can transmit to a channel (S, G) and access control is straightforward. Since the SSM address is local to each source, address allocation is no longer a problem. One reason for SSM's better scalability over sparse mode multicast routing protocols such as PIM-SM and CBT is that the distribution tree for the (S, G) channel is always rooted at the source S, thus, eliminating the need for an RP or core discovery mechanism, which usually requires flooding the whole network.

EXPRESS multicast [20] is an SSM approach. It supports large-scale single (or almost single) source applications such as Internet TV, distance learning and file distribution. It uses a simple integrated protocol, ECMP, to support subscription, multicast channel maintenance, voting, and counting. To support almost single source applications like distance learning, an SR approach is proposed, where an application-selected host acts as the SR and alternative senders send their packets to the SR for delivery to the group. The distinct advantage of the SR approach is that it gives applications the flexibility to select the SRs, unlike in PIM-SM or CBT where the RP/core placement is part of the network configuration.

3) Simple Multicast: SM [19] uses the same basic idea as EXPRESS or SSM—identifying a multicast session using a pair of addresses (C, M). The difference is that here C is the IP address of the core instead of the source and M is the multicast address, but is not restricted to the range 232/8.

[5]An example of almost single source multicast is distance learning, where the lecturer is the primary source, but students may also become secondary sources when they ask questions.

SM is suitable for both intra- and interdomain multicast routing. Scalability is achieved through trivial address allocation (addresses only have to be unique per core), decoupling core selection, and discovery from the multicast protocol (e.g., in SM, end systems may select the core C for a group through email announcement) and a bidirectional shared tree.

It is worth noting that for SM to coexist with non-SM aware routers, [19] suggests carrying both C and M in the SM header instead of in the destination address field. Moreover, the destination address is set to a reserved multicast address, the ALL-SM-NODES. However, in EXPRESS, the address G in the channel ID (S, G) is put into the destination address field because EXPRESS uses a reserved 232/8 multicast address space.

B. Scalable Multicast Routing for Large Number of Groups

With the development of multicast in the Internet, protocols that scale to a large number of concurrent groups are demanded, especially for the core/backbone multicast routers. One can try to aggregate multicast addresses in the core routers as is done in unicast, but this is not easy and elegant and efficient solutions are still lacking. Alternative approaches eliminate multicast states on the nonbranching routers on a multicast distribution tree or resort to unicast routing to avoid using multicast address and maintaining multicast states in the routers.

1) Multicast Address Aggregation: In today's Internet, unicast addresses are allocated in a hierarchical way, according to the hosts' positions. The CIDR [3] address allocation scheme (described earlier) is adopted to allocate unicast addresses more efficiently and enable "route aggregation." Therefore, the unicast address is aggregatable.

However, the multicast address is not easily aggregatable. As in the "host group" model, a multicast group's address is the ID of the group. It is independent of the locations of the sender(s)/receivers. Furthermore, because a multicast group has one or more senders and multiple receivers and they may be in different locations, aggregating several multicast groups is still difficult even if a location-aware multicast address allocation scheme such as MASC is used. Therefore, every multicast router has to maintain at least one entry for each group using it for multicasting. In the future, when Internet multicast is widely deployed, this may be a formidable

task for multicast routers, especially for those in the backbone networks.

Several aggregation schemes have been proposed, with aggregation either on a group-ID-centric or on an interface-centric way.

In a group ID-centric approach [31], groups with adjacent group IDs (e.g., multicast addresses) will be aggregated if their iif and oif sets match. This scheme is called "strict aggregation." Due to the geographical distribution of the group members, strict aggregation is seldom possible. To improve the chance of aggregation, "pseudostrict aggregation" and "leaky aggregation" are also proposed. In pseudostrict aggregation, if two or more groups have identical iif and oif sets, even if they are not adjacent, they can still be aggregated as long as there is no entry for intervening groups in the router. In the leaky scheme, the design goal is to restrict the number of MFT entries to approximate the number of high bandwidth groups. Thus, a low bandwidth group will be aggregated to the same MFT entry as a high bandwidth group even if their oif sets do not match. The price to pay is bandwidth wastage due to unnecessarily sending low bandwidth traffic on some links.

In [32], an interfacecentric aggregation scheme is proposed ([33] also mentioned such an idea). In this scheme, each interface of a multicast router has its own iif and oif filters. An incoming packet needs to pass the iif filter, then the oif filters on all other interfaces are checked independently to see whether this packet should be forwarded through those interfaces. Therefore, at each interface, a relatively large number of multicast groups can be aggregated. Due to current router technology that uses multiple parallel processors in the router and even one processor per interface, the separate installation of forwarding filters on each interface is not considered an extra processing burden.

Even if the above schemes can effectively aggregate ranges of multicast addresses, a naming mechanism is still needed for the aggregated addresses. In unicast, the aggregated addresses are represented by a single address with the same longest prefix of the aggregated addresses and a mask identifying the length of the prefix. This method works for unicast because unicast address allocation is topology based, so that one longest prefix entry usually represents a network. However, in multicast, not only addresses with the same prefix, but also other adjacent addresses may be aggregated. Therefore, it is necessary to name arbitrary intermeshed aggregations of multicast addresses. In [33], a naming method called AMA is proposed. With AMA, even nonadjacent ranges of multicast addresses can be represented by just one entry in the routing table, allowing efficient aggregation.

The aggregatability of multicast addresses is closely related to the address allocation schemes. In the BGMP/MASC scheme, multicast addresses are allocated hierarchically to different domains and a multicast distribution tree of a group is rooted at the domain that owns the address of the group. Therefore, for the downstream routers receiving traffic from the root domain, the multicast addresses can be aggregated. This helps to aggregate forwarding states at incoming interfaces.

A harder problem is related to the receiver topology and the aggregation at the outgoing interfaces. Usually, the multicast address of a group is allocated before the beginning of the session and the receiver topology is not totally known at that time. Even if the receivers' locations are known, assigning group addresses based on the receiver topology is still hard to do. There are only a few cases where aggregation at the outgoing interfaces is significant, such as group members all belonging to the same ISP and there are many such groups. However, considering the Internet multicast paradigm, such cases are unlikely.

Therefore, there seems to be no satisfactory aggregation scheme. On the other hand, whether aggregation is really needed is debatable. A calculation is performed in [31] on the size of the MFT. It is argued that high-end routers will soon have enough high-speed forwarding table memory to satisfy the needs of IPv4. However, there is another need to aggregate multicast addresses—to improve the utility of the multicast forwarding entry [31]. Usually, a multicast routing protocol uses an underlying unicast protocol to provide routing information and a forwarding entry is installed to cache the information to speed up forwarding, but for very short-lived multicast sessions or low bandwidth sessions, it may not be worth it to install a forwarding entry. It is believed that entry utility will increase in proportion to the corresponding group's bandwidth. The above leaky aggregation scheme aggregates states for low bandwidth groups to achieve higher entry utility.

2) Eliminating Multicast States on Nonbranching Nodes: From the above introduction to multicast address aggregation schemes, we can see that it is very hard to aggregate multicast addresses. To avoid the burden of installing a multicast forwarding entry for each group on each on-tree router, schemes are proposed to install MFTs only on branching nodes of a multicast distribution tree. The philosophy here is that the nonbranching nodes only deliver multicast traffic in a unicast way and so multicast states are not necessary.

a) Dynamic-tunnel multicast: The motivation behind DTM [34] is the observation that many locally dense groups will become sparse in the backbone, as illustrated in Fig. 7. On the distribution trees of these groups, there will be some long unbranched paths. Routers on such unbranched paths will contain so-called "unimulticast forwarding state," which refers to a forwarding state with only one immediate downstream receiver and without local group member—these routers are referred to as "unimulticast routers." The unimulticast forwarding states can be eliminated by setting up a unicast tunnel between the endpoints of the unbranched path and the multicast datagrams are encapsulated and forwarded in a unicast fashion along the unbranched path.

The tunnel is dynamic. Both routing and group membership changes may change the tunnels on a group's distribution tree. The dynamic tunnel is maintained by periodical "request" messages sent from its downstream end point. To limit

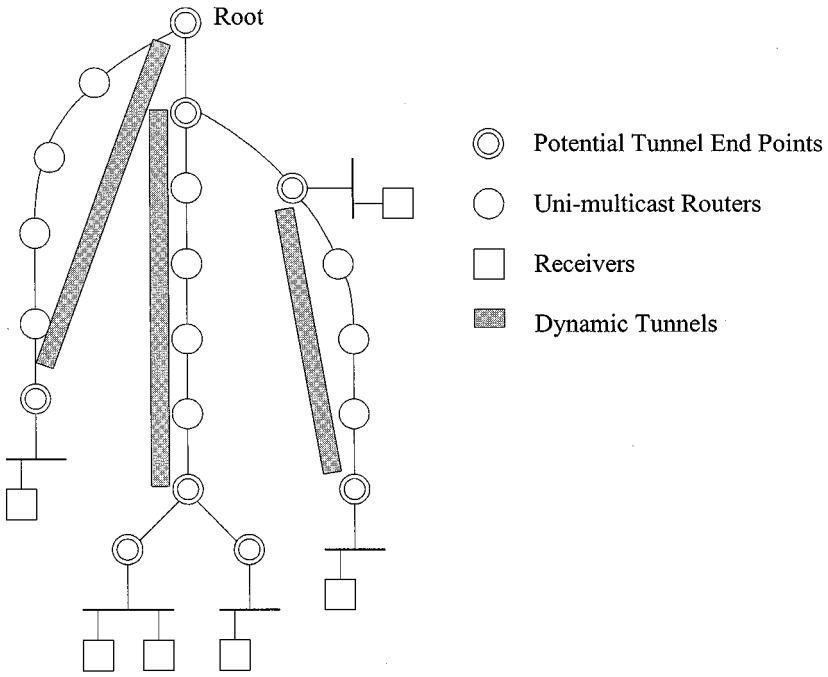

Fig. 7. Illustration of sparse multicast group and DTM.

the control overhead of dynamic tunnels, a minimum tunnel length is observed to avoid tunnels which are too short. The encapsulation of the tunnels is another source of overhead and may introduce packet fragmentation.

b) Recursive unicast tree: REUNITE [35] is a scheme with the same design goal as DTM—to accommodate sparse groups having a large number of nonbranching nodes, but with different approaches. While in DTM, the nonbranching nodes are bypassed by the dynamic tunnels, REUNITE uses recursive unicast to implement multicast service, eliminating the need of maintaining multicast states on the data plane[6] in the nonbranching nodes.

In REUNITE, a special node serves as the root of the multicast distribution tree. In single source sessions, the root is usually the source node. The group is identified by the root's IP address and a specified root port number, written as ⟨root_addr, root_port⟩.

An example of REUNITE packet forwarding is illustrated in Fig. 8 [35]. In this figure, S is the source and the root, while R1, R2, and R3 are the receivers. R1 first joins the multicast session by sending a Join message to S. S then installs an MFT entry recording R1 and sends unicast packets to R1 with the "root_port" as the source port number. Later, R3 joins the group also by sending a Join message to S. The Join message will be intercepted by the branching node N3. N3 installs an MFT entry and will replicate to R3 each passing packet sent from the specified port of S to R1. In this example, R1 is said to join at S and R3 at N3. R2 joins later at N4 in the same way. Therefore, MFT entries are only maintained in the root and the branching nodes N3 and N4. Nonbranching nodes N1 and N2 are relieved of such burden. As a result, there is only one

[6]An MFT on the data plane needs to be looked up when data packets arrive. In contrast, the state maintained on the control plane is only invoked by control messages.

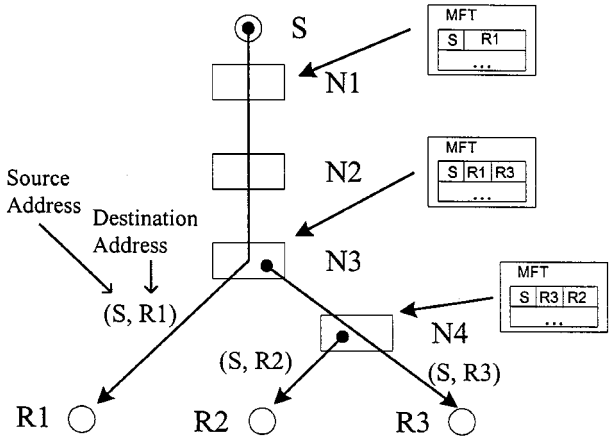

Fig. 8. Example of packet forwarding in REUNITE (adapted from [35]).

multicast forwarding state in the distribution tree for each receiver, no class D address is used at all in REUNITE and the data plane MFT is maintained only at branching nodes. Therefore, REUNITE scales well for large number of sparse multicast groups.

In REUNITE, the MFT entries are refreshed by periodical Join messages from the receivers and Tree messages from the root. The Join message traverses the reverse shortest path from a receiver to the root, while the Tree message and data packets traverse the shortest paths from the root to the receivers. Therefore, paths in the two directions may not be the same when asymmetric routes exist. In this case, data distribution in REUNITE is more efficient than other multicast routing protocols that use RPF.

However, REUNITE has several disadvantages. From the aspect of protocol dynamics, a member departure may affect

other receivers and cause restructuring of the multicast distribution tree. In addition, there will be duplicated packets during tree restructuring. Another major disadvantage is the additional processing overhead of data forwarding. Since a multicast group is identified by ⟨root_addr, root_port⟩ pair, routers cannot distinguish a multicast packet from an unicast one until they check the source port number and perform a lookup using the ⟨root_addr, root_port⟩ pair in the MFT.

3) Xcast/ SGM Approaches: Although DTM and REUNITE eliminate the multicast states on nonbranching nodes, they still have multicast states on branching nodes. The basic purpose of multicast is to avoid delivering multiple copies of packets on the same physical link or sending the same packet many times. Maintaining multicast states on routers is not the only method to achieve these goals. The alternative is a connectionless approach whereby multicast destination information is carried in each packet that is routed by the underlying unicast routing protocols and no multicast state is needed in the network. Several such connectionless multicast protocols are proposed, such as SGM [36], [37], Somecast [38], CLM [39], and MDO6 [40]. As in [41], we call them xcast/sgm (xcast: explict multicast, with an explicit listing of destinations).[7] In the following section, we introduce SGM to illustrate some basic operations of the xcast/sgm approaches.

a) Small group multicast: SGM is proposed to support very large number of small groups. An example of SGM is a videoconference involving three or four parties. This is also called "narrowcast" multicast.

In SGM, the sender is assumed to know all the receivers. Usually, a higher level mechanism is needed to organize the participants and distribute membership information, e.g., one can advertise a multicast session on a webpage or organize a video conference by e-mail. The packet sent from the sender contains the list of all receivers. Since the group is small, this will not introduce too much overhead. When an SGM-aware router receives such a packet, it will forward the packet to each next hop router that has downstream receivers and the forwarded packets are possibly modified to have receiver lists containing only the relevant downstream receivers. The last hop packet to a receiver usually contains only one destination and can either be an ordinary unicast packet or an SGM packet with only one entry in the receiver list.

Two schemes are proposed to support the new SGM packet type. The first one defines SGM as a new L3[8] packet type. In the L2 header, the new network protocol is specified, while in the L3 header, the source address is the sender's IP address and the destination address becomes a list of receivers' addresses. The second scheme, which is illustrated in Fig. 9 [37], defines SGM in the level between L3 and L4. In this scheme, the L2 header will still specify that the next level header is IP and the IP header has the sender's address as the source address, the next hop router's address as the destination address and the "SGM" as the next level protocol. The receiver list is contained at the SGM level.

[7]Please note that in [41], REUNITE is also included in the xcast/sgm proposals.

[8]Lx refers to Layer x in the seven-layer ISO/OSI model.

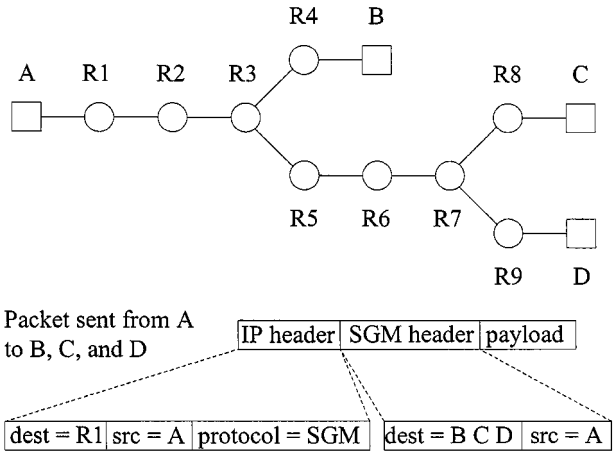

Fig. 9. Implementing SGM between L3 and L4.

