

The QuORRUM Protocol: Efficient Tree Repair for Qualification-Based Multicast*

L. A. Flynn
Bell Laboratories, Lucent Tech.
67 Whippany Rd.
Whippany, NJ, 07981 U.S.A
laflynn@lucent.com

H. P. Dommel
Computer Engineering Department
Santa Clara University
Santa Clara, CA U.S.A
hpdommel@scu.edu

Abstract

Increasing sophistication of Internet services requires more refined methods of service differentiation in the delivery of data and media flows. Earlier work on Quality-of-Service (QoS) routing has focused on the network mechanics to find the best path for admitted connections under specific QoS constraints, and to maximize global resource utilization. We look at the related notion of qualification-based routing, where routers check on the fulfillment of one or more constraints before handling a particular packet flow. Packets are demarcated and evaluated by qualifiers in their headers, which may indicate for example an authorization level in secure transmissions, or a payment receipt for a media-on-demand application. In particular, this paper explores the coordinative processes for tree formation and repair in qualification-based multicast routing, with the goal to leverage group-centric communication for future mass media Internet applications. We introduce the QuORRUM protocol to implement qualifier-driven sparse-mode tree building and repair and show that it performs better than other protocols in terms of bandwidth usage and repair times.

Keywords: *Multicast tree formation and repair, sparse-mode protocols, QoS routing.*

1. Introduction

The next generation of applications and services in broadband and high performance networking, both in wired and ad hoc contexts, creates new challenges in media transmission in conjunction with effective provision of Quality-of-Service (QoS). In particular, multicasting has been deemed in the past as a panacea for scalable and

resource-aware dissemination of large volumes of data. Various point-to-multipoint packet delivery schemata have been suggested in the past to make group communication in the Internet as practical and efficient as the dominant method of unicasting. Multicast routing inherently shows various problems with regard to persistence, visibility, and prevalence of routes, affecting the monitoring [12] and connectivity of multicast networks [10]. One such problem is multicast tree formation and repair, where numerous solutions have been proposed to make dissemination topologies adaptive and resilient to the dynamics of subscriber groups and networks.

In this paper we examine the marriage of multicasting with QoS [13], with special focus on sparse-mode solutions for networks, where packets are only forwarded by routers when fulfilling one or more qualification criteria. We define qualification-based multicast as a network transmission mode where routers and data streams possess qualifiers, and packets are only forwarded among multicast-enabled routers capable of handling qualified data streams. In that sense, such multicast is a generalized, elegant notion of QoS routing, where routers forward only qualified packets to other routers, thus using the dynamic ability to observe specific and interrelated service constraints, whether traffic is of a real-time or differently constrained nature. Qualifiers are hence an extensible methodology for conditional packet routing and may represent a security level for Virtual Private Networks [4], a payment receipt for a pay-for-reception mediacast [3, 14], or any other certificate validating that a router is authorized to carry a given type of traffic.

The rest of this paper is structured as follows: Section 2 summarizes the research context. Section 3 discusses repair approaches in other multicast protocols. Section 4 describes three different versions of the QuORRUM protocol and novel repair methods. Section 5 presents a performance analysis for QuORRUM. Section 6 summarizes our contributions and maps out future work.

*This research was funded by the Defense Advanced Research Projects Agency (DARPA) under Grant No. N6601-00-1-8942.

2. Related Work

Multicast has been hailed in the past as the most effective solution to cope with volume transmissions to users with similar interests. With Layer 3 routing support, such group-centric communication could naturally find its most effective realization in the network core, however, various deployment problems [7] have limited the spread of such native multicast to domain-specific testbed scenarios, previously in the Mbone and currently primarily in the Internet 2 [2]. Traditional routing for group communication, as with DVMRP or MOSPF, assumes uniform distribution of group members in a given area and plentiful bandwidth. Sparse "rural" distribution of members is hence not properly handled, as periodic membership reports are transmitted over links irrespective of group status. Likewise, dense-mode "urban" protocols require a more concentrated and effective method to handle large and dynamic membership.