Besides scaling to large number of concurrent small multicast groups, SGM has the advantage of supporting "subcast." Subcast refers to sending to a subset of multicast group members. Subcast is essential in reliable multicast because sometimes retransmission to a subset of group members is needed. In SGM, subcast can be easily accomplished by including only a subset of the receivers in the receiver list.

b) Discussions: By using the underlying unicast routing protocols to deliver multicast traffic, the xcast/sgm protocols avoid the conventional multicast state and signaling burden in the network. Moreover, mechanisms implementing QoS routing, traffic engineering, and policy routing designed for unicast routing may be reused. The tradeoff is the overhead of carrying and processing extra information in the packet header. Therefore, the xcast/sgm protocols only support multicast groups with limited members and are complementary but not expected to replace existing IP multicast protocols.

In the traditional Internet multicast architecture, IGMP is used in the first hop between hosts and multicast routers and various multicast routing protocols are used between routers. This isolation facilitates the development and choice of different multicast routing protocols and allows the multicast routers to assume the routing responsibility. In the xcast/sgm architecture, the first hop is no longer distinguished from other hops and the complexity is moved into the end hosts. Since routers are released from maintaining multicast states, many concurrent multicast sessions can be supported using unicast routing and state aggregation. For routers not in the core networks, maintaining multicast states may be a burden and so some xcast/sgm protocols, such as CLM, give routers the flexibility of trading off between link bandwidth consumption, per session state and signaling, and per packet processing [39].

4) End-System Multicast: ESM [42] is another multicast approach for small and sparse groups. In ESM, "end systems implement all multicast related functionality including membership management and packet replication." The end system may either be a host or a network proxy. Although ESM and xcast/sgm both shift multicast support from routers to end systems, ESM is a genuine higher layer protocol run-

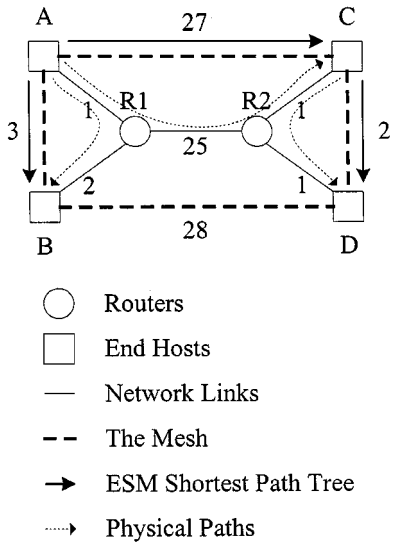

Fig. 10. Example of ESM.

ning on top of IP: unlike xcast/sgm, where routers are responsible for such functions as packet duplication, in ESM the network only provides unicast service.

To perform ESM, a protocol named "Narada" is proposed in [42] for end systems to self-organize into an overlay structure in a distributed way. Every member of a group maintains the complete group membership list and exchanges membership information with neighbors. A mesh topology is constructed by Narada among group members and the per source distribution tree is constructed in the same way as DVMRP by running a distance vector protocol on top of the mesh. That is, each group member receives packets only from the neighbor on the shortest path to the source and then forward the packet to all its neighbors who use it as the next hop to reach the source. Fig. 10 gives an example of ESM. In this figure, link costs are shown besides the links. Dashed lines form the mesh topology. For the group in which A is the sender and B, C and D are receivers, Narada constructs the shortest path tree between these end hosts, as shown in Fig. 10. According to the tree structure, A sends packets to B and C and C forward them to D. The physical paths delivering the packets are depicted as dashed arrows in Fig. 10.

To join a multicast group, a host needs to obtain a list of group members (the list is not necessarily complete) by an out-of-band bootstrap mechanism.

The ESM scheme will introduce performance penalty such as duplicate packets and larger end-to-end delay. The penalty will be small if the distribution tree is well organized and the number of group members is small.

Yallcast [43] is an alternative ESM approach. It allows a group of endhosts to autoconfigure into a tunneled topology. It builds a tunneled shared tree for content distribution and a tunneled mesh for broadcasting content and control information. The difference between Narada and Yallcast is that Yallcast uses the tree-first approach and the tree is not a subset of the mesh. The group ID in the Yallcast protocol can be encoded as an URL such as, yallcast://rendezvous.host.name:port/group.name.

It is worth noting that although one or more rendezvous hosts are associated to a group in Yallcast, they are not attached to the tree and mesh. They have the information of all the existing group members and provide this information to joining members. This is a centralized approach compared with Narada's fully distributed approach.

C. Summary of Scalable Multicast Routing

In this section, we study two aspects of the scalability of multicast routing: one is the scalability for large groups and the other is the scalability for large number of concurrent groups. Sparse mode multicast routing protocols like PIM-SM and CBT, interdomain multicast routing solutions such as MBGP/PIM-SM/MSDP and BGMP/MASC, and alternative multicast service models including SSM/EXPRESS and SM are proposed to scale multicast routing to large groups where group members are scattered into large areas. The other challenge is to deal with the large number of concurrent groups, since multicast addresses and forwarding states are very difficult to aggregate. Some attempt to aggregate multicast addresses/states directly, while others try to eliminate multicast states on nonbranching nodes of the multicast distribution tree (DTM and REUNITE) or try to shift some or all of the multicast support into endhosts (xcast/sgm and ESM).

Some of the newly proposed schemes change the "open group" model of traditional multicast to a "close group" model. The SSM protocols allow only one source to send to a certain group. IGMPv3 [7] gives receivers the ability to select senders. In some schemes, e.g., source-only multicast [44], the sender may require join requests from receivers to be delivered to itself or to some proxy nodes (for scalability reasons) to be approved. In the xcast/sgm protocols, the senders need to know all the receivers, while in end system approaches, a joining member needs to obtain an existing member list by an out-of-band bootstrap mechanism (Narada) or from a rendezvous host (Yallcast). Compared with the "open group" model, the "close group" model provides more security and facilitates multicast billing.

A problem of IP multicast is address allocation. Usually, each multicast group requires a worldwide unique address to distinguish traffic from other groups. Today the address is allocated randomly with some knowledge of addresses already used. This scheme works well only when multicast is not prevalent. MASC is proposed to allocate multicast addresses hierarchically according to the network topology. This approach provides the required uniqueness in address allocation. The "channel" service models like EXPRESS and SM use a (host_addr, group_addr) pair to identify a multicast session. Some other schemes like REUNITE use a (host_addr, host_port) pair for this purpose. Most of the xcast/sgm protocols do not use multicast addresses [41]. The tradeoff is the increased header processing per packet in these protocols. In the end system approaches, all the multicast functionality is shifted into the end hosts and group addresses are used locally, so the address allocation is easy. Each host needs only maintain states of groups it has joined. However, the multi-

cast distribution trees must be established by end systems, resulting in heavy signaling traffic and degraded performance.

III. Reliable Multicast

While network layer multicast [4] provides best effort, unreliable, one-to-many, or many-to-many delivery, some applications have requirements beyond this. Bulk-data transfers, such as file distribution and web cache updates, require error-free delivery of data, but they can usually tolerate relatively large delay and delay jitter. On the other hand, real-time streaming applications like video/audio multicasting and interactive applications such as video conferencing require strict delay and/or delay jitter bounds, but they can tolerate a certain level of packet loss. Furthermore, some applications, such as shared whiteboards, distributed games, and DIS [45], require both error-free and real-time delivery. Other applications, such as those in distributed systems [46], require not only error-free, but also ordered delivery.

It is believed that a "one-size-fits-all" transport protocol for multicast is unlikely [47]. Therefore, many multicast transport protocols are proposed to meet the requirements of various applications. In particular, many reliable multicast protocols have been proposed in the past decade [21], [48]–[50]. In this section, we will focus on various schemes and mechanisms dealing with the challenges and problems in the design of RMPs. Major protocols introduced will be summarized in a table at the end of this section.

A. Definition of Reliability

To introduce the work on reliable multicast, we first need to give a definition of "reliability." We define broad sense "reliability," which comprises the following three aspects.

1) Error-Free Delivery: Error-free delivery refers to delivering all data to all receivers eventually. This is the narrow sense "reliability" used in Internet multicast. "All receivers" may or may not include late-join or temporarily partitioned receivers. If they are included, at least the sender needs to keep all transmitted data during the multicast session [51].

Based on the narrow sense definition, reliability can be further classified into semireliability (or quasi-reliability [52]), time-bounded reliability, and total reliability [53]. When semireliability applies, the transport layer may retransmit a lost packet or use error correction coding to provide an acceptable loss probability to applications, but does not guarantee totally error-free delivery of all data to all receivers, as is done in total reliability. Semireliability is often used by loss-tolerant real-time multicast applications. Time-bounded reliability [54] is also defined for real-time applications. If a packet is not received after a time threshold, it will not be recovered. As pointed out in [53], time-bounded delivery usually implies a semireliable protocol, but the converse does not necessarily hold.

2) Atomicity: Atomicity guarantees either all of the applications/processors or none of them receive a message [46]. It can be achieved, e.g., by ensuring that "once one group member (or a majority) delivers a message, the rest of the group must deliver the message by some time" [55].

3) Ordering: There are different levels of ordering defined, such as single source ordering, multiple source ordering, and multiple group ordering [55]. They are listed by increasing strength of ordering guarantees. There is also a causal ordering defined in [46], which keeps the time precedence relations between multicast messages.

There are other definitions of multicast reliability (see [56] for a list). In this section, we will focus on multicast transport layer issues. Deciding the number of reliability functionalities to be provided at the transport layer and the number to be provided by higher layers is an application-specific issue. In this section, we focus primarily on narrow sense "reliability"—error-free delivery, which is required by most reliable multicast applications on the Internet. On the other hand, atomicity and higher levels of ordering are often achieved at the expense of larger delivering latency and higher signaling and processing overhead. They are often required in distributed systems applications [46], but deemed irrelevant in Internet reliable multicast transport scenarios [57]. The argument is that they can be dealt with easily at higher layers [58], [59]. Therefore, we will use the narrow sense "reliability" hereafter.

B. Application-Level Framing

Before proceeding, we will briefly discuss the concept of ALF [26], [60]. ALF breaks data into ADUs for transmission. Higher performance is achieved by allowing applications to process received ADUs immediately even if they are out of order. Later, it is observed that the ADU is more meaningful in multicast than the numbered packets used in unicast protocols [26]. For example, the packet sequence number may be ambiguous for a later-join receiver in a multicast session since it does not know the beginning of the session. However, if ALF is used, the later-join receiver can ask for retransmission of missed application units from the sender or other nodes (servers, receivers, or routers), since ALF decouples the sender from retransmission. ALF is also tightly related with the LWS architecture [61], [62]. To meet the requirements of diverse applications, ALF advocates leaving as much functionality and flexibility as possible to applications [26]. Example protocols using the ALF concept include SRM [26] and RTP [25]. They provide very thin transport layers to incorporate applications with various requirements.

However, in some applications, it may be difficult to divide data into meaningful ADUs suitable for transmission. For example, in software distribution, sometimes the ADU is the whole software package, which is often too large to fit into a packet. Another example is interactive voice application, where several small ADUs are often put into one packet for higher transmission efficiency. In the following discussions, we will still use the conventional "packet" as the data unit of the transport layer, but the description of various schemes is also applicable to schemes using ADUs.

C. ARQ and FEC

IP multicast provides only unreliable and best effort delivery at the network layer. For those applications that require

reliable multicast, two mechanisms can be used at the transport layer, namely, ARQ and FEC.

1) Automatic Repeat Request: ARQ is a "retransmission on demand" mechanism, where the sender is alerted to packet losses through feedback from receivers and lost packets will be retransmitted by either the sender or other nodes. An ARQ scheme can either be sender- or receiver-initiated. In a sender-initiated scheme, the sender maintains state information of receivers and detects packet losses [49]. Receivers need to acknowledge every received packet by ACK to the sender. If the sender does not receive the ACK for a packet after time out, it will assume that the packet is lost and a retransmission or a congestion avoidance mechanism will be triggered. In a receiver-initiated scheme, receivers have the responsibility of detecting losses, e.g., by observing gaps in received packets. After a loss is detected, a NACK will be issued to report the loss and request retransmission. Usually, in multicast transmission, receiver-initiated schemes are more scalable than sender-initiated schemes [49], since the burden of maintaining reliability is distributed among receivers and NACKs are only issued when packet losses occur.

As feedback mechanisms, ACK and NACK differ in several ways. ACK can be used for basically two purposes: one is to confirm reception and the other is to detect losses and prevent congestion collapse. If the sender has received ACKs of a packet from all receivers, it can remove the packet from its memory [50], [63]. The sender can also use timeout of ACKs as indication of loss. These two functions of ACKs can be used together or separately [50]. Protocols using ACKs suffer from the problem of ACK implosion—which will be discussed later and usually require relatively large latency to detect packet losses compared with NACK protocols. Furthermore, knowledge of the receiver set is required by ACK protocols, which is difficult to maintain for large receiver population. On the other hand, NACK serves as a quick explicit indication of packet loss. It can be used to avoid ACK implosion, since it is issued only when packet loss occurs and only from relevant receivers. However, NACK alone does not allow the sender to discard a buffered packet and NACK implosion may occur. ACK and NACK are often used together, where NACKs are issued when packet losses occur and ACKs are sent back periodically to confirm reception and solicit further transmission. This scheme is called NAPP. Another type of ACK, used in reliable RMTP [51] and RMTP-II [54], is the periodical ACK with a bitmap, which indicates the reception status of recent packets for the receiver. This scheme is receiver-initiated because receivers detect the loss, but loss information is carried in the bitmap of periodical ACKs explicitly.

Usually, a hierarchical mechanism, which organizes group members into a logical tree structure, can improve the performance of ACK and NACK protocols. This mechanism can alleviate feedback implosion, achieve efficient scoped retransmission[9] and provide timely delivery. The LBRM [64] is a tree-based NACK protocol, RMTP, RMTP-II, Lorax [63], and TRAM [65] are tree-based ACK protocols (RMTP-II has an NACK option), and the TMTP [66] is a tree-based NAPP protocol.

2) Forward Error Correction: FEC provides reliability by introducing redundant information in the transmission. Usually, pure FEC uses error control coding to detect and correct corrupted data at the receivers without requiring retransmission. It trades processing power and bandwidth[10] for higher reliability and smaller recovery latency.

Here, we study FEC at the multicast transport layer, which is not the same as that used in the link layer. Usually, link layer FEC detects and corrects bit errors in transmission by using various error control coding and interleaving schemes [67]–[69]. However, transport layer FEC operates on the packet (or message) level. The transport layer will know from lower layers whether a packet is successfully received or lost. Therefore, FEC at the transport layer can take advantage of erasure[11] correcting codes [70]–[72] to deal with missing packets with known packet numbers.

The RSE code [70] is the commonly used erasure correcting code in multicast FEC. The RSE code encodes a block of k packets into an n-packet codeword, with $h = n-k$ redundant (parity) packets. The code is systematic, which means that the original k packets are included in the codeword in clear form. Receiving any k out of the n packets in the codeword is enough for decoding the original k packets, as shown in Fig. 11.

However, the optimal amount of redundancy in FEC usually is difficult to set *a priori* due to heterogeneous loss probabilities in a multicast group and the burstiness of losses. In addition, FEC alone cannot guarantee total reliability. Therefore, in reliable multicast schemes, FEC is often used together with ARQ mechanisms. There are basically two integrated FEC/ARQ (hybrid ARQ) approaches [73], namely, the layered approach [74], [75] and the integrated approach [75]. In the layered approach, FEC is transparent to the ARQ-based mechanism. It is used to significantly reduce transmission errors seen by the ARQ protocols. In the integrated approach, k data packets are transmitted without or with part of the parity packets. A receiver will request more parity packets when it cannot successfully receive k packets from the original transmission.

In reliable multicast, FEC is mainly used to deal with independent losses, reduce and simplify feedback (for losses) and retransmissions, and provide timely delivery. Independent losses will require the sender to retransmit for each loss separately. This will lead to scalability problems since the number of independent losses increases with the number of

[9]Scoped retransmission limits retransmissions to a predefined region of the network.

[10]Bandwidth here refers to that consumed by redundant data. However, FEC helps to reduce feedback and retransmission and often saves bandwidth when error occurs.

[11]At the packet level, an erasure is a corrupted packet with known packet number [71].

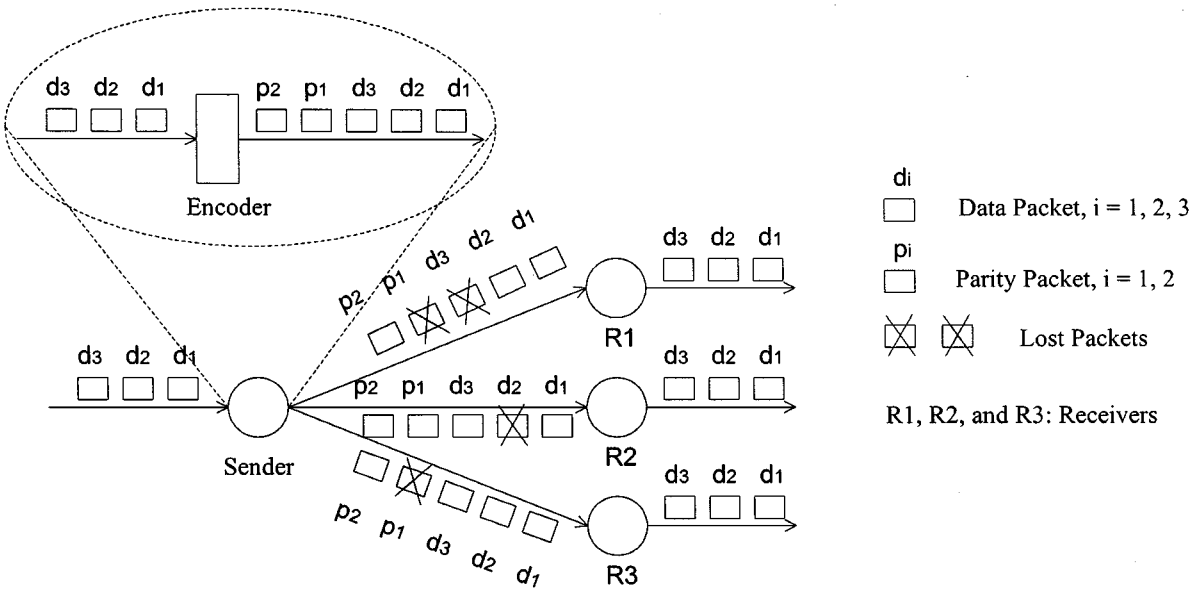

Fig. 11. Error recovery using RSE with parameters $n = 5$ and $k = 3$.

Table 1
Comparison Between Pure FEC and ARQ in Reliable Multicast Protocols

FEC	ARQ
Suitable for large groups with large Round-Trip Times (RTTs), or when the feedback channel is unavailable	Suitable for small groups with feedback channel
Suitable for networks with homogeneous loss probability	Suitable for networks with heterogeneous loss probability
Efficient in overcoming independent loss	Efficient in overcoming shared loss
Suitable for real-time interactive applications	Suitable for non-interactive applications
Only provides semi-reliability	Provides total reliability

receivers. However, using the RSE code, the sender only needs to transmit a single stream of repair packets for a block of transmitted packets, with the number of repair packets equal to the maximum number of lost packets among all receivers. This also simplifies the feedback mechanism. The sender only needs to know the maximum number of lost packets among all receivers instead of which packets are lost for which receivers and the maximum number is easier to handle in suppression or aggregation schemes (see Section III-D). Furthermore, as mentioned above, FEC can significantly reduce NACKs, thus, alleviating the problem of feedback implosion. For those real-time applications requiring reliable transmission, FEC can be used to reduce the expected time of reliable receipt of data or provide bounded loss for hard real-time deadlines [76]. For networks where the feedback channel is expensive or unavailable, e.g., satellite networks, FEC can be used to provide open-loop control to achieve semireliable delivery. Table 1 compares the major differences between using pure FEC and ARQ in reliable multicast protocols [77].

D. Feedback Implosion

Reliable multicast protocols often suffer from the feedback implosion problem [52], [78] in which a large amount of feedback from many receivers is sent almost synchronously to the sender. This will lead to network congestion and overwhelm the sender.

In a basic ACK scheme, every receiver sends ACK directly to the sender for each packet received. These ACKs will provide the sender feedback information for both error control and congestion control. However, this scheme will lead to ACK implosion when there is a large number of receivers. Basically, this problem can be alleviated by hierarchical ACK approaches or NACK-based approaches.

1) Hierarchical ACKs: Hierarchical ACK approaches often use a tree structure to aggregate ACKs from receivers. Usually, the sender (or a top node in RMTP-II [54]) is at the root of the tree and leaf nodes are receivers. The interior nodes of the tree can be receivers, special servers, or multicast routers. Since nodes in the tree only send ACKs to their immediate parent and the fan-out of the tree is usually bounded, the ACK implosion problem is avoided.

There are basically two ways to aggregate ACKs along a tree from receivers to the sender (or top node) [54]. One is optimistic aggregation (also known as hierarchical acknowledgment [50], [63]) and the other is pessimistic aggregation. In optimistic aggregation, an interior node acknowledges packets as soon as it has received them. It is then responsible for reliably delivering the packets to its children. In pessimistic aggregation, an interior node sends ACKs to its parent only after it has received ACKs from all of its children and the sender advertises its received highest sequence number to confirm packets received by all the members. Optimistic aggregation requires less buffer space at the sender and at some interior nodes since they discard a buffered packet as soon as their immediate children acknowledge the packet. On the other hand, the pessimistic approach is more robust. A node can always receive packets from its ancestor node when its parent dies.

ACKs can be sent every time a packet is received, or periodically. The former method will generate larger volume of traffic compared with the latter and is usually unnecessary. In the periodical scheme, an ACK from a node typically carries the lowest sequence number not yet received and a bitmap indicating the packet reception status at the node [51], [79]. The bitmap can be used in selective retransmissions.

Hierarchical ACKs can provide different types of information to the sender [53]. An interior node can send upstream an ACK when all its immediate children or all the downstream children have ACKed or send an ACK after a certain time interval and list children that have and have not ACKed. Furthermore, an ACK can also carry group membership information, such as a counter of the number of group members, or congestion control information, such as estimated RTT and loss probability. The information carried in ACKs depends on the requirement of the protocol and the ACK aggregation mechanism should be adjusted accordingly.

Conceptually, the ACK tree built at the transport layer is a logical tree (or a control tree), which is not the same as the multicast distribution tree (routing tree) at the network layer. However, the performance of hierarchical ACK schemes greatly depends on the congruence of the ACK tree with the underlying routing tree. The greatest challenge is how to construct and maintain a good ACK tree in a scalable way [53]. In RMTP-II, the ACK tree is manually configured and maintained by regular traffic (periodical ACKs, heartbeat packets) in the tree. This method does not scale and cannot deal with rapid membership and network topology changes. TMTP, Lorax, and TRAM use ERS to construct the ACK tree automatically, but the constructed trees are often suboptimal and less fault-tolerant and robust to topological changes [80].

To form an optimal ACK tree automatically, routing information is usually needed. Therefore, some router-assisted schemes are proposed to ensure the congruence of the ACK tree and the routing tree, using routing information available in the routers. We will discuss router-assisted schemes in Section III-F.

2) NACK-Based Approaches: ACK implosion can also be alleviated by NACK-based approaches, since NACKs are sent less frequently and only from relevant receivers. However, mechanisms are still needed to prevent NACK implosion, especially when loss probability is high, losses are correlated, and the multicast group is large. Unlike ACKs, which are usually necessary and the goal is to aggregate them, NACKs can be suppressed, i.e., NACK messages are often redundant and one NACK can represent many others carrying similar information.

Commonly used NACK suppression mechanisms are timer-based [26], [81], [82]. A receiver will set up a timer for a NACK and send it out after the timer expires. If a NACK containing the same information (e.g., reporting the same lost packet) or a retransmission is received before the timer expires, the timer is reset and the NACK is cancelled. A NACK can be multicast to the group to suppress others or, alternatively, unicast to the sender (or its parent), thereby prompting the sender (or its parent) to multicast a confirm message (the received NACK or required repairs) to the group for suppression [57]. The latter approach is used when multicasting from the receivers is expensive or impossible [83].

The timer can be set randomly or deterministically. XTP [24], [84] and SRM [26] use random timers with uniformly distributed values. In SRM, the distribution interval is set based on the estimated one-way delay between the receiver and the sender. The performance of this mechanism depends largely on the accuracy of the estimation. In [85] and [86], an exponentially distributed random timer is proposed and it is claimed to have lower feedback latency and better feedback suppression compared with the uniformly distributed timer. On the other hand, in networks with delay guarantees, an optimal deterministic timer [82], set according to the DTRM algorithm, can be used to ensure that only one NACK is fed back for a lost packet.

NACK-based schemes embody the receiver-initiated principle in which the receiver assumes the responsibility of loss detection. However, schemes using only NACK cannot provide reliable communication with finite memory [50]. This problem can be solved using the NAPP scheme.

The timer-based NACK suppression mechanism is first proposed in the LAN environment [81]. It performs poorly in wide area networks. It needs to multicast among members for suppression and RTT estimation. Alternatively, NACK suppression can be combined with tree-based schemes. In the combined schemes, NACK suppression is only performed in the local groups of a tree and, therefore, works as good as in a LAN environment. Tree-NAPP is such a protocol and has been implemented in TMTP, which is shown to perform better than other flat NACK-based protocols [50]. Another example is protocol L1 described in [48], where NACK suppression is performed in the local "stub domains" [87] and at most one NACK is expected from a "stub domain" for each loss.

3) Other Approaches: In the above discussion, we introduced major approaches for the feedback implosion problem. In hierarchical ACK schemes, ACKs are aggregated using a tree structure. In timer-based NACK suppression schemes, NACKs are suppressed based on random or deterministic timers. ACKs and NACKs can be used together, as in the

NAPP approach. For scalability reasons, NACK suppression is best used locally, e.g., in a LAN or in local groups of a tree structure. There are still other approaches proposed for alleviating the feedback implosion problem, including probabilistic querying schemes, the representative-based scheme and ring-based schemes. The first two approaches, namely, probabilistic querying and representative-based schemes, are more suited for multicast flow and congestion control. The basic idea is decoupling feedback for flow and congestion control from feedback for error control. This will be studied further in the next section.

In the basic probabilistic querying scheme suggested in the QMTP [52], a receiver that wants to send a feedback will do so with a given probability b. A sender will request feedback again if it has not received it after a timeout period. This reduces the total number of feedback messages and feedback implosion is alleviated. However, this scheme usually cannot provide accurate feedback information and the best feedback probability b is difficult to determine.

Another probability-based feedback control scheme is proposed for congestion control of multicast video [88]. In this scheme, a receiver will send feedback only when its random key matches the sender's key with the specified significant bits. The sender reduces the number of significant bits in each round to let more receivers respond and estimates the size of the group.

In [89] and [90], a representative-based scheme with feedback suppression is proposed. In this scheme, the sender dynamically selects a small set of receivers to represent the most congested subtrees of the multicast group, based on feedback messages (ACKs and NACKs) from the receivers. Selected representatives send feedback immediately, while other receivers wait for a random time larger than zero and send feedback if similar feedback has not been received. It is expected that the representatives send immediate feedback on behalf of the most congested subtrees.

Token-ring-based multicast protocols, such as RBP [22] and RMP [91], alleviate feedback implosion by allowing only the token site to multicast ACKs. The ACKs are used to confirm reception, guarantee atomicity and ordering, and help other receivers detect losses. Other receivers send NACKs to the token site for retransmission. The responsibility of being the token site is rotated among the receivers.

E. Retransmission Strategies

In ARQ and combined FEC/ARQ schemes, on detecting a loss, retransmission is scheduled to recover the loss. The sender is the ultimate responder of retransmission requests. However, if the sender is required to respond to every retransmission request, it will be overwhelmed when there are many receivers and for those receivers far away from the sender, the recovery latency is relatively large. SRM, in its basic version, allows any receiver to respond to retransmission requests by multicasting globally. However, this approach will introduce unnecessary retransmissions and duplications, especially when only a small fraction of the receivers asks for retransmission persistently (the "crying baby problem" [64]).

1) Local Recovery: To achieve scalable retransmission in reliable multicast, the burden of retransmission is often distributed among the sender and other nodes, called repair servers, in the group. The repair servers could be common receivers [26], [92], designated servers representing local groups [51], [54], [66], [93], logging servers [64], servers collocated with multicast routers [48], [83], or even active routers [80]. A commonly used strategy is "local recovery," where the repair server closest to the receivers requesting retransmissions will respond. Local recovery enjoys the following advantages.

1) *Avoiding unnecessary bandwidth usage and packet processing:* Receivers behind a congested link can be treated individually and locally, without overloading the whole group with retransmissions. Unnecessary bandwidth usage and packet processing outside the local area can be avoided.
2) *Smaller recovery latency:* Since losses are usually recovered by nearby repair servers instead of the sender which may be far away, the expected recovery latency will be significantly reduced.
3) *Distributing recovery burden among network entities:* In local recovery, besides the sender, repair servers share the responsibility of retransmission. This relieves the sender from recovering for every lost packet and avoids the single point of failure.

Local recovery is often performed with a hierarchical structure. In RMTP and RMTP-II, repair servers are called DRs. They are organized into a logical tree with common receivers as leaves of the tree and perform acknowledgment processing and retransmission for their local subtrees. The DR is treated as an "interior node" [54] in RMTP-II, which can be operated by network managers and offers network management tool for control purposes. The repair server is called DM and GC in TMTP and LGC [93], respectively. The two protocols use very similar tree structures. DM is a representative of its subnet or domain and GC is responsible for its local group, which contains receivers in close proximity. The DMs or GCs are then organized into a logical tree. LBRM uses logging servers as repair servers. The major difference between logging server and other types of repair servers is that a logging server logs the entire data during the whole multicast session for error recovery and "later comers" [48].

The problem is determining where to put the repair servers. Usually, a repair server is put on each local site, which could be a subnet, a group of receivers close to each other and/or share the same loss, or even a single host [64]. For the current MBone, it is found that losses are most likely to occur at "tail links"—links between the backbones and stub domains [94]. Therefore, one strategy is to put repair servers at the edge of backbones just above "tail links." Fig. 12 shows an example based on the Internet topology modeled in [87].

To perform local recovery, scoping mechanisms are needed to restrict both repair request and retransmission to a limited area. The following methods can be used.

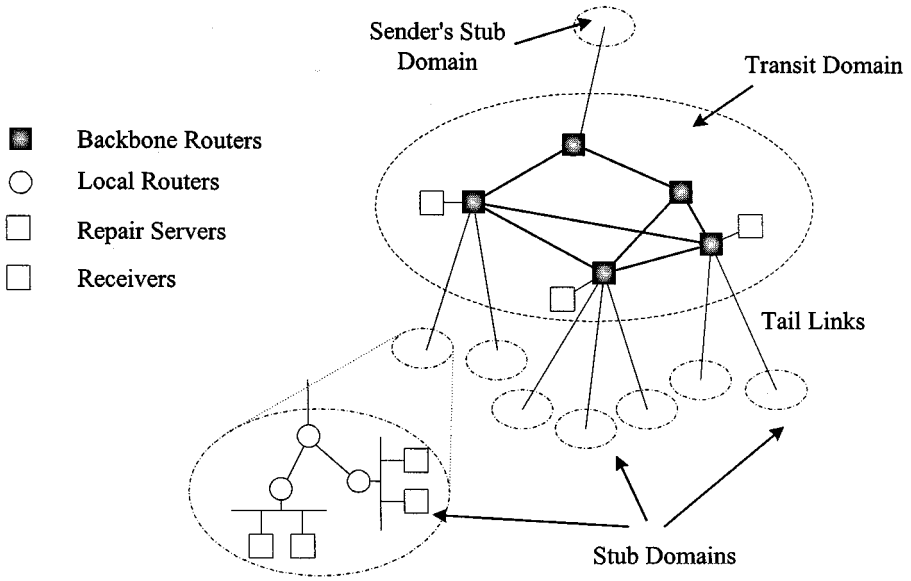

Fig. 12. Strategically located repair servers at the edge of the backbone, just above tail links.

1) *TTL-based scoping:* TTL-based scoping limits the scope of requests and repairs for local losses by the TTL field in the IP header. SRM sets an appropriate "hop count" in the TTL field [26] or assigns high thresholds to links at the boundaries of a naturally defined local-recovery neighborhood [95]. The efficiency of TTL-based scoping greatly depends on the network topology and mechanisms are needed to set appropriate TTL values.
2) *Administrative scoping:* A special region of the multicast address space—the "administratively scoped" address space is used to define the local recovery neighborhood.
3) *Multiple multicast groups:* Another method uses separate recovering groups for requests and repairs. One way is creating a group for every lost packet, but this requires fast group creation, sometimes high bandwidth for group control messages and potentially many group addresses. Another way is creating groups for loss regions instead of for individual losses. This is suitable for the case where stable loss neighborhoods exist [96], [97].
4) *Logical tree-based approaches:* Hierarchy-based (or tree-based) approaches organize receivers into a logical tree. A receiver asks for retransmission for a lost packet from its parent on the tree. If its parent has the required packet, it will remulticast it in the local subtree or unicast it to the receiver directly. Otherwise, the parent node will ask for recovery from its upstream node in the hierarchy. When the sender receives the request, it multicasts a repair to the entire group. Usually, tree-based approaches require nodes on the tree to keep information of their parent and children (if any) and the logical tree is expected to be congruent with the underlying routing tree.