Multicast routing is accordingly typically is classified into sparse mode (SM) versus dense mode (DM) operations, with cross-over points where protocol modes could be selected according to a group density metric. PIM-SM [8] is a typical SM multicast protocol, where joining nodes send a graft packet along a reverse path to the multicast core. DM multicast, as represented by the PIM-DM protocol [6], periodically floods the entire network of reachable routers with a data stream and prunes along reverse paths all links which are not part of the receiver tree, which consumes much bandwidth. SM protocols are hence more scalable than DM protocols as they limit network disturbance to the part of the network where actual receivers reside.

Adding one or multiple QoS constraints to routing has previously been tackled from many different angles [5] and implemented in various network- or edge-based routing mechanisms. Qualification-based multicast focuses on whether packets fulfill collective transmission constraints derived from user actions, group identity or network conditions, in contrast to QoS routing where packet flows are admitted if network routes with sufficient resources for the requested parameters can be found. Supporting qualifiers in place of QoS parameters in qualification-based routing based on DM multicast flood and prune mechanics contradicts the intuition to minimize resource consumption.

An algorithm for rapid and scalable receiver-initiated multicast tree repair is hence needed, which does not require a unicast table at routers to track qualifiers. Instead of using a DM solution, a SM method for qualifier-driven tree formation and repair will require less bandwidth and processing overhead, due to receivers initiating joins on demand. As we will see, retrofitting legacy routing mechanisms with effective repair mechanisms for qualification-based routing has various shortcomings.

3. Tree Repair with Legacy Multicast

We review the characteristics of existing tree repair mechanisms for multicast to set the stage for a novel SM protocol exhibiting improved repair performance. Tree repairs are necessary when an on-tree node becomes unqualified or otherwise unavailable. Previous protocols, except for on-demand multicast [11], which uses expanding ring search, require a separate unicast table to be built for each qualifier in order for a multicast to reach to all member nodes. Unicast routing tables must be updated frequently to maintain integrity.

One way to rebuild the multicast tree in DM is to use timers at each node. At timeout, a node deletes its old multicast parent and children entries as well as its prune timer information for this multicast. The next multicast packet received for this group will cause the node to find a new parent and new children for group, subsequently sending multicast packets to all its new children.

A second DM method is based on the idea to rebuild the tree after floods. Upon receiving a unique new type of flood packets, a node removes its multicast routing information about earlier prunes, children, and the parent for a particular multicast group. Next, the node replaces the entry with new information based on the neighbor which has sent the first special flood packet with the next sequence number chosen as new parent. All other qualified neighbors are then considered to be multicast tree children. Using this method, nodes need not maintain any state about prune timers for any neighbors.

By rebuilding the tree using either DM method, nodes can correctly accept previous children as new parents, rather than being cut off the tree if a tree ancestor becomes unqualified. Both repair methods may cause floods, which waste bandwidth if occurring frequently. Repairs may also incur significant repair delays as a function of the frequency of flooding. In contrast, rare flooding may cause lengthy off-tree times for affected nodes. We surmise that more effective solutions are possible by using a sparse mode strategy to heal breaking points in a multicast tree by suppressing floods and limiting the scope of rejoin requests. Such repair tactics are based on focused rejoins bridging broken tree branches on demand, vs. the unfocused reflooding of network sections that may or may not be affected.

We make these universal assumptions about qualifier setup and exchange for repair protocols: Routers are assumed to operate correctly and information is assumed to be stored without errors. Information regarding qualification status is periodically exchanged among neighboring routers. When a neighbor becomes unqualified, routing information is changed so multicast packets are no longer communicated to that neighbor. Each qualifier is uniquely tagged across a network and transmission sessions.

4. QuORRUM Protocol

In this section we discuss the QuORRUM (*Qualified Ordered Rapid-repair Receiver-initiated Unified Multicast*) protocol, which implements a sparse-mode algorithm for efficient creation and repair of qualification-based multicast trees. QuORRUM is a receiver-initiated protocol, reconstructing a delivery tree so that the least-latency tree structure is preserved, and is based on four central ideas: 1) A novel flood suppression technique simultaneously limits repair bandwidth independent of the number of nodes trying join (or rejoin) the multicast tree. 2) Entire subtrees of routers are immediately informed when their reverse path becomes unqualified or unavailable. 3) Requests to join or repair reach the multicast source, and permit rekeying the tree when necessary while preserving latencies and the existing tree precedence structure. 4) Requests to join or rejoin follow a reverse tree path after reaching the tree, thus limiting the scope of the request.