   In a logical tree, when a retransmission is to be multicast in a subtree, IP encapsulation can be used as suggested in RMTP. A DR in RMTP encapsulates a multicast repair packet into a unicast packet with a new packet type—SUBTREE_MCAST and sends it to a nearby router. At the router, the packet is decapsulated and sent to the subtree rooted at the router as if it comes from the sender.
5) *Router-assisted approaches:* In router-assisted schemes (which will be described in detail later), scoped multicast can be performed more efficiently with the help of routers. These approaches use the underlying multicast routing tree instead of another logical tree for error recovery. Basically, routers (or their collocated servers) record information of NACKs and repairs and use the recorded information to intercept repairs and NACKs, suppress duplicated ones, and multicast repairs only to those interfaces on which corresponding NACKs are received.

2) Unicast or Multicast for Retransmission?: Retransmissions can be performed through either unicast or multicast. Unicast is usually used when the retransmission is for one or a few receivers, while multicast is used to send the same repair packets to many receivers and to suppress redundant repair requests. SRM advocates "multicast everything," and it multicasts repairs to the entire group or local groups. This approach can achieve higher reliability and robustness and is appropriate when losses are correlated and multicasting from receivers is possible and not expensive. On the other hand, RAMP [98], [99] suggests using unicast retransmission for reliable multicast on all-optical circuit-switched gigabit networks, since losses are rare and independent on these networks. Many reliable multicast schemes, such as LBRM and RMTP, employ a mechanism which dynamically selects multicast or unicast for retransmission.

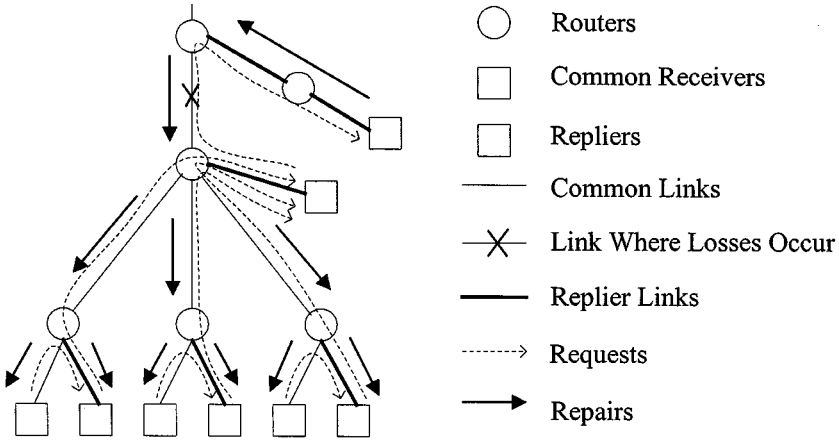

Fig. 13. Request and repair procedure in LMS.

F. Router Support for Reliable Multicast

Reliability is a transport layer issue and is, strictly speaking, not the responsibility of routers, which operate at the network layer. However, router support in the network can often help to improve the performance of reliable multicast protocols. Major proposed schemes using router support can be roughly divided into two categories. One uses minimal router support to direct retransmission request (e.g., NACK) to proper repliers,[12] thus, reducing feedback implosion. Examples are LMS [92], search party [100], and RMCM [101]. The other uses active routers (or active servers collocated with routers) for feedback aggregation and/or suppression and for local recovery. Examples include ARM [80], AER [83], PGM [102], and RMANP [103].

1) Router Support to Deliver Repair Requests: When a loss occurs on a link of the multicast distribution tree, all receivers downstream to the link will suffer losses and require retransmissions. Ideally, a node immediately upstream of the loss should retransmit the lost packet by multicasting it to the subtree downstream from the link where the loss occurs. Multiple NACKs for the same lost packet should be aggregated and restricted in the subtree experiencing the loss. LMS, search party, and RMCM try to mimic these operations with the assistance of multicast routers.

In LMS, each on-tree router selects one of its downstream links as the replier link, which leads to a replier for the subtree rooted at the router. The replier link is updated when replier state changes. Routers adjacent to the source will select the source link as the replier link. After detecting a loss, a receiver will send a retransmission request to a nearby router. On receiving a retransmission request, a router will redirect it to the replier link if it comes from other downstream links, or forward it to an upstream router if it comes from the replier link. Thus, only one retransmission request is sent upstream from a router for a certain lost packet and feedback implosion is reduced. The request and repair procedure is illustrated in Fig. 13 [92].

[12]A replier is a receiver assigned to retransmit as required.

Router supports in LMS include recording replier link at routers, inserting extra information into requests at TPs,[13] and performing directed multicast. The mechanisms used by LMS do not violate the clean layering principles, i.e., the routers do not manipulate any transport layer information and the end hosts do not need to know the topology [100].

LMS suffers from two problems. First, failure of repliers directly above or below a loss will disrupt recovery until the soft state expires. Second, LMS does not function well in bidirectional shared multicast routing trees. These two problems are addressed by search party and RMCM, respectively.

Search party redirects a retransmission request with randomcast—a service that forwards packets randomly inside a multicast distribution tree, instead of forwarding it to a replier link. This allows the burden of retransmission for a particular lossy link and the impact of one replier failure to be shared in the group. Moreover, routers need not maintain replier link information. The tradeoff is the increased recovery latency and overhead, since sometimes a request cannot go beyond the loss subtree[14] or is forwarded upstream unnecessarily.

In LMS, when a request comes from the replier link of a router, the router forward it upstream to the corresponding source. However, in protocols using a shared multicast routing tree, routers often do not know which interface leads to a particular source. When unidirectional shared tree is used, e.g., in PIM-SM, this can be solved by forwarding the packet to the core (RP) of the tree. The core will then unicast the request to the proper source and the source will respond with a repair packet tunneled to the TP [92]. However, when a bidirectional shared tree is used, such as in CBT, the core is no longer always upstream to a router with respect to a certain source and will not necessary know where the source is.

RMCM proposes a scheme to properly direct NACKs and replies in a bidirectional shared multicast routing tree. Each router selects a replier link for each of its interface. A source adds an IP option, called path_info, to data packets periodically. Each on-tree router then records the incoming interface of the packet in the option. When the packet reaches a

[13]The router redirecting a request to a replier is called the TP of that request.

[14]The tree below the loss link.

receiver, path_info is stored there. When a receiver sends a NACK, path_info will be included to let routers know from which interface the original data packet is received and to choose the replier link of that interface.[15]

2) Using Active Routers for Reliable Multicast: Protocols using a hierarchical structure, such as RMTP, RMTP-II, and TMTP, often find that it is difficult to construct an efficient logical tree for feedback control and retransmission automatically. This problem can be solved by using the underlying multicast routing tree directly, with routers actively taking part in the reliable multicast protocols. The tradeoff is the extra burden on the network.

a) ARM: ARM is a NACK-based scheme that uses active routers at strategic locations to suppress duplicated NACKs and perform local recovery. ARM assumes that the forward multicast paths correspond to reverse unicast paths.

Active routers on a multicast tree perform "best effort" caching of data passing through it. After detecting a loss, a receiver sends a NACK to the source. When the NACK arrives on an active router on the path, the router will check whether it has the requested data. If it does, a recovery is multicasted to the link from which the NACK arrives; otherwise, it forwards the NACK upstream. The active router also maintains a NACK record and a REPAIR record for each lost packet for a short amount of time to suppress duplicated NACKs for the same packet. The NACK record also contains a subscription bitmap to determine outgoing links to forward subsequent repairs.

b) AER and PGM: As in ARM, AER and PGM also provide active services in the network to enhance the performance of reliable multicast protocol, but with different approaches.

In AER and PGM, active services are provided by active servers collocated with routers at strategic locations in the network. These servers join the multicast group, cache data packets for loss recovery, suppress and aggregate NACKs and intercept downstream NACKs, and repair packets to save bandwidth. A signaling mechanism is adopted to establish the reverse path from the receivers to the source and to invoke or revoke active services.

c) RMANP: In ARM, AER, and PGM, active routers (servers) perform customized operations based on different packet types. This is similar to the operation model of the "active network" [104]. In an active network, routers perform customized computations on the messages flowing through them. RMANP is one such protocol that supports reliable multicast on an active network.

RMANP defines several types of capsules [104] for different operations, e.g., data capsule, retransmission capsule, and NACK capsule. Different capsules are carried in different types of messages and will invoke their associated codes in the active nodes. Using this mechanism, RMANP can perform such operations as data caching, local recovery and NACK suppression in a way similar to that in ARM, AEM, or PGM.

[15]After the NACK has passed the TP, the replier link is selected based on the interface on which the NACK arrives.

RMANP has been implemented over the ANTS [105] platform, a Java-based toolkit for experimenting with active networks. It is shown that the capsule code size is acceptable while the execution times are not, but the latter can be improved through several ways suggested in [103].

G. Summary of Reliable Multicast Protocols and Mechanisms

In this section, we gave an explicit definition of reliability for Internet multicast and studied mechanisms, including ARQ and FEC, to provide reliability. Scalability is possibly the biggest challenge in reliable multicast protocol design. Schemes are needed to prevent feedback implosion and for efficient retransmission. Basically, feedback implosion can be prevented by aggregating and/or suppressing acknowledgment inside a multicast group. Scalable retransmission can be achieved through the local recovery strategy. Unlike TCP, which provides reliability for unicast at the transport layer, reliable multicast protocol can often take advantage of active network support to enhance performance.

We investigated various reliable multicast schemes and protocols in this section. The major protocols mentioned are listed in Table 2 with their main properties.

Another topic closely related to reliable multicast is multicast flow and congestion control. In fact, many reliable multicast protocols introduced in this section also include a flow/congestion control scheme, which will be studied in the next section.

IV. Multicast Flow and Congestion Control

Flow control and congestion control are among the fundamental problems of Internet multicast. This is an active research area with many challenges and open issues. We first overview the proposed schemes in Section IV-A, introducing major challenges encountered and classifying proposed schemes. In Section IV-B, we discuss various fairness criteria, which are major design goals of different schemes and protocols. In Sections IV-C and IV-D, we introduce representative protocols. Section IV-E summaries this section.

A. Overview of Multicast Flow and Congestion Control

To facilitate our discussion in this section, we give the following definitions of multicast flow control and congestion control.

Multicast Flow Control: A set of techniques that match the data transmission rate to the capacities of receivers and to the service rates in the paths leading to those receivers.

Multicast Congestion Control: A set of techniques that regulate the data transmission rate in response to network conditions and the principles and mechanisms of sharing congested links among many sessions.

It should be noted that the common congestion control concept involves several other mechanisms not included in the above definition, such as over provisioning and traffic shaping [106], which are beyond the scope of this paper.

Table 2
Major Reliable Multicast Protocols Discussed in Section III

Protocols	Main Properties	Reference(s)
SRM	Receiver-initiated NACK-based protocol, using random timers to suppress NACKs and retransmissions.	[26]
LBRM	Receiver-initiated tree-based protocol, using distributed logging servers for retransmission, including a variable heartbeat mechanism for fast loss detection, and dynamically choosing between multicast and unicast for retransmissions.	[64]
RMTP	Receiver-initiated tree ACK-based protocol with selective retransmission. Statically chosen DRs aggregate ACKs and perform local recovery.	[51]
RMTP-II	Tree ACK-based protocol with NACK and FEC options. The control tree consists of a top node, DRs and receivers, supporting unordered, source-ordered, and time bounded delivery, few-to-many multicast, asymmetrical networks, and different membership control.	[54]
TMTP	Tree NAPP-based protocol. The tree is organized using ERS with bounded fan-out. TTL scoping is used for local recovery.	[66]
LMS, Search Party, and RMCM	Using router support to deliver repair request, LMS uses pre-selected replier link at each TP; Search Party selects replier links in a probabilistic manner; RMCM is similar to LMS, but designed for bi-directional shared multicast routing trees.	[92], [100], and [101]
ARM, AER, PGM, and RMANP	Using multicast routing tree for local recovery, relying on active service provided by active routers (or their collocated servers). RMANP is implemented on an active network and defines different types of capsules for different operations.	[80], [83], [102], and [103]
Lorax	Shared ACK tree-based protocol which organizes the ACK tree using ERS. Nodes on the tree are labeled using a hierarchical scheme.	[63]

Reference [106] pointed out the difference between congestion control and flow control in the unicast scenario: congestion control is a global issue and it ensures that the subnet is able to carry the offered traffic, while flow control is used to prevent a sender from transmitting data too fast as to overwhelm the receiver. In multicast, since there are many receivers, flow control must meet the requirements of many receivers, while congestion control has to deal with the fairness issues not only among multicast sessions, but also between multicast sessions and other traffics such as TCP flows. Sometimes, flow control and congestion control can be performed together, e.g., in the window-based control mechanisms adopted by TCP [107]. This is also the case in multicast. In the following discussions, we will not try to distinguish flow control and congestion control mechanisms since they are often closely related. Readers can easily identify the function(s) of a certain mechanism based on the above definitions.

Today's multicast traffic is usually transmitted on top of UDP, which lacks flow and congestion control mechanisms. This will lead to improper usage of network resources, favoring multicast flows over other responsive traffics, such as TCP flows, in best effort networks and will introduce "congestion collapse" [108], [109], which refers to bandwidth wastage due to delivering packets that will be dropped for sure later in the network.

One way to avoid congestion is to provide performance guarantees through resource reservation and admission control. RSVP [110] is proposed for providing resource reservation in Internet multicast. It is a signaling protocol that sets up QoS parameters in the routers on a multicast routing tree. Since enough resources are guaranteed, no congestion will occur in the network. However, QoS guarantees are likely to be provided only to a small fraction of Internet traffic in the near future.

In this paper, we will focus on transport layer mechanisms for multicast flow and congestion control. In the remainder of this section, we will first introduce the major challenges, then categorize proposed schemes.

1) Challenges: The major challenges in multicast flow and congestion control are scalability, heterogeneity, and fairness.

a) Scalability: A multicast group usually has more than one receiver. When the receiver population is large, scalability becomes a challenge, not only for error recovery as described in the previous section, but also for flow and congestion control. To control feedback messages efficiently, it is often helpful to distinguish shared loss and independent loss in the control mechanisms. Fig. 14 illustrates the two types of losses.

When a packet is lost on a link in the multicast distribution tree, all the downstream receivers will observe the loss and possibly report it. This is the shared loss phenomenon, which will lead to the aforementioned "feedback implosion" problem, especially when the loss is near the sender. This problem can be overcome in a similar way as in error re-

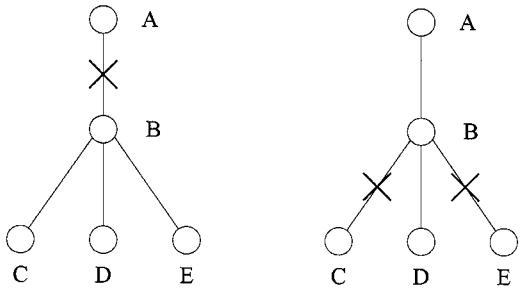

Fig. 14. (a) Shared loss. When the loss occurs at link AB, nodes C, D, and E all observe the same loss and independent loss. (b) Independent losses. The two losses at links BC and BE are independent.

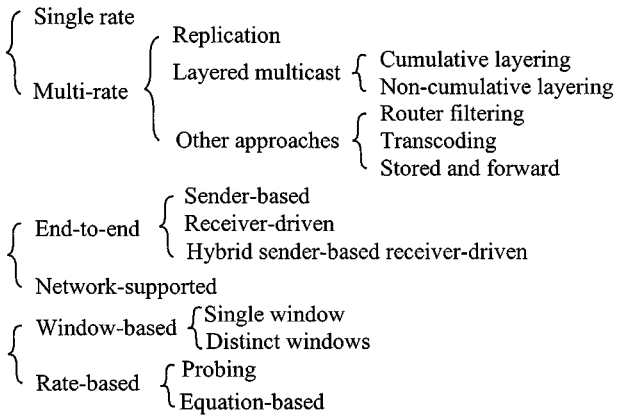

Fig. 15. Classification of multicast flow and congestion control schemes.

covery schemes through feedback aggregation and suppression, using a hierarchical structure, a timer-based mechanism, or a combination of them. The probabilistic schemes and representative-based schemes also help alleviate feedback implosion.

Independent losses refer to losses on different paths to different receivers, which are not caused by a common upstream loss. If every independent loss is reported to the sender and the sender uses reported losses as indicators of congestion in the network and reacts to each of them, the "loss path multiplicity" [111] (or "drop-to-zero" [112], [113]) problem will occur where the sender is easily overthrottled to a very low transmission rate. This problem may be solved by decoupling feedback for error control and feedback for flow and congestion control [114]. The probabilistic schemes, the representative-based schemes, and the filtering mechanism introduced in [115] and [111] all embody this principle.

b) Heterogeneity and interreceiver fairness: Another major challenge for multicast flow and congestion control is the heterogeneity of group members and network capacities. For example, the bottleneck link capacity leading to receivers in a multicast group can vary from 33.6 kb/s for a dialup link to more than 100 Mbps in a LAN. It is often desirable for a receiver to have a transmission rate that matches its receiving rate. This leads to the interreceiver fairness requirement: the transmission rate of a multicast group should satisfy faster receivers in the group while not overwhelming slower ones at the same time. Some formally defined interreceiver fairness will be given in Section IV-B. The multirate multicast flow and congestion control schemes, which will be introduced in Section IV-D, help improve interreceiver fairness.

c) Intersession fairness: A basic requirement for end-to-end multicast flow and congestion control is fairness among receivers or sessions. "Interreceiver" fairness, the fairness between receivers in a multicast session, is described above. In contrast to this, "intersession" fairness includes fairness among multicast sessions and between multicast and unicast sessions. The basic requirement is that a multicast flow should be responsive in a best effort network and should not use extremely high or low bandwidths compared with other traffics.