4.1. Definitions and Mechanisms

To elucidate the operation of QuORRUM, we introduce the following definitions and characterize various complex repair mechanisms.

4.1.1 Special Terms

Best-effort multicast does not make any extra service guarantees for the delivery of packets in a multicast transmission. A *qualified* multicast uses specific processing constraints or qualifications tested by routers to forward *qualified* packets. A *subcast* is a multicast to a subset of member nodes, which is practical for select packet retransmissions to subsections of a tree. *Direct Orphan* refers to a node previously on the multicast tree whose direct parent node has become unqualified, or failed, or the link to the parent has failed. An *orphan* node is a potential receiver that became temporarily or permanently disconnected from the tree, after an ancestor node became unqualified. A *dead node* refers to a newly unqualified, failed, or unreachable node in the multicast tree. Finally, a *CAck* is a Cumulative Acknowledgement which indicates that all descendants on the multicast tree have received and acknowledged a packet. Fast repair is possible by using several key mechanisms.

4.1.2 Repair Mechanisms and Data Structures

A **Controlled Cast** (CC) is a special type of multicast, which is forwarded to all qualified links except the link that the packet was received on first, starting from the joining or rejoining node. When any node on the existing multicast tree is reached, the CC is forwarded to the multicast

source. Nodes keep a record of their parent and children on a particular CC tree, identified by a tuple with the joining-node identifier and the flood-sequence number. When a CC packet is received from a non-parent on a particular CC tree, the packet is met with a CACK. If the Controlled Cast is best-effort, the packet is simply dropped.

A **Not-Your-Parent** (NYP) subcast is a heuristic used to accelerate how all multicast receivers learn about their current status. Whenever a node that is the direct multicast child of a node realizes that its parent is no longer qualified, it sends a NYP subcast to its tree children. The subcast reliably propagates knowledge of tree separation to the node's descendants, so that they can quickly find new ways to reach the multicast tree if a rejoin is possible.

An **Alert Flood** (AF) is a unique and reliable CC from an orphan to all neighbors to jumpstart an emergency tree repair. Once the AF reaches leaf nodes, CACKs are sent from the tree edge and are eventually received by each interface of the orphaned node that AF packets were sent out on. At this point the orphan starts a triggered flood (see below) to attempt to rejoin the tree. All nodes, which could be parents or ancestors of the orphaned nodes on a rebuilt multicast tree, become aware of the repair process and create memory structures using information from AF packets. AF packets are characterized as a data structure by a tuple (*orphanID*, *sequence number*). When a node sends or receives a first AF, it instantiates a repair data structure with a sequence number and the parent ID on the AF CC tree for this node, and IDs of neighbors that this node forwarded the AF to, or which have sent AF CACKs. All of this information allows the node to correctly route repair packets.

Triggered flood (TF) is a special type of limited flooding of the qualified nodes within the network which reaches only nodes already on the multicast tree and nodes reached by an *Alert Flood Controlled Cast* (AF CC). Possible repair paths to the orphans will be traversed by a TF as potential repair path. The first path to reach the orphan will be chosen as the repaired path. A TF packet includes the sequence number of this triggered flood and the orphan's id. If the node had previously been on the multicast tree, it sends the TF packet to all its tree children and out to all interfaces on which it received an AF packet. If the node was not on the tree before it received the TF, it records the interface from where it receives the first TF packet from its new parent on the general multicast tree. It also notes all interfaces from which it receives AFs or sends AFs to, and ultimately sends out a TF packet to all its children on the multicast tree for which it has state. This process percolates recursively through the tree until all orphans on a qualified path from the multicast source have been reached.

A **Pathfinding Flood** (PF) is a unique and reliable CC of pathfinding packets from an orphan, which starts the PF

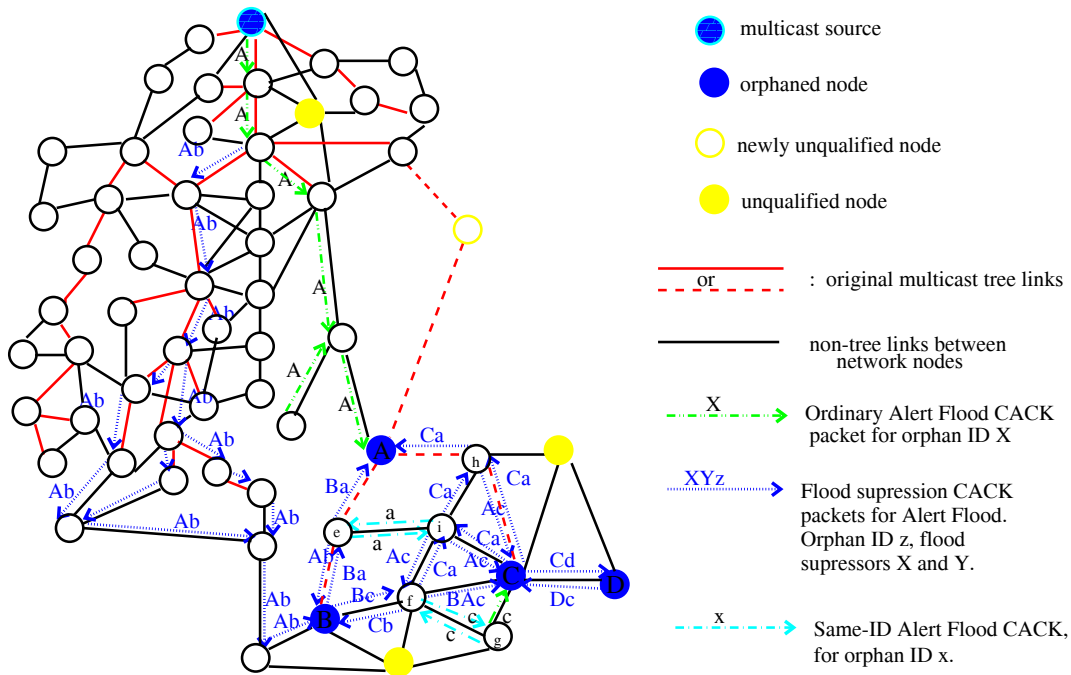


Figure 1. Flood Suppression of Alert Flood and Alert Flood CACKs(QuORRUM methods 1, 2, and 3)

immediately after receiving NYP CACKs from all neighbors it sent a NYP packet to. PF traverses nodes exactly the same as the standard AF, but the PF CACKs travel differently than standard AF CACKs. PF CACKs are called *path-found* packets, which indicate that an unbroken path has been found from the orphan to the multicast source. When a node receives the first path-found packet, it forwards the packet to all other neighbors which had sent this node a pathfinding packet, in contrast to standard AF CACKs. A path-found notification places an off-tree node back on the multicast tree and regards all neighbors as children.

Flood suppression (FS) minimizes bandwidth use by AF and AF CACKs when there are multiple orphans attempting multicast tree repair. The FS packet is a special AF CACK with a field indicating the original orphanID this node received. Using FS, at most only two AF packets and two AF CACK packets can traverse each link. Flood suppression also minimizes the number of triggered floods. Any node holding a tree repair data structure for one orphanID and receiving an AF with a different orphanID immediately sends a FS packet. Any node which receives a FS packet marks the flood suppressor orphanID(s) in its tree repair data structure, and which neighbor sent the AF CACK. When this node received all expected AF CACKs, it sends a FS packet to its parent on the AF CC tree. This continues until the orphan which originated the AF has received all of its AF CACKs, of which one must be a FS packet with at least one flood suppressor orphanID.