The definition of "intersession fairness" is largely policy-based. Different definitions are possible due to various requirements of applications, customers, and service providers. Popular fairness criteria include max–min fairness [116]–[118] and "TCP-friendliness" [109], [119], [120]. They will be discussed in Section IV-B. A related consideration is the "preferential treatment" between unicast and multicast flows. One choice is to treat them equally [115] and another choice is to give multicast flows more bandwidth to promote multicast deployment, since they use bandwidth more efficiently for multipoint transmissions [121].

2) Classification of Schemes: As shown in Fig. 15, proposed multicast flow and congestion control schemes can be categorized according to whether they are: single rate or multirate, end-to-end or network-supported, and window-based or rate-based.

In a single rate scheme, data are sent to all receivers of a multicast session at the same rate. This rate is usually restricted to the receiving rate of the slowest receiver in the session. Another choice allows the transmission rate to exceed the capacities of some receivers, but within preset tolerance bounds. The goal is to achieve the maximum value of a predefined interreceiver fairness [122]. Due to high heterogeneity among network paths and receivers, single rate schemes usually cannot achieve good interreceiver fairness.

A better approach is the multirate scheme, which sends data at multiple rates to receivers with different capabilities. One way to achieve multirate transmission is replication (or "simulcasting") [123], [124] in which the same original data is encoded into a number of streams with different rates. Another way is layered multicast [125]–[129], which divides the data into several layers. The different streams/layers of a multicast session are sent through different multicast groups. In the cumulative layering schemes, there is an order among the layers, i.e., a higher layer can only be decoded at receivers with all of its lower layers received. No such ordering exists in the noncumulative layering schemes. Multirate transmission can also be achieved by router filtering [130], transcoding [131], and store-and-forward approaches. In the router filtering scheme, routers drop packets at outgoing interfaces so that downstream receivers will receive packets at their maximum fair rates. FEC is adopted in this

scheme to provide reliability. In a transcoding scheme, intermediate network nodes decode and reencode the received high-rate data stream to a lower rate stream when congestion occurs. While in a store-and-forward scheme, intermediate network nodes cache received data and send them downsteam with proper rates. Sometimes, this scheme requires excessive storage at routers.

Many of the proposed schemes are end-to-end. They require no network support beyond multicast delivery. All flow and congestion control functionality is provided by end hosts (senders and receivers). End-to-end schemes can be further divided into sender-based, receiver-driven, and hybrid schemes. In a sender-based scheme, the sender adjusts the transmission rate(s) in response to feedback from receivers. Multirate schemes usually adopt the receiver-driven approach. For example, in layered multicast receivers may make their own decisions on joining/leaving layers based on observed network conditions. A hybrid scheme is receiver-driven, but the sender also adjusts sending rates of the layers or streams.

The end-to-end protocols are easy to deploy in the Internet. However, they have difficulty coordinating receivers in a multicast group and catching up with fast variations of congestion status in the network. Better performance can be achieved in network-supported schemes, which adopts additional functionality, such as feedback aggregation and router filtering, in the network. The tradeoff is the increased complexity in the network.

In a window-based scheme, either the sender or receivers maintain a congestion window. The congestion window represents the amount of data which may be sent in one RTT. The window size increases when there is no congestion and decreases when congestion is detected, e.g., when packet loss occurs. The sending rate is adjusted by choosing the window size according to network status. Some window-based protocols maintain a common window for all receivers of a multicast group, while others use distinct windows for each receiver, as suggested in [132]. Since the common window is usually set as the minimum window allowed by all the receivers, the common window approach sometimes restricts the throughput of the multicast session to a value that is much lower than the value allowed in the network [132].

In a rate-based scheme, the transmission rate is adjusted directly, through a probing or equation-based approach. In the probing approach, the transmission rate is increased in the absence of congestion and decreased when congestion occurs. In the equation-based approach, using measured loss probability and RTT values, the proper transmission rate is calculated using the TCP throughput models [109], [133], [134].

We will introduce multicast flow and congestion control protocols based on the classification of single rate/multirate. In Section IV-C, we will introduce single-rate protocols, focusing on how the window-based and rate-based schemes are used to adjust the transmission rates. Network support is often useful in aggregating feedback from receivers. In Section IV-D, we will introduce multirate protocols. The emphasis is put on cumulative layering protocols. Both end-to-end and network-supported protocols will be covered.

B. Fairness

In this section, we introduce fairness criteria adopted in multicast flow and congestion control schemes. Major criteria discussed include max–min fairness, interreceiver fairness and TCP-friendliness.

1) Max–Min Fairness: Max–min fairness deals with fairness between multiple sessions and receivers in a network. It is first defined in unicast scenarios [116], [117] and later extended to single-rate multicast [118] and multirate multicast [135].

In "max–min," the bandwidth allocation algorithm maximizes the allocation of bandwidth to the sources receiving the smallest allocation. The max–min fair allocation is described in [117] as follows:

1) resources are allocated in order of increasing demand;
2) no source gets a resource share larger than its demand;
3) sources with unsatisfied demands get an equal share of the resource.

Therefore, under a max–min fair allocation of bandwidth, if a source wants to increase its share of bandwidth, it is necessary to decrease the bandwidth allocated to another source that already receives a lower or equal allocation. The above definition can also be easily extended to include the sources' weights in the allocation, which reflect the sources' relative resource share [117].

Reference [118] extends the max–min fair criterion to a network with single-rate multicast sessions. It first defines a rate vector (which consists of rates allocated to each path in the network) to be feasible if each rate is not negative, the sum of rates on each link does not exceed the capacity of the link, and all the paths belonging to a multicast session are allocated the same rate. Then, max–min fair is defined for a rate vector $\vec{R}$ as the following. A rate vector $\vec{R} = \{R_1, R_2, \ldots\}$ is max–min fair if it is feasible and for each session s, one cannot generate a new feasible rate vector simply by increasing the allocated rate R_s without decreasing the allocated rate of some other session t with a rate R_t already smaller than R_s in the rate vector $\vec{R}$.

Similarly, the max–min fairness for multirate multicast [135] could be defined as the followng. An allocation of receiver rates is said to be max–min fair if it is feasible and for each receiver $d_{s,k}$ (the kth receiver in session s), one cannot find another feasible allocation by increasing the allocated rate $R_{s,k}$ without decreasing the allocated rate of some other receiver $d_{t,l}$ with a rate $R_{t,l}$ already smaller that $R_{s,k}$ in this allocation.

2) Interreceiver Fairness: In a multicast session, the sending rate to a receiver is often constrained (either too high or too low) by other receivers in the session. If the rate is higher than the receiver's desired rate, losses will occur; if the rate is lower than the desired rate, the throughput of the receiver will be smaller than what is supported and expected. In either case, the receiver suffers from performance

degradation. A single receiver fairness is defined by the relationship between the actual sending rate to a receiver and the receiver's desired rate [122], [136]. In [122] and [136], a receiver's desired rate is called the "isolated rate," which is defined to be the rate that the receiver would obtain if unconstrained by the other receivers in the session or formally as the receiver's rate under a multirate max–min fair allocation. For a single-rate multicast, the single receiver fairness for receiver i is defined as (1), where R is the sending rate and r_i is the receiver's isolated rate

$$F_i(R) = \begin{cases} \frac{R}{r_i}, & \text{if } R \le r_i \\ \frac{r_i}{R}, & \text{if } R > r_i \end{cases} = \frac{\min(r_i, R)}{\max(r_i, R)}. \quad (1)$$

Based on the definition of single receiver fairness, interreceiver fairness for single-rate multicast can be defined as the weighted sum of fairness values of all the receivers in the multicast group

$$IRF(R) = \sum_{i=1}^{n} \alpha_i F_i(R) \quad (2)$$

subject to $\sum_{i=1}^{n} \alpha_i = 1$ and $0 \le \alpha_i \le 1$, $i = 1, \ldots, n$, where $IRF()$ is the interreceiver fairness function, n is the number of receivers in the group and α_i is the weight of receiver i. Similarly, [124] gives a definition for multirate interreceiver fairness

$$IRF(R_{g1}, R_{g2}, \ldots, R_{gk}) = \sum_{j=1}^{k} \left[\sum_{i=1, i \in gj}^{n} \alpha_i F_i(R_{gj}) \right] \quad (3)$$

subject to $\sum_{i=1}^{n} \alpha_i = 1$ and $0 \le \alpha_i \le 1$, $i = 1, \ldots, n$, where the multicast group is assumed to be divided into k subgroups $\{g1, g2, \ldots gk\}$.

3) TCP-Friendliness and Other Criteria: TCP is dominant in today's Internet and its end-to-end congestion control mechanisms are crucial to the robustness of the Internet. Since a TCP flow reduces its sending rate on detection of congestion, flows without appropriate congestion control mechanisms can obtain larger share of bandwidth on congested links and will possibly lead to "congestion collapse" [109] in the network. TCP-friendliness is a fairness criterion to guide behaviors of non-TCP based best effort traffics and to prevent them from starving TCP flows. A TCP-friendly (or TCP-compatible) flow can be described as the following [109], [119], [120]. A flow is TCP friendly if its long-term throughput does not exceed the throughput of a conformant TCP connection under the same circumstances.

Accordingly, we can informally define the multicast version of this criterion as the following. A multicast session is TCP friendly if on any of its source-to-destination paths, its long-term throughput does not exceed the throughput of a conformant TCP connection on that path.

This definition is suitable for both single-rate and multirate multicast sessions. For a single-rate session, the requirement is to restrict its throughput to be less than a conformant TCP session on the most congested path in the multicast distribution tree [112], [115]. For a multirate session, the multicast throughputs on some paths may exceed throughputs of TCP sessions on some more congested paths.

TCP's window-based congestion control mechanisms react to multiple time scales of congestion from within one RTT to a longer period consisting of at least several RTTs [132]. However, it is usually acceptable for a congestion control mechanism to only respond in the longer time scale [137]. Therefore, TCP-friendliness is defined on long-term throughput (or average throughput).

TCP uses the AIMD [138] algorithm for congestion control. On detecting a loss, it decreases the size of its window by a factor of two and attempts to get extra bandwidth by increasing the window linearly when there is no congestion. The long-term throughput of a TCP flow can be approximated by the following equation [109], [133]:

$$T_{TCP} = \frac{C \times S}{RTT \times \sqrt{p}} \quad (4)$$

where C is a constant, S is the packet size, RTT is the round trip time including queueing delay, and p is the packet loss rate. Equation (4) applies when the loss rate is below 5%. A more accurate equation modeling TCP's throughput is given in [134], which takes into account the effect of retransmission timeouts and applies to a wider range of loss rates.

In the real world, the packet size may vary from time to time, accurate RTT and loss rate are generally difficult to obtain, and various TCP implementations can achieve throughputs that vary considerably. Therefore, the above equation-based approach should only be used for providing an approximate estimate or giving a loose bound of TCP's throughput [109].

On observing that the AIMD algorithm does not work well for applications such as Internet audio and video—the user-perceived quality will drop drastically when the throughput decreases multiplicatively, [139] proposed a set of "binomial algorithms," such as inverse-increase/additive-decrease and SQRT,[16] for these applications. These algorithms are stable, have throughputs varying with $1/\sqrt{p}$ as required by (4), but they only share bandwidth fairly with TCP when an active queue management scheme like RED [140] is used at the bottleneck link.

There are other fairness criteria proposed. For example, two types of fairness, rate-oriented fairness and window-oriented fairness, are defined in [132].

1) An algorithm is said to have rate-oriented fairness if the average throughput of each session at equilibrium is independent of its round trip time and only depends on its loss probability.
2) An algorithm is said to have window-oriented fairness if the average amount of outstanding data of each session at the equilibrium point is independent of its round trip time and only depends on its loss probability.

Neither rate-oriented nor window-oriented fairness is equivalent to the above-defined TCP-friendliness. However,

[16]SQRT increases throughput inversely proportionally and decreases throughput proportionally to the square root of the size of the current window.

it is observed that TCP's congestion control mechanisms have window-oriented fairness [132].[17] Therefore, a mechanism having window-oriented fairness could be considered more "TCP-like" than one having rate-oriented fairness. However, it is also possible for a mechanism with rate-oriented fairness to be TCP friendly. For example, if we adopt $C = 1.22$ in (4) according to [109] and the following rate-oriented fairness criterion:

$$T \leq \frac{1.22 \times S}{RTT_{\max} \times \sqrt{p}} \tag{5}$$

where $RTT_{\max}$ is the maximum possible RTT, then the mechanism will be TCP friendly.

If a rate-based congestion control mechanism is used, the rate-oriented fairness criterion enjoys the benefit of not requiring RTT estimation. This is essential in multicast scenarios where estimating RTTs from all of the many receivers is extremely difficult [132].

There are also proposals that advocate giving more bandwidth to multicast flows to encourage using multicast delivery [121]. Mechanisms based on this allocation policy, however, are likely to be non-TCP friendly.

Irrespective of whether TCP-friendliness is adopted, a basic requirement for best effort multicast (or other non-TCP) traffics should be responsiveness, i.e., they should gracefully decrease the transmission rate in response to congestion and restore their bandwidth share when congestion disappears. Furthermore, neither multicast nor unicast sessions should receive bandwidth allocations much higher or much lower than other competitive sessions. This corresponds to the "bounded fairness" concept defined in [141].

C. Single-Rate Protocols

In designing a single-rate protocol, two problems need to be addressed. One is how to collect feedback from receivers and the other is how to adapt transmission rate to network status based on the feedback. The feedback from receivers can be piggybacked on ACKs or NACKs or sent back separately. The feedback messages can be aggregated or suppressed in similar ways as in reliable multicast protocols, as introduced in Section III-D. The transmission rate can be adjusted using window-based or rate-based approaches. In the following, we introduce representative single-rate protocols based on whether they are window-based or rate-based. Their feedback processing mechanisms will be introduced as necessary.

1) Window-Based Protocols: TCP adopts a window-based scheme for both error control and flow and congestion control. Window-based schemes are also possible and sometimes desirable for multicast flow and congestion control. However, TCP's mechanisms are not directly applicable in the multicast scenario. Scalability, heterogeneity and fairness issues need to be addressed. In this section, we will introduce some representative protocols, including NCA [115], MTCP [112], and a protocol using distinct windows for each receiver. The purpose is to illustrate how a window-based protocol is designed.

a) NCA: NCA is a single-rate TCP-friendly multicast congestion control protocol. It consists of two parts: a nomination algorithm and a rate adjustment algorithm.

The purpose of the nomination algorithm is to dynamically select a nominee representing the worst path. An active network model as used in PGM is assumed. Each receiver periodically sends to its upstream active server the estimated loss probability p and RTT. An active server identifies a worst receiver among its children based on $RTT \times \sqrt{p}$ according to (4) and reports that information upstream. Eventually, the sender will identify a worst receiver of the entire group as the nominee and ask it to send ACK for every packet it receives. The worst receiver and nominee information is maintained as soft states in the sender and active servers.

The rate adjustment algorithm operates in a similar way as TCP NewReno [142] using ACKs from the nominee. The main difference between them is that the algorithm does not retransmit packets on detecting losses, since NCA is a congestion control protocol decoupled from the error control functionality.

b) MTCP: MTCP [112] is a hierarchical window-based protocol for multicast flow and congestion control as well as error control. The hierarchy includes the sender as the root, receivers as leaves and SAs in between, as shown in Fig. 16.

The sender and each SA maintain a congestion window (cwnd) and a transmit window (twnd). The cwnd estimates the congestion level of the network and is maintained using congestion control mechanisms similar to TCP Vegas. The twnd indicates outstanding packets at the sender or an SA, which is increased when a packet arrives at the SA (or the sender) and decreased when the packet is acknowledged by all the children of the SA (or the sender). Each SA sends to its upstream SA (or the sender) congestion summaries, which include a minimum congestion window (minCwnd) and a maximum transmit window (maxTwnd). The minCwnd is the minimum value of the SA's own cwnd and the cwnds reported by its immediate downstream children. The maxTwnd is the maximum value of the SA's own twnd and the twnds reported by its immediate downstream children. SAs whose children are leaf nodes just include their own cwnd and twnd in the congestion summary.

Congestion summaries are piggybacked on ACKs and NACKs or sent periodically when ACKs and NACKs are lacking to prevent protocol deadlocks. Each receiver also sends an advertised window (awnd) upstream indicating the number of available buffers. The awnds are aggregated along the hierarchy. Defining a current window (curwnd) at the sender as the difference between its minCwnd and maxTwnd and the sender's awnd as the minimum value reported by its children, the number of packet sent each time should be no more than min(curwnd, awnd).

c) Using distinct windows: A problem of MTCP and some other window-based multicast flow and congestion

[17] It is also emphasized in [132] that the definition of fairness orientation is based on the algorithm's performance at the equilibrium point, which is not the average performance. In addition, TCP's average window size is not totally independent of the RTT.

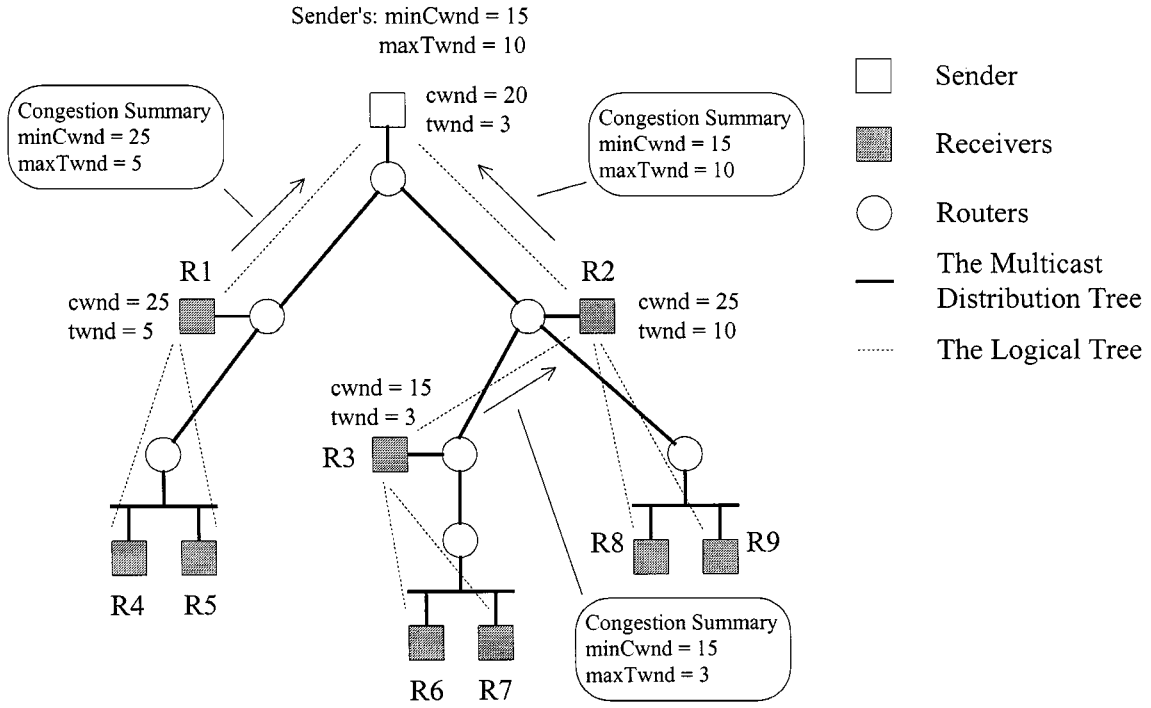

Fig. 16. Illustration of MTCP's hierarchical structure and status reports. (Internal receivers on the logical tree are also SAs.)

control protocols is that a common window is used for many receivers, which is set as the minimum window allowed by all the receivers. This will restrict the throughput of the multicast session to a value that is lower than the value allowed in the network [132].

Reference [132] proposed a multicast flow and congestion control protocol using distinct windows for each receiver. A token concept is adopted which allows out-of-order ACKs in the protocol. Using the token concept, the sender maintains a token pool for each receiver. Initially, there are a window-size number of tokens in each receiver's pool. Tokens decrease with packets sent and increase with ACKs received. A packet is allowed to send only when enough tokens exist in every receiver's token pool.

To scale to large number of receivers, the protocol in [132] distributes the responsibility of maintaining distinct windows to receivers. Each receiver j maintains its own window w_j and reports to the sender a value n_j, which denotes the highest packet sequence number allowed to arrive at j within one RTT and is calculated as [18]:

$$n_j = m_j + \min(w_j, B_j) \tag{6}$$

where m_j is the largest packet sequence number so far received and B_j is the buffer available at the receiver. The feedback messages containing n_j are aggregated in a hierarchical structure. Their minimum value $n_{\text{send}} = \min_j n_j$ will be adopted by the sender to restrict itself from transmitting packets with sequence number beyond n_{send}. It is also possible to aggregate or suppress feedback of n_j along the multicast routing tree using the hierarchical approach described earlier. Alternatively, an SRM-like approach, as discussed in [132], may be used, either together with the hierarchical approach or by itself.

2) Rate-Based Protocols: In a single-rate rate-based protocol, the sender adjusts its sending rate based on feedback from receivers. To scale to large multicast groups with many receivers, feedback messages need to be aggregated or suppressed before being processed by the sender. The sender then adjusts the sending rate using either a probing approach or an equation-based approach. The feedback from receivers can be packet loss ratios and/or RTTs measured at receivers, calculated desired receiving rates by receivers or simply ACKs and NACKs.

In the LTRC [143] protocol, an EWMA [140] of packet loss is maintained at each receiver according to (7)

$$\text{avg} \leftarrow (1 - w_p)\,\text{avg} + w_p p_c \tag{7}$$

where p_c is the current loss ratio, calculated by dividing the number of lost packets observed by the number of expected packets, avg is the average loss, and $(1 - w_p)$ is the weight of the previous average value in calculating the new one. The average loss is updated on receiving each data packet, observing a loss and receiving a repair packet. In the last case, the average loss is updated as

$$\text{avg} = w_r \text{avg} \tag{8}$$

where w_r is a weight. The average loss is reported in each NACK (or ACK) sent from the receiver.

The TFMCC [137] protocol adopts the equation-based approach. Each receiver calculates the throughput of a TCP session in the same circumstance by using the estimated values

[18]Similar equation is provided in [132] which accounts for retransmissions when calculating n_j.

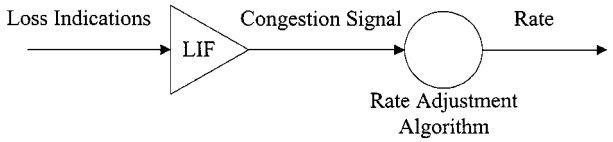

Fig. 17. General model of FLICA algorithms.

of RTT, loss rate p, retransmission timeout value t_0, and the equation derived from [134]

$$B(p) = \frac{S}{RTT\sqrt{\frac{2bp}{3}} + t_0 \min\left(1, 3\sqrt{\frac{3bp}{8}}\right) p(1+32p^2)} \tag{9}$$

where $B(p)$ is TCPs throughput in bytes, S is the packet size in bytes, and b is the number of packets acknowledged by each TCP ACK. The receiver then reports the $B(p)$ value to the sender in NACKs. NACKs carrying larger $B(p)$ values are likely to be suppressed by those carrying smaller values. Various suppression mechanisms may be used. The sender then adjusts its sending rate according to the received value. Several mechanisms for estimating RTT, t_0 and p, are also suggested in [137].

In the representative-based congestion control protocol proposed in [89] and [90], some receivers are dynamically selected as representatives of a multicast group, based on feedback messages (ACKs and NACKs) from receivers. Receivers sending NACKs have priority to be selected as representatives over those sending ACKs. The sender periodically multicasts the selected representative set to the group. Receivers that find themselves in the set will operate as representatives. Each representative represents a congested subtree of the multicast distribution tree and sends immediate feedback to the sender. A timer-based suppression scheme is adopted. Feedback from other receivers will likely be suppressed by feedback from representatives. The sender adjusts its sending rate using the probing approach.

To solve the "loss path multiplicity" problem, [111] proposed to filter out some of the LIs of lost packets. FLICA is a set of algorithms designed for this purpose. The general model of FLICA includes a LIF and a rate adjustment algorithm, as shown in Fig. 17. Only the congestion signals that pass the LIF are used by the rate adjustment algorithm. This general model can be applied to a wide range of implementations, including the aforementioned LTRC and representative-based scheme. The function of LIF can also be distributed in the multicast tree.

A receiver can report its LI, such as estimated loss ratio p, to the sender. An exponential smoothing filter is suggested in [111] to estimate p

$$p_{i+1} \leftarrow \begin{cases} (1-\alpha)p_i + \alpha, & \text{if packet } i \text{ is lost} \\ (1-\alpha)p_i, & \text{if packet } i \text{ is received} \end{cases} \tag{10}$$

where α is the gain factor of the filter.

D. Multirate Protocols

Multirate flow and congestion control protocols are proposed to meet the requirement of heterogeneous receivers in a multicast group, so as to achieve higher interreceiver fairness. In this section, two major approaches for multirate transmission will be discussed: replication and layered multicast. Most of the existing layered multicast protocols adopt cumulative layers. However, noncumulative layering protocols are also proposed. Reference [129] studied pros and cons of cumulative and noncumulative layering approaches and designed a noncumulative layering protocol. In this section, we will only introduce cumulative layering protocols, including both end-to-end and network-supported protocols.

1) Replication: For point-to-multipoint transmission, multicast achieves higher bandwidth efficiency than unicast at the cost of control granularity [144]. The replication approach sits between the two extremes represented by multicast and unicast. The Destination Set Splitting (DSS) protocol uses replication for reliable multicast, in which the sender splits receivers into several groups and carries on independent conversations with each group using an ARQ protocol [145]. In this section, we will introduce one replication protocol used for multicast flow and congestion control: DSG [123], [124].

a) DSG: In DSG, the same original data is encoded into a small number of streams with different rates. The rate of each stream is dynamically adjusted within prescribed limits according to feedback from the receivers. The streams are transmitted using different multicast groups. A receiver joins an appropriate multicast group to receive the data. It may switch to a nearby stream when its desired reception rate cannot match the sending rate of the current stream due to changes in the network or in its own receiving requirement.

Fig. 18 shows an example of DSG. In this figure, link capacities are shown besides the links. The original data are encoded into three streams with bandwidths 896 kb/s, 384 kb/s and 128 kb/s, respectively. Based on their receiving capacities, receiver R1 subscribes to stream 3, receivers R2 and R3 subscribe to stream 1 and receivers R4 and R5 subscribe to stream 2.

By using multiple streams for the same original data, DSG can achieve higher interreceiver fairness than single-rate multicast protocols. However, bandwidth efficiency of DSG is also decreased, since there are some links carrying multiple streams that contain the same original data. This inefficiency can be avoided by using the layering approach.

2) Layered Multicast: In a cumulative layered multicast approach for streaming applications, the signal is encoded into one basic layer and several enhancement layers. The basic layer contains basic information for decoding the signal and can be independently decoded. A higher layer provides refinement information to previous layers and can only be decoded together with all the lower layers. Different layers are transmitted in different multicast groups. Receivers join the basic layer and as many enhancement layers as they can handle. The basic layered multicast is illustrated in Fig. 19.

In this figure, there are three layers with bandwidths 128 kb/s, 256 kb/s, and 512 kb/s, respectively. Link capacities are shown besides the links. Due to bandwidth limitations, receiver R1 only subscribes to layer 1, receivers R4 and R5 subscribe to both layers 1 and 2, and receivers R2 and R3 are able to receive all three layers.

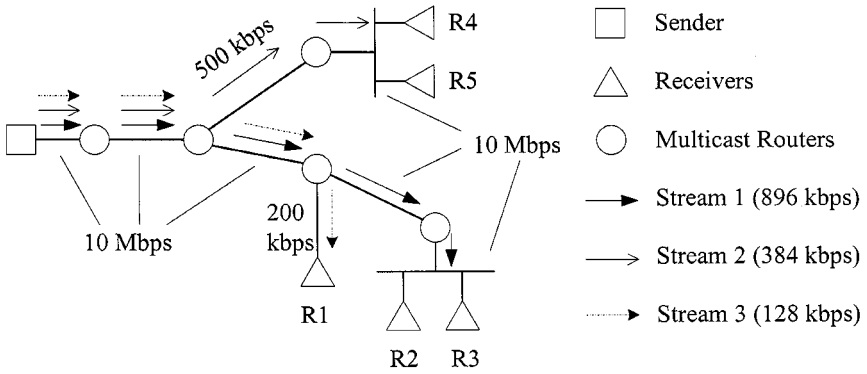

Fig. 18. Illustration of DSG.

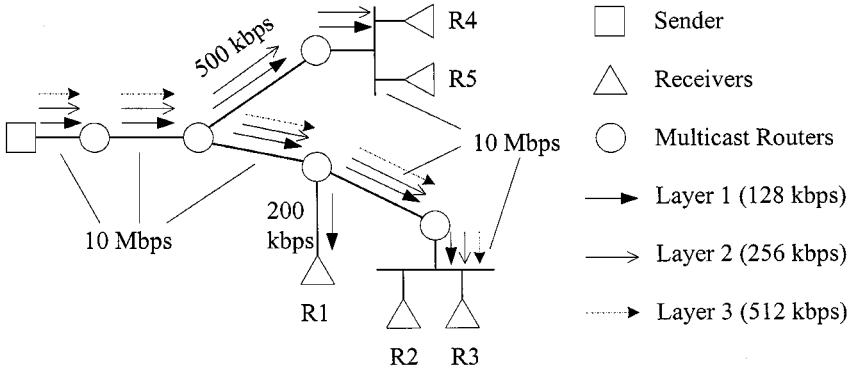

Fig. 19. Illustration of layered multicast.

The layering approach can also be used for reliable multicast flow and congestion control [115], [127], [146], [147]. In reliable data transfer, all data should be delivered to all receivers eventually. With layered multicast, receivers with higher bandwidths can obtain the entire data in a shorter time. For this purpose, the data need to be arranged in layers in an appropriate way. FEC often helps in this procedure. By introducing redundancies in the original data, it gives receivers the flexibility of not receiving all of the transmitted packets.

a) Receiver-driven layered multicast: RLM [125] serves as the basic solution of layered multicast. It is an end-to-end receiver-driven protocol. RLM receivers dynamically determine the appropriate number of layers to join by the "join experiment" and "shared learning" mechanisms.

In RLM, a receiver will spontaneously subscribe to a higher layer (if any) when the current layers are properly received. If the subscription causes congestion, the receiver quickly drops the new layer; otherwise, the receiver will retain the layer for enhanced receiving quality. This procedure is called the join experiment. Since the join experiment may lead to congestion in the network, it should be conducted only when the subscription is likely to succeed. For this purpose, a join timer is managed for each layer using an exponential backoff strategy. A receiver can only subscribe to a higher layer when the current layer's join timer expires.

If every receiver conducts the join experiment independently, the resulting congestion will lead to poor scalability of the protocol. The solution adopted by RLM is shared learning. A receiver announces a join experiment to the entire group before hand, so that all receivers can learn from other receivers' failed join experiments. Uniform dropping is assumed in the network, which makes the result of a join experiment available to receivers only joining lower layers.

RLM suffers from some drawbacks.

1) Join experiment introduces losses.
2) Shared learning requires: a) multicasting to the entire group and b) each receiver to maintain a variety of state information which may not be necessary otherwise.
3) RLM exhibits significant and persistent instability with VBR traffic [148].
4) RLM cannot achieve fairness among RLM sessions or between RLM and TCP sessions.

b) Other layered multicast protocols: Later proposed protocols [126]–[128], [149]–[156] extend RLM in several aspects. We introduce some of them in this section to illustrate how they solve one or more of the above mentioned problems of RLM. Analyses of some of the protocols as well as RLM can be found in [148], [149], [157].

LVMR [126] is a hierarchical rate-control protocol. In LVMR, a multicast group is divided into domains and each domain contains several subnets. There is an IA in each domain and a SA in each subnet. Join experiment is conducted by a receiver within its own subnet in a similar way as RLM. The SA in the subnet will report join-experiment failures or subnet congestion to the IA of the domain. Using this information, the IA builds a knowledge base for intelligently coordinating join experiments in its domain. By using the hierarchical mechanism, information is distributed between the sender, IAs, SAs, and receivers. Compared with RLM,

each entity maintains only the information relevant to itself and multicasting to the entire group for shared learning is avoided. The hierarchical structure of LVMR is also used for error recovery [158].