When an orphan receives its first path-found packet, it unicasts a **Start Triggered Flood** (STF) packet to the multicast source, which can be done via the reverse path back to the source through qualified nodes only. STF packets, like TF packets, contain an outer list of orphanIDs, and an inner list of suppressor orphanIDs for each orphanID in the outer list. An orphan receiving any FS packets only rejoins the tree if its orphanID is listed in the outer orphanID list of the TF. All this node's suppressor orphanIDs must be listed in the outer orphanID list of the triggered flood. Orphans which were flood-suppressed ultimately verify that their tree repair process reached all the nodes that would have been reached if no FS of the AF had occurred. The multicast source, when receiving a path-found packet from a neighbor, adopts the path to that neighbor as a valid path for a multicast stream. When the source timer expires, the source sends a TF. The purpose of the timer is to minimize TFs, in case other orphans issued STF packets. Disregarding variations due to temporal packet queue length differences, the rebuilt tree has the same shape as the original tree if no node had become unqualified. The rebuilt multicast tree consists of the lowest latency paths considering all the possible repair paths that would have been explored if each of the orphans had created a separate AF without FS.

Figure 1 depicts a snapshot of FS CACKs phasing through a sandbox network to rebuild the multicast tree. Without FS, 31 links would have 4 AF packets traverse each link. For every AF packet, an AF CACK traverses - in this

case 124 AF CACKs would pass through the network without FS and 51 links would have each processed 4 different TF packets. OrphanID C's AF is suppressed by orphans $\{B, A, D\}$. OrphanID B's AF is directly suppressed by orphans $\{A, C\}$. OrphanID A's AF is suppressed by $\{B, C\}$. OrphanID D's AF is suppressed by $\{C\}$. When a TF arrives at each orphan, it checks to see if it can join. Here this is the case if nodes $\{B, A, D, C\}$ are listed in the flood packet as having sent STF packets in mutual suppression by other orphans.

Same-orphanID CACKs (SO) are necessary in order to make sure that AFs and TFs stop somewhere. Figure 1 shows SO CACKs between nodes e and i, and between nodes f and g. We differentiate between two types of SO CACKs: *AF SO CACKs* occur when a node returns an Alert Flood CACK immediately for the subsequent Alert Flood packet. However, the node does not return an AF CACK to the neighbor which sent the original AF packet until this node has received AF CACKs from each of the neighbors it forwarded the original AF packet to. *TF SO CACKs* occur when a node returns a TF CACK immediately for the subsequent triggered flood packet. However, the node does not return a TF CACK to the neighbor which sent the original TF packet until this node has received TF CACKs from each of the neighbors it forwarded the original TF packet to.

Usage of keys [9] is suggested to ensure that multicasting is protected in the following ways: no prior encrypted multicast transmission should be readable by newly joined nodes, and no current multicast transmission should be readable by nodes which are currently off the official multicast tree. When the receiver set changes, all affected multicast keys must be changed.

4.2. QuORRUM Repair Methods

We discuss how these concepts fit with the operational semantics of QuORRUM, introducing three flavors of QuORRUM implementing different repair strategies, as compared in Figure 2.

Method 1 uses the TF only to update the nodes in the tree with the new repair sequence number. It begins with orphaned nodes subcasting NYP with an AF, where FS minimizes bandwidth consumption for multiple stranded nodes. As soon as the AF reaches the source, the multicast stream travels immediately after the path-found packet on the reverse path toward the orphaned node, which then rejoins and initiates a TF when it receives all AF CACKs. When a node receives a pathfinding packet but has already received a path-found packet with the same repair sequence number, it immediately replies with a path-found packet regardless of the orphanID.

	QuORRUM Method 1	QuORRUM Method 2	QuORRUM Method 3
Latency	lowest	medium	highest
Bandwidth	Higher than method 3	Higher than method 3	Very low
Priority Structure of Tree Maintained?	no	yes	yes
Method:	When ontree node receives alert flood, immediately sends mcast stream.	One triggered flood starts the multicast stream behind it.	Two floods, 2 nd only on repaired path. Stream only on repaired path, always.

Figure 2. QuORRUM Methods 1, 2, and 3

Method 2 uses only one flood for rebuilding, where the first TF packet is followed by the multicast data stream. The stream may travel over some links that eventually will be pruned.

Method 3 uses two TFs for rebuilding the tree, beginning with children of a dead node subcasting NYP. An orphan triggers a multicast flood from the source via an AF, which travels along all on-tree nodes. When the first flood packet arrives at the orphan, it selects the sending neighbor as its new multicast parent. Recursively, each successive node then sends a special flood CACK to its tree parent. When the flood reaches the leaves of the repair tree, the source sends another flood to integrate orphaned nodes into the tree, and on-tree nodes begin to send multicast data packets to new tree links. Packets will hence flow only on new and legitimate paths, which limits bandwidth use.