RLC [127] is a TCP-friendly layered multicast protocol. To mimic the behavior of TCP (AIMD), the cumulative bandwidths of layers are exponentially distributed. RLC mimics TCP's behavior represented by (4) except that the throughput does not depend on RTTs between sender and receivers. A coordination protocol based on SP is adopted in RLC to synchronize the receivers. SPs correspond to periodically sent and specially flagged packets in the data stream. A receiver can only attempt a join immediately after an SP. A sender-initiated probing mechanism is used in RLC instead of join experiments. Periodically, the sender doubles its transmission rate during a short burst interval, stops transmitting after the burst for an equally long interval (the relaxation period), and resumes transmission at the normal rate. Congestion observed during the burst serves as a warning for not increasing the subscription level. Sender-initiated probing avoids the inefficiencies introduced by long leave delays of failed joins.

PLM [128] is a protocol based on the assumption of an FS network in which every router implements a FS. PLM uses a receiver-driven version of packet pair [159] to infer the available bandwidth in the network. The sender sends via cumulative layers and emits two packets back-to-back on each of the layers. Since FS is assumed in the network, receivers can infer the bottleneck bandwidth through intervals between every pair of packets and make their join or leave decisions. By adopting packet pair for bandwidth inference, PLM avoids losses incurred by join experiments. The simulations in [128] show that PLM can rapidly converge to the optimal link utilization and enjoys inter-PLM fairness and TCP fairness.[19]

Priority dropping is investigated in [149] in which routers drop packets from higher layers (which have lower priorities) when congestion occurs to protect lower layers. As a result, priority dropping uses network bandwidth more efficiently. The scheme is stable and shares bandwidth fairly between multicast sessions, if their layer bandwidth distributions are the same. However, implementing priority dropping in the network is complex.

RLMP [150] tries to achieve the good performance of priority dropping while avoiding its high complexity. It uses only two priority levels: the highest subscribed layer has a low priority and other layers have a high priority. The highest layer absorbs packet losses when congestion occurs. Long-term loss rates of both low and high priority layers are used by the receiver to determine the optimal subscription level. RLMP is stable even under bursty traffic and achieves fairness between competing multicast sessions (by sharing the "loss rate knowledge" among receivers).

RALM [151] proposed a new router-initiated suspension mechanism, which suspends (temporarily stops transmitting) lower priority layers when congestion occurs. Routers detect congestion by monitoring the status of queues at their outgoing interfaces. The precedence of layers of different multicast groups is maintained in the control plane and data delivery is not affected. RALM achieves fairness among multicast sessions and between multicast and TCP sessions, scales well to number of receivers and number of sessions, and is incrementally deployable in the Internet. The tradeoff is the additional complexity introduced in the network.

[19]"TCP fairness" in [128] means that "PLM does not significantly affect the performance (throughput, delay, etc.) of TCP flows when sharing the same bottleneck."

E. Summary of Multicast Flow and Congestion Control

In this section, we investigated current research for multicast flow and congestion control. We began with an overview of challenges and proposed schemes. Fairness criteria were discussed in depth, as they were the requirements and design goals of various control schemes. We then introduced representative protocols and schemes based on whether they are single rate or multirate.

In the following, as a brief summary of this section, we discuss some design choices of a multicast flow and congestion control protocol.

Time Scale of Control: The time scale of control can vary from within one RTT to several RTTs or to even larger time intervals. Too large a time scale will cause a time lag in learning about the congestion status, which will lead to instability of a protocol, especially under bursty traffic. Too short a time scale may lead to highly fluctuating throughputs, which may not be acceptable in some applications, such as real-time streaming. A protocol should choose a proper control time scale and achieve it with proper control mechanisms.

In a rate-based scheme, adjusting sending rate based on feedback from receivers usually takes effect at least several RTTs after congestion occurs. Window-based schemes can react within one RTT, even if the window size is kept constant [132]. This is because the throughput of a window-based scheme depends on RTTs from sender to receiver(s), which changes with congestion status in the network. In a layered multicast protocol, if an end-to-end receiver-driven approach is adopted, the time scale of control is usually large, due to the relatively long time of detecting congestion at receivers and performing multicast joining/leaving operations. Network-supported protocols, such as RALM, can react in a shorter time scale using network mechanisms.

Fairness Criteria: We discussed various fairness criteria in Section IV-B. Better interreceiver fairness gives receivers more flexibility on choosing their receiving rates. Multirate schemes have better interreceiver fairness than single-rate schemes. However, the former are usually more complex than the latter. The scheme should be chosen based on the heterogeneity of targeted networks and receivers as well as the requirements of the applications.

For the intersession fairness, TCP-friendliness is the design goal of many proposed protocols. However, following TCPs short time scale behavior is generally not desirable for real-time applications. Many protocols for real-time applications smooth some of the control parameters, such as the measured loss probability, over a longer time interval, so that they are compatible with TCP only on a longer time scale.

Network Support: Network support provides performance enhancements to end-to-end approaches. When employing network mechanisms in a protocol, one should ensure that the added complexity in the network can be compensated by the obtained performance gains. Major challenges of a network-supported protocol are minimizing the extra burden introduced in the network and enabling incremental deployment. A protocol that works only after every router in the network has implemented it cannot be easily deployed in the Internet.

V. Conclusion

Beginning with a brief introduction to the traditional IP multicast model, the "host group" model, and today's multicast architecture, we discussed in depth three active research areas of Internet multicast: scalable multicast routing, reliable multicast, and multicast flow and congestion control. The first is a network layer issue and the latter two are transport layer issues. Challenges and solutions as well as representative schemes and protocols were introduced. The main goals of this paper are to summarize the recent work on Internet multicast in the above three areas and to provide directions for future research.

Research efforts in the network layer and transport layer are often closely related. For example, routing information and router mechanisms can be adopted by reliable multicast protocols and multicast flow and congestion protocols for enhanced performance. In addition, alternative multicast routing models, such as EXPRESS, xcast/sgm approaches, and ESM, introduce new challenges on the design of transport layer protocols. This is an area for further research.

As transport layer issues, multicast error control (in reliable multicast) and flow/congestion control are also closely related. Some approaches, such as the hierarchical structure and feedback aggregation/suppression mechanisms, can be adopted by both error control and flow/congestion control protocols under the same framework. Moreover, there is a coupling of error control and flow/congestion control in some window-based or rate-based schemes. Decoupling these two types of control is sometimes desirable in multicast. A common way to achieve this is decoupling feedback for errors and feedback for congestion [114].

Besides the above areas, extensive work has also been done in some other areas of Internet multicast research, such as QoS support in multicast, integration of multicast and MPLS, and wireless multicast, which are also important but beyond the scope of this paper.

References

[1] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," Network Working Group, RFC 1771, Mar. 1995.

[2] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification," Network Working Group, RFC 2460, Dec. 1998.

[3] V. Fuller *et al.*, "Classless interdomain routing (CIDR): An address assignment and aggregation strategy," Network Working Group, RFC 1519, Sept. 1993.

[4] S. Deering, "Host extensions for IP multicasting," Network Working Group, RFC 1112, Aug. 1989.

[5] S. Casner. (1994) Frequently asked questions (FAQ) on the multicast backbone (MBone). Information Sciences Inst., Univ. Southern California. [Online]ftp://ftp.isi.edi/mbone/faq.txt

[6] W. Fenner, "Internet group management protocol, version 2," Network Working Group, RFC 2236, Nov. 1997.

[7] B. Cain, S. Deering, I. Kouvelas, and A. Thyagarajan, "Internet group management protocol, version 3," Internet draft, draft-ietf-idmr-igmp-v3-09.txt, Jan. 2002.

[8] W. Liao and D. Yang, "Receiver-initiated group membership protocol (RGMP): A new group management protocol for IP multicasting," in *Proc. IEEE Int. Conf. Network Protocols*, Toronto, ON, Canada, Oct./Nov. 1999, pp. 51–58.

[9] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol (DVMRP)," Network Working Group, RFC 1075, Nov. 1988.

[10] A. Adams, J. Nicholas, and W. Siadak, "Protocol independent multicast dense mode (PIM-DM): Protocol specification (revised)," Internet draft, draft-ietf-pim-dm-new-v2-01.txt, Feb. 2002.

[11] J. Moy, "Multicast extensions to OSPF," Network Working Group, RFC 1584, Mar. 1994.

[12] B. Fenner, M. Handley, H. Holbrook, and J. Kouvelas, "Protocol independent multicast sparse mode (PIM-SM): Protocol specification (revised)," Internet draft, draft-ietf-pim-sm-new-v2-01.txt, Mar. 2002.

[13] A. Ballardie, "Core-based trees (CBT version 2) multicast routing," Network Working Group, RFC 2189, Sept. 1997.

[14] ——, "Core-based trees (CBT) multicast routing architecture," Network Working Group, RFC 2201, 1997.

[15] S. Kumar *et al.*, "The MASC/BGMP architecture for interdomain multicast routing," in *Proc. ACM SIGCOMM*, Vancouver, BC, Canada, Aug./Sept. 1998, pp. 93–104.

[16] J. Moy, "OSPF version 2," Network Working Group, RFC 2178, Apr. 1998.

[17] T. Bates *et al.*, "Multiprotocol extensions for BGP-4," Network Working Group, RFC 2283, Feb. 1998.

[18] D. Farinacci *et al.*, "Multicast source discovery protocol (MSDP)," Internet draft, draft-ieft-msdp-spec-13.txt, Nov. 2001.

[19] R. Perlman, "Simple multicast: A design for simple, low-overhead multicast," Internet draft, draft-perlman-simple-multicast-03.txt, Oct. 1999.

[20] H. W. Holbrook and D. R. Cheriton, "IP multicast channel: EXPRESS support for large-scale single-source applications," in *Proc. ACM SIGCOMM*, Cambridge, MA, Aug./Sept. 1999, pp. 65–78.

[21] K. Obraczka, "Multicast transport protocols: A survey and taxonomy," *IEEE Commun. Mag.*, vol. 36, pp. 94–102, Jan. 1998.

[22] J. M. Chang and N. F. Maxemchuk, "Reliable broadcast protocols," *ACM Trans. Comput. Syst.*, vol. 2, pp. 251–273, Aug. 1984.

[23] S. Armstrong, A. Freier, and K. Marzullo, "Multicast Transport Protocol," Network Working Group, RFC 1301, 1992.

[24] T. Strayer. Xtp Web Page. [Online]. Available: http://www.ca.sandia.gov/xtp.html

[25] H. Schulzrinne, S. Casner, R. Frederich, and V. Jacobson, "RTP: A transport protocol for real-time applications," Network Working Group, RFC 1889, 1996.

[26] S. Floyd, V. Jacobson, C. G. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Trans. Networking*, vol. 5, pp. 784–803, Dec. 1997.

[27] J. Macker and W. Dang, "The Multicast Dissemination Protocol (MDP) Framework," Internet draft, draft-macker-mdp-framework-00.txt, Nov. 1996.

[28] J. R. Cooperstock and S. Kotsopoulos, "Why use a fishing line when you have a net? An adaptive multicast data distribution protocol," in *Proc. USENIX*, San Diego, CA, Jan. 1996, pp. 343–352.

[29] H. Holbrook and B. Cain, "Source-specific multicast for IP," Internet draft, draft-holbrook-ssm-00.txt, Mar. 2000.

[30] S. Bhattacharyya *et al.*, "A framework for source-specific IP multicast deployment," Internet draft, draft-bhattach-pim-ssm-00.txt, July 2000.

[31] P. I. Radoslavov, D. Estrin, and R. Govindan, "Exploiting the bandwidth-memory tradeoff in multicast state aggregation," Univ. Southern California, Los Angeles, CA, Tech. Rep. 99-697, 1999.

[32] D. Thalar and M. Handley, "On the aggregatability of multicast forwarding state," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 1654–1663.

[33] R. Briscoe and M. Tatham, "End to end aggregation of multicast addresses," Internet draft, draft-briscoe-ama-00.txt, Nov. 1997.

[34] J. Tian and G. Neufeld, "Forwarding state reduction for sparse mode multicast communication," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1998, pp. 711–719.
[35] I. Stoica, T. S. E. Ng, and H. Zhang, "REUNITE: A recursive unicast approach to multicast," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 1644–1653.
[36] R. Boivie, "A new multicast scheme for small groups," IBM T. J. Watson Res. Center, Yorktown Heights, NY, Res. Rep. RC21512(9704), 1999.
[37] R. Boivie and R. Feldman, "Small group multicast," Internet draft, draft-boivie-sgm-01.txt, July 2000.
[38] D. Helder and S. Jamin, "IPv4 option for somecast," Internet draft, draft-dhelder-somecast-00.txt, July 2000.
[39] D. Ooms, W. Livens, and O. Paridaens, "Connectionless multicast," Internet draft, Internet draft, draft-ooms-cl-multicast-02.txt, Apr. 2000.
[40] I. Yuju, "Multiple destination option on IPv6 (MDO6)," Internet draft, Internet draft, draft-imai-mdo6-01.txt, Mar. 2000.
[41] D. Ooms, "Taxonomy of xcast/sgm proposals," Internet draft, Internet draft, draft-ooms-xcast-taxonomy-00.txt, July 2000.
[42] Y. Chu, S. G. Rao, and H. Zhang. (2000) A Case for end system multicast. [Online]. Available: http://www.cs.cmu.edu/~sanjay/papers/sigmetrics-2000.ps.gz
[43] P. Francis. (1999) Yallcast: Extending the internet multicast architecture. [Online]. Available: http://www.yallcast.com/docs/index.html
[44] H. Hummel, "Source only multicast," Internet draft, draft-hummel-pim-so-00.txt, June 2000.
[45] "Standard for Distributed Interactive Simulation-Application Protocols," Univ. Central Florida, Orlando, FL, Tech. Rep. IST-CR-94-50, 1994.
[46] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, 2nd ed. Reading, MA: Addison-Wesley, 1994.
[47] A. Mankin *et al.*, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols," Network Working Group, RFC 2357, 1998.
[48] S. K. Kasera, J. Kurose, and D. Towsley, "A comparison of server-based and receiver-based local recovery approaches for scalable reliable multicast," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1998, pp. 988–995.
[49] D. Towsley, J. Kurose, and S. Pingali, "A comparison of sender-initiated and receiver-initiated reliable multicast protocols," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 398–406, Apr. 1997.
[50] B. N. Levine and J. Garcia-Luna-Aceves, "A comparison of knows classes of reliable multicast protocols," in *Proc. IEEE Int. Conf. Network Protocols*, Columbus, OH, Oct./Nov. 1996, pp. 112–121.
[51] J. C. Lin and S. Paul, "RMTP: A reliable multicast transport protocol," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1996, pp. 1414–1424.
[52] R. Yavatkar and L. Manoj, "Optimistic strategies for large-scale dissemination of multimedia information," in *Proc. ACM Multimedia*, Anaheim, CA, Aug. 1993, pp. 13–20.
[53] M. Handley *et al.*, "The reliable multicast design space for bulk data transfer," Network Working Group, RFC 2887, 2000.
[54] B. Whetten and G. Taskale, "An overview of reliable multicast transport protocol II," *IEEE Network*, vol. 14, pp. 37–47, Jan./Feb. 2000.
[55] H. Garcia-Molina and A. Spauster, "Ordered and reliable multicast communication," *ACM Trans. Comput. Syst.*, vol. 9, pp. 242–271, Aug. 1991.
[56] D. G. Petitt, "Solutions for reliable multicasting," Master, Naval Postgraduate School, Monterey, CA, 1996.
[57] B. Whetten *et al.*, "Reliable multicast transport building blocks for one-to-many bulk-data transfer," Internet draft, draft-ietf-rmt-buildingblocks-02.txt, Mar. 2000.
[58] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint communication: A survey of protocols, functions and mechanisms," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 277–290, Apr. 1997.
[59] D. R. Cheriton and D. Skeen, "Understanding the limitations of causally and totally ordered communication," in *Proc. ACM Symp. Operating System Principles*, Ashville, NC, Dec. 1993, pp. 44–57.
[60] D. D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," in *Proc. ACM Symp. Communications Architectures and Protocols*, 1990, pp. 200–208.
[61] V. Jacobson, S. McCanne, and S. Floyd, "Lightweight sessions—A new architecture for realtime applications and protocols," in Networkshop'93, Melbourne, Australia, Nov. 1993.
[62] S. McCanne, "Scalable multimedia communication using IP multicast and lightweight sessions," *IEEE Internet Comput.*, pp. 33–45, Mar./Apr. 1999.
[63] B. N. Levine, D. B. Lavo, and J. Garcia-Luna-Aceves, "The case for reliable concurrent multicasting using shared ACK trees," in *Proc. ACM Multimedia*, Boston, MA, Nov. 1996, pp. 365–376.
[64] H. W. Holbrook, S. K. Singhal, and D. R. Cheriton, "Log-based receiver-reliable multicast for distributed interactive simulation," in *Proc. ACM SIGCOMM*, Cambridge, MA, Aug./Sept. 1995, pp. 328–341.
[65] M. Kadansky *et al.*, "Tree-based reliable multicast (TRAM)," Internet draft, draft-kadansky-tram-02.txt, Jan. 2000.
[66] R. Yavatkar, J. Griffioen, and M. Sudan, "A reliable dissemination protocol for interactive collaborative applications," in *Proc. ACM Multimedia*, San Francisco, CA, Nov. 1995, pp. 333–344.
[67] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[68] S. G. Wilson, *Digital Modulation and Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[69] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC*, Geneva, Switzerland, May 1993, pp. 1064–1070.
[70] A. J. McAuley, "Reliable broadband communication using a burst erasure correcting code," in *Proc. ACM SIGCOMM*, Philadelphia, PA, Sept. 1990, pp. 297–306.
[71] L. Rizzo and L. Vicisano, "Effective erasure codes for reliable computer communication protocols," *ACM Comput. Commun. Rev.*, vol. 27, pp. 24–36, Apr. 1997.
[72] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, Vancouver, BC, Canada, Sept. 1998, pp. 56–67.
[73] S. Paul, *Multicasting on the Internet and its Applications*. Norwell, MA: Kluwer, 1998.
[74] C. Huitema, "The case for packet level FEC," in *Proc. IFIP Protocols for High-Speed Networks*, France, Oct. 1996, pp. 109–120.
[75] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," in *Proc. ACM SIGCOMM*, Cannes, France, Sept. 1997, pp. 289–300.
[76] D. Rubenstein, J. Kurose, and D. Towsley, "Real-time reliable multicast using proactive forward error correction," Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, Tech. Rep. 98-19, 1998.
[77] G. Carle and E. W. Biersack, "Survey of error recovery techniques for IP-based audio-visual multicast applications," *IEEE Network*, vol. 11, pp. 24–36, Nov./Dec. 1997.
[78] J. Crowcroft and K. Paliwoda, "A multicast transport protocol," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 247–256.
[79] S. Paul, K. K. Sabnani, and D. M. Kristol, "Multicast transport protocols for high-speed networks," in *Proc. IEEE Int. Conf. Network Protocols*, Boston, MA, Oct. 1994, pp. 4–14.
[80] L. H. Lehman, S. J. Garland, and D. L. Tennenhouse, "Active reliable multicast," in *IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1998, pp. 581–589.
[81] S. Ramakrishnan and B. N. Jain, "A negative acknowledgment with periodic polling protocol for multicast over LANs," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1987, pp. 502–511.
[82] M. Grossglauser, "Optimal deterministic timeouts for reliable scalable multicast," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1996, pp. 1425–1432.
[83] S. K. Kasera *et al.*, "Scalable fair reliable multicast using active services," *IEEE Network*, vol. 14, pp. 48–57, Jan.–/Feb. 2000.
[84] W. T. Strayer, B. Dempsey, and A. Weaver, *XTP: The Xpress Transfer Protocol*. Reading, MA: Addison-Wesley, 1992.
[85] J. Nonnenmacher and E. W. Biersack, "Optimal multicast feedback," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1998, pp. 964–971.
[86] ——, "Scalable feedback for large groups," *IEEE/ACM Trans. Networking*, vol. 7, pp. 375–386, June 1999.
[87] K. Calvert, M. Doar, and E. Zugera, "Modeling internet topology," *IEEE Commun. Mag.*, vol. 35, pp. 160–163, June 1997.
[88] J. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in *Proc. ACM SIGCOMM*, London, U.K., Aug./Sept. 1994, pp. 58–67.
[89] D. DeLucia and K. Obraczka, "Multicast feedback suppression using representatives," in *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997, pp. 463–470.