Using any of the repair methods, at the time the last flood has reached all leaves, all reachable orphans have correctly joined the multicast group through a parent, and marked all other neighbors with valid qualifiers as children.

5. Performance Analysis

Our performance analysis looks at the speed of multicast tree repairs, bandwidth consumption with control and data packets, and the cost of flooding. We contend that all three QuORRUM methods perform better than alternative DM approaches. The nomenclature is summarized in Table 5.

5.1. Repair Time

In the worst case the longest tree path is a chain of all nodes, hence $t_P = t_l + Q_m$ is the maximum time for a packet to travel such path in the network. Using flood

f	Number of links from which prunes were initiated
F_d	Average frequency of node death in a network
i	Immediately pruned stream over link
L	Number of qualified links
$M(U_t)$	Maximum time to update unicast table
N	Number of nodes in the network
N_x	Number of links pruned during time period x
o	Full stream during neighbor prune time
P	Pruned stream over link
Q	Number of qualification levels
Q_m	Maximum queuing delay for a packet
t	Time between prune timeouts for a periodic prune
t_l	Latency required for packet to traverse a link
t_P	Overall travel time for packet on average path
T_R	Overall time to repair tree
U_b	Maximum bandwidth to update the unicast table
U_t	Time required to update unicast table
W_f	Time period for source to wait before starting flood
Z	Full data stream (eventually dropped) until a downstream node completes pruning

Table 1. Analysis Nomenclature.

suppression, QuORRUM methods 1-3 incur repair time T_R which is composed of tree traversal times for NYP/NYP CACKs ($2N * t_P$), for AF/AF CACKs ($4N * t_P$), for STFs ($N * t_P$), the time a source must wait before starting flooding (W_f), and the traversal times for TF/TF CACK ($2N * t_P$), and the final TF ($N * t_P$), hence

$$T_R = 10N * t_P + W_f \quad (1)$$

In contrast, PIM-DM multicast require that prune timers expire and a node's unicast table be updated before a node will forward or receive multicast packets from a different neighbor, incurring a maximum repair time of $M(U_t) + t + N * T_P$, where t refers to the prune timer at the orphaned node's new parent. A new parent waits for its prune timer to expire before sending data packets to the orphan. Assuming that unicast table updates are the most time-consuming portion in PIM-DM repair, QuORRUM repair methods are hence much faster on the average. Method 2 has the smallest maximum repair time due to its immediate reactivity. Method 3 is slowest among all methods and may take longer than the PIM-DM repair which sends data packets after all unicast tables have been updated.

5.2. Repair Bandwidth

Repair bandwidth refers to the maximum bandwidth required to reconnect an orphan node. If only one repair path exists for an orphan, QuORRUM Methods 1 and 2 use slightly less bandwidth. Method 3 uses the least bandwidth for repair except when other methods use less. In the worst case, QuORRUM methods 1-3 use small multiple of one regular PIM-DM flood bandwidth for repair. While PIM-DM needs a multiple of the number of qualifiers as bandwidth to build one routing table, QuORRUM does not. Data packets flow along all qualified paths reachable from the multicast source, which is the same for QuORRUM or

PIM-DM in a nonqualified network. When group membership is more static and disqualification is infrequent, periodic flooding uses a lot of network bandwidth with no gain. QuORRUM repair is only activated when an orphan needs to get back on-tree, and packet streams started by a tree repair process only travel along paths that can possibly reattach an orphan to the multicast tree. In particular, in method 3 data packets never flow beyond valid multicast links.

QuORRUM control packets include NYP, NYP CACKs, AFs, AF CACKs, TFs, TF CACKs, and STFs, traversing a limited range of network links only once in the absence of packet loss. The worst case for NYP occurs when a dead node has the entire rest of the tree below it, which incurs $2 * N$ for packet traversals across a link. The worst case for AFs, AF CACKs, TFs, or TF CACKs is the case where every single qualified link in the network is reached. The first TF or TF CACK traversal costs $2 * L$. The Alert/AlertCACK maximum uses four times the number of qualified links, $4 * L$. For STF the worst case is that the unicast path from orphan to multicast source traverses all the network nodes, and that all nodes in the network were orphans, that is $(N * (N + 1))/2$. If all nodes were orphans, one node initiates the longest path and all subsequent orphans connect to it. For methods 1 and 2 the maximum number of data packets traversing a maximum number of links to be pruned is $\sum_{x=1}^{L-1} x * Z$.

In contrast, PIM-DM traversals may reoccur periodically and from sending the first data packet until a node receives a prune, many multicast data packets may have been sent. If all other nodes in the network are on-tree, then DM Repair Methods use slightly less bandwidth. Only it takes little bandwidth to update the unicast table after a node death, PIM-DM will uses less bandwidth for tree repair. For PIM-DM to allow nodes with qualified paths to rejoin the multicast, there must be one unicast table per qualification level. A PIM-DM multicast requires a node's unicast table to be updated before a node will forward or receive multicast packets from a different neighbor. The worst case for DM methods occurs when all nodes are not receivers, all qualified links are reachable along qualified paths from the multicast source, and the orphan has a link directly to the multicast source. In this case, flooding wastes bandwidth on all qualified links except one. At every flood period, the repair bandwidth is $(L - 2) * P + i * f + \sum_{x=1}^{\infty} (x - 1) * N_x * o$. Until the orphaned node's unicast table has been updated, bandwidth is used by data packets along non-tree links not accepted by the orphan. The maximum bandwidth at every flood period, with $Q * U_b$ denoting the cost to update all the unicast tables based on qualification level, is

$$Q * U_b + (U_t/t) * \{(L - 2) * P + i + \sum_{x=1}^{L-1} x * o\} \quad (2)$$

5.3. Simulations

Five repair methods were simulated on 250 different combinations of parameters forming multicast trees in networks with 10 to 90 nodes. The percentage of nodes receiving a multicast varied between 20, 40, 60, 80, and 100 percent. For each network size, five different topologies were randomly generated using open-source software [15], for a total of 90 topologies. All links were duplex with 1.5Mb capacity, and latencies determined by a geometric random process. Simulations ran for 12.0 (for 70 nodes and less) or 13.0 seconds (for 80 nodes and more), since the larger networks required more time to build the trees. Multicast traffic was constant bit rate (CBR), starting at simulation time 0.25 sec. All nodes were initially considered qualified. Larger networks had 1.61 seconds to build trees before a node became disqualified, and smaller networks were given only 0.61 seconds. Time for repairs after disqualification was the same in the 12.0 runs as for the 13.0 second runs. Orphans may have a qualified path remaining to the source, or they may be cut off from the source for good. If there is a path remaining to the source, each of the repair methods will eventually rejoin this orphaned node to the multicast data stream. Repair protocols handle dropped control packets with timers and subcasting of duplicate control packets to links lacking CACKs. Loss of control packets significantly increases time to repair. Hence links use Class Based Queueing (CBQ) with highest priority for new control packets, middle priority for grafts and prunes, and least priority for CBR traffic. Packet sizes for prunes and grafts were 1K (ns2 [1] default), CBR multicast data packets were 500 bytes and new control packets were also 1K, transmitted at intervals of .005ms.

For QuORRUM, in low-joined-ratio multicast groups in the worst case repair requests travel the full length of the qualified network. Method 1 takes an average maximum repair time which is $O(N)$, more similar to the overhead of repair methods 2 and 3. Method 3 results in a slightly lower average number of data packets received by joined nodes; this is due to the longer repair time before an orphan rejoins the multicast. With higher join-percentages, Method 1 generally has the fastest repair time, and Method 2 repairs faster than Method 3, since Method 3 requires an additional triggered flood before stranded nodes start receiving the multicast. Here the average path that a request for a repair must travel before reaching an on-tree node is much shorter than with lower joined-percentages. For Methods 1-3, prunes never time out. Additionally, no floods occur unless triggered. For all repair methods, every 0.2 sec. nodes discover if the qualification status of their neighbors has changed, and stop communication with that neighbor. Since random nodes could become unqualified, some leaf nodes did not cause any repair to be executed. Control packet over-

head, which in Methods 1,2 and 3 is less than observed for DM methods, includes prunes along with the new additional alert packets. Smaller flood intervals result in larger wasted bandwidth but shorter times for nodes to come back on-tree, opposite to longer flood intervals. Figure 3 shows the average off-tree times for nodes which for QuORRUM Methods 1-3 is the same as the repair time. The shown graphs only report on five topologies, however results were similar for all other topologies.