[90] ——, "A Multicast Congestion Control Mechanism for Reliable Multicast," Comput. Sci. Dept., Univ. Southern California, Los Angeles, CA, Tech. Rep. 97-685, 1997.
[91] B. Whetten, S. Kaplan, and T. Montgomery. (1994) A high performance totally ordered multicast protocol. [Online]. Available: http://www.nersc.gov/~jed/papers/HRM/references/highp-multi.ps
[92] C. Papadopoulos, G. Parulkar, and G. Varghese, "An error control scheme for large-scale multicast applications," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar.–Apr. 1998, pp. 1188–1196.
[93] M. Hofmann, "Enabling group communication in global networks," in *Proc. Global Networking*, Calgray, AB, Canada, June 1997, pp. 321–330.
[94] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the mbone multicast network," in *Proc. IEEE GLOBECOM*, London, U.K., Nov. 1996, pp. 94–99.
[95] S. Floyd *et al.*. (1995) A reliable multicast framework for light-weight sessions and application level framing. [Online]ftp://ee.lbl.gov/papers/srm1.tech.ps.Z
[96] C.-G. Liu *et al.*, "Local error recovery in SRM: Comparison of two approaches," Univ. Southern California, Los Angeles, CA, Tech. Rep. 97-648 , 1997.
[97] S. K. Kasera *et al.*, "Scalable reliable multicast using multiple multicast channels," *IEEE/ACM Trans. Networking*, vol. 8, pp. 294–310, June 2000.
[98] A. Koifman and S. Zabele, "RAMP: A reliable adaptive multicast protocol," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1996, pp. 1442–1451.
[99] R. Braudes and S. Zabele, "Requirements for multicast protocols," Network Working Group, RFC 1458, 1993.
[100] A. M. Costello and S. McCanne, "Search party: Using randomcast for reliable multicast with local recovery," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 1256–1264.
[101] Y. Gao, Y. Ge, and J. C. Hou, "RMCM: Reliable multicast for core-based multicast trees," in *Proc. IEEE Int. Conf. Network Protocols*, Osaka, Japan, Nov. 2000, pp. 83–94.
[102] T. Speakman *et al.*, "PGM Reliable Transport Protocol Specification," Internet draft, draft-speakman-pgm-spec-04.txt, 2000.
[103] M. Calderon *et al.*, "Active network support for multicast applications," *IEEE Network*, vol. 12, pp. 46–52, May/June 1998.
[104] D. L. Tennenhouse *et al.*, "A survery of active network research," *IEEE Commun. Mag.*, pp. 80–86, Jan. 1997.
[105] D. Wetherall, J. Guttag, and D. L. Tennenhouse, "ANTS: A toolkit for building and dynamically deploying network protocols," in *Proc. IEEE OPENARCH'98*, San Francisco, CA, USA, April 1998, pp. 117–129.
[106] A. S. Tanenbaum, *Computer Networks*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[107] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 158–173.
[108] S. Floyd and K. Fall. (1997) Router mechanisms to support end-to-end congestion control. [Online]. Available: ftp://ftp.ee.lbl.gov/papers/collapse.ps
[109] ——, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Trans. Networking*, vol. 7, pp. 458–472, Aug. 1999.
[110] L. Zhang *et al.*, "RSVP: A new resource reservation protocol," *IEEE Network*, vol. 7, pp. 8–18, Sept. 1993.
[111] S. Bhattacharyya, D. Towsley, and J. Kurose, "The loss path multiplicity problem in multicast congestion control," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 856–863.
[112] I. Rhee, N. Balaguru, and G. N. Rouskas, "MTCP: Scalable TCP-like congestion control for reliable multicast," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 1265–1273.
[113] B. Whetten and J. Conlan, "A rate based congestion control scheme for reliable multicast," *Tech. White Paper, GlobalCast Commun.*, Oct. 1998.
[114] S. Ha, K.-W. Lee, and V. Bharghavan, "A simple mechanism for improving the throughput of reliable multicast," in *Proc. IEEE Int. Conf. Computer Communications and Networks*, Boston, MA, Oct. 1999, pp. 372–377.
[115] S. Bhattacharyya, "Flow and congestion control for reliable multicast communication in wide-area networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, 2000.
[116] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. COM-29, pp. 954–962, July 1981.
[117] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephone Network*. Reading, MA: Addison-Wesley, 1997.
[118] H.-Y. Tzeng and K.-Y. Siu, "On max–min fair congestion control for multicast ABR service in ATM," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 545–556, Apr. 1997.
[119] J. Mahdavi and S. Floyd. (1997) TCP-friendly unicast rate-based flow control. [Online]. Available: http://www.psc.edu/networking/papers/tcp-friendly.html
[120] B. Braden *et al.*, "Recommendations on queue management and congestion avoidance in the internet," Network Working Group, RFC 2309, Apr. 1998.
[121] A. Legout, J. Nonnenmacher, and E. W. Biersack, "Bandwidth allocation policies for unicast and multicast flows," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 254–261.
[122] T. Jiang, M. H. Ammar, and E. W. Zegura, "Interreceiver fairness: A novel performance measure for multicast ABR sessions," in *Proc. ACM SIGMETRICS*, Madison, WI, June 1998, pp. 202–211.
[123] S. Cheung, M. H. Ammar, and X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 1996, pp. 553–559.
[124] T. Jiang, M. Ammar, and E. W. Zegura, "On the use of destination set grouping to imporve interreceiver fairness for multicast ABR sessions," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 42–51.
[125] S. McCanne and V. Jacobson, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM*, Palo Alto, CA, Aug. 1996, pp. 117–130.
[126] X. Li, S. Paul, and M. Ammar, "Layered video multicast with retransmissions (LVMR): Evaluation of hierarchical rate control," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1998, pp. 1062–1072.
[127] L. Vicisano and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1998, pp. 996–1003.
[128] A. Legout and E. W. Biersack, "PLM: Fast convergence for cumulative layered multicast transmission schemes," in *Proc. ACM SIGMETRICS*, Santa Clara, CA, June 2000, pp. 13–22.
[129] J. Byers, M. Luby, and M. Mitzenmacher, "Fine-grained layered multicast," in *Proc. IEEE INFOCOM*, AK, Apr. 2001, pp. 1143–1151.
[130] M. Luby, L. Vicisano, and T. Speakman, "Heterogeneous multicast congestion control based on router packet filtering," RMT Working Group, 1999.
[131] P. Assuncao and M. Ghanbari, "Multi-casting of MPEG-2 video with multiple bandwidth constraints," in *Proc. 7th Int. Workshop Packet Video*, Brisbane, Australia, Mar. 1996, pp. 235–238.
[132] S. J. Golestani, "Fundamental observations on multicast congestion control in the internet," in *Proc. IEEE INFOCOM*, New York, Mar. 1999, pp. 990–1000.
[133] T. J. Ott, J. H. B. Kemperman, and M. Mathis. (1996) The stationary behavior of ideal TCP congestion avoidance. [Online]. Available: ftp://thumper.bellcore.com/pub/tjo/TCPwindow.ps
[134] J. Padhye *et al.*, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM*, Vancouver, BC, Canada, Aug./Sept. 1998, pp. 303–314.
[135] D. Rubenstein, J. Kurose, and D. Towsley, "The impact of multicast layering on network fairness," in *Proc. ACM SIGCOMM*, Cambridge, MA, Aug./Sept. 1999, pp. 27–38.
[136] T. Jiang, E. W. Zegura, and M. Ammar, "Interreceiver fair multicast communication over the internet," in *Proc. Int. Workshop Network and Operating System Support for Digital Audio and Video*, Basking Ridge, NJ, June 1999, pp. 103–114.
[137] M. Handley and S. Floyd, "Strawman specification for TCP friendly (reliable) multicast congestion control (TFMCC)," Internet Reliable Multicast Group, 1998.
[138] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comput. Netw. ISDN Syst.*, vol. 17, no. 1, pp. 1–14, June 1989.
[139] D. Bansal and H. Balakrishnan, "TCP-friendly congestion control for real-time streaming applications," Massachusetts Inst. Technol., Cambridge, MA, Tech. Rep. MIT-LCS-TR-806, 2000.
[140] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, pp. 397–413, Aug. 1993.

[141] H. A. Wang and M. Schwartz, "Achieving bounded fairness for multicast and TCP traffic in the internet," in *Proc. ACM SIGCOMM*, Vancouver, BC, Canada, Aug./Sept. 1998, pp. 81–92.
[142] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," Network Working Group, RFC 2582, 1999.
[143] T. Montgomery, "A loss tolerant rate controller for reliable multicast," West Virginia Univ., Morgantown, WV, Tech. Rep. NASA-IVV-97-011, 1997.
[144] X. Li, M. H. Ammar, and S. Paul, "Video multicast over the internet," *IEEE Network*, vol. 13, pp. 46–60, Mar./Apr. 1999.
[145] M. H. Ammar and L.-R. Wu, "Improving the throughput of point-to-multipoint ARQ protocols through destination set splitting," in *Proc. IEEE INFOCOM*, Florence, Italy, May 1992, pp. 262–271.
[146] L. Vicisano and J. Crowcroft, "One to many reliable bulk-data transfer in the MBone," in *Proc. 3rd International Workshop High Performance Protocol Architectures*, Uppsala, Sweden, June 1997.
[147] L. Vicisano, "Notes on a cumulative layered organization of data packets across multiple streams with different rates," Univ. College London, U.K., Comput. Sci. Res. Note RN/98/25, 1998.
[148] R. Gopalakrishnan *et al.*, "Stability and fairness issues in layered multicast," in *Proc. Int. Workshop Network and Operating System Support for Digital Audio and Video*, Basking Ridge, NJ, June 1999, pp. 31–44.
[149] S. Bajaj, L. Breslau, and S. Shenker, "Uniform versus priority dropping for layered video," in *Proc. ACM SIGCOMM'98*, Vancouver, B.C., August/September 1998, pp. 131–143.
[150] R. Gopalakrishna *et al.*, "A simple loss differentiation approach to layered multicast," in *Proc. IEEE INFOCOM'2000*, Tel Aviv, Israel, March 2000, pp. 461–469.
[151] Z. Zhang and V. O. K. Li, "Router-assisted layered multicast," in *Proc. IEEE ICC*, New York, Apr.–May 2002, to be published.
[152] X. Li, S. Paul, and M. Ammar, "Multi-session rate control for layered video multicast," in *Proc. Symp. Multimedia Computing and Networking MMCN'99*, San Jose, CA, Jan. 1999.
[153] S. Jagannathan, K. C. Almeroth, and A. Acharya, "Topology sensitive congestion control for real-time multicast," in *Proc. Int. Workshop Network and Operating System Support for Digital Audio and Video*, Chapel Hill, NC, June 2000.
[154] C. Albuquerque, B. J. Vickers, and T. Suda, "Multicast flow control with explicit rate feedback for adaptive real-time video services," in *Proc. Conf. Performance and Control of Network Systems II*, Boston, MA, Nov. 1998, pp. 110–121.
[155] B. J. Vickers, C. Albuquerque, and T. Suda, "Adaptive multicast of multi-layered video: Rate-based and credit-based approaches," in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar./Apr. 1998, pp. 1073–1083.
[156] D. Sisalem and A. Wolisz, "MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments," in *Proc. IEEE/IFIP IWQoS*, Pittsburgh, PA, June 2000, pp. 65–74.
[157] A. Legout and E. W. Biersack, "Pathological behaviors for RLM and RLC," in *Proc. Int. Workshop Network and Operating System Support for Digital Audio and Video*, Chapel Hill, NC, June 2000, pp. 164–172.
[158] X. Li *et al.*, "Layered video multicast with retransmissions (LVMR): evaluation of error recovery schemes," in *Proc. Int. Workshop Network and Operating System Support for Digital Audio and Video*, St. Louis, MO, May 1997, pp. 161–172.
[159] S. Keshav, "Congestion control in computer networks," Ph.D. dissertation, Elect. Eng. Comput. Sci. Dept., Univ. California, Berkeley, CA, 1991.

Victor O. K. Li (Fellow, IEEE) was born in Hong Kong in 1954. He received the S.B., S.M., E.E., and Sc.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 1977, 1979, 1980, and 1981, respectively.

He joined the University of Southern California (USC), Los Angeles, CA, in February 1981 and became a Professor of Electrical Engineering and Director of the USC Communication Sciences Institute. Since September 1997, he has been with the University of Hong Kong, Hong Kong, where he is Chair Professor of Information Engineering and Managing Director of Versitech Ltd., the technology transfer and commercial arm of the university. He also serves on various corporate boards. Sought by government, industry, and academic organizations, he has lectured and consulted extensively around the world, given keynote addresses, and served on advisory boards of numerous international conferences. He is very active in the research community and has chaired various international conferences and served on the editorial boards of various international journals. His research interests include information technology, including high-speed communication networks, wireless networks, and Internet technologies and applications.

He was a Distinguished Lecturer with the University of California at San Diego, the National Science Council of Taiwan, and the California Polytechnic Institute.

ZaiChen Zhang (Student Member, IEEE) was born in Nanjing, China, in 1975. He received the B.S. and M.S. degrees in radio engineering from Southeast University, Nanjing, China, in 1996 and 1999, respectively. He is currently working toward the Ph.D. degree in electrical and electronic engineering, the University of Hong Kong, Hong Kong, China.

From July 1997 to July 1998, he was also a Research Scientist with STC-HP Joint R&D Center, Beijing, China. His current research interests include Internet multicast and wireless networks, with emphasis on multicast flow and congestion control.