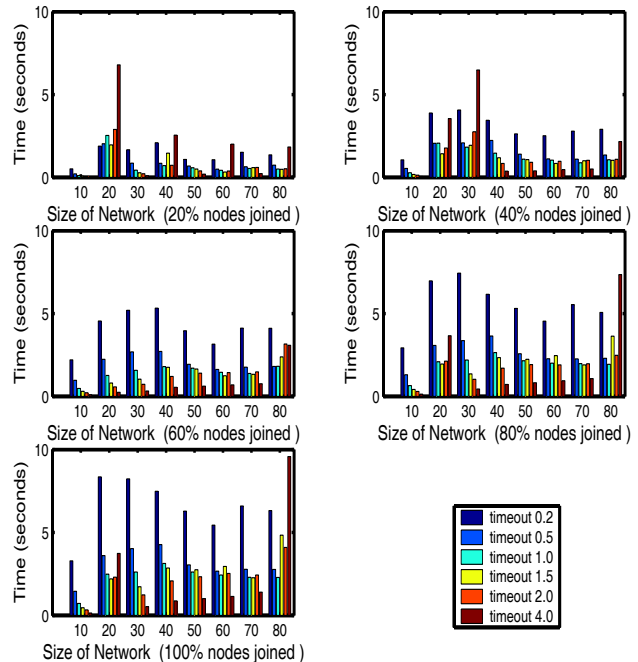


Figure 3. Total off-tree time for joined nodes in Topology 1 for standard flood and prune repair with different flood timeouts

DM repair methods essentially rely on flooding the network with data and control packets. Hence percent-joined ratios do not affect time to repair and repair latency is traded for bandwidth wasted depending on flood timeouts. The total number of control packets for DM methods always increases with time. For DM repair, the difference in time between a new data packet's reception and the previous one is added to this node's time off the tree. The shorter the flood timer, the higher the number of dropped data packets. Due to the standard timeout of 0.5 seconds for the PIM-DM protocol, some dense mode repairs are faster than sparse-mode repairs. The trade-off between shorter repair time vs. higher bandwidth, and lower bandwidth vs. longer repair time is visible. PIM-DM-style repair methods waste more bandwidth when no repairs are needed.

Over time the useful packet ratio of QuORRUM repair methods improve, while the useful packet ratio of the DM

repair methods remain constant. Given that J_p is the total of useful packets, or multicast stream packets received by joined nodes, and A_p is the total of all kinds of packets received by all nodes, the fraction J_p/A_p gives the percentage of useful traffic. With higher average joined ratios for the multicast, the fraction J_p/A_p gets higher since losses due to a node death tend to be a smaller percentage of the total multicast traffic in the network.

QuORRUM maintains a relatively low number of dropped packets even with increasing network sizes. Repairs only use extra bandwidth for control and data packets until the tree rebuild has completed such that the multicast uses less network bandwidth over time. This makes QuORRUM an effective choice when scalability and low repair bandwidth are key factors. We reach the conclusion that QuORRUM repair methods consume less average bandwidth in terms of data and control packets.

6. Conclusions

We have outlined the challenges of designing a sparse-mode multicast tree repair algorithm for qualified networks and introduced a new genre of qualification-based build and repair protocols for multicast routing. In a sense, qualifiers can be viewed as a more general approach to implementing conditional and selective packet forwarding and handling. Qualification-based multicast can hence be seen as a network-centric methodology to observe QoS constraints in transmission paths in absence of integrated or differentiated services. Routers must be enabled to handle qualification-driven traffic by inspecting qualifiers in addition to legacy header information.

In particular, we discussed the novel receiver-initiated QuORRUM protocol in three algorithmic variants to support more efficient tree formation and rapid repair for selective, qualifier-driven multicasts. While one unicast table per constraint is a way to allow nodes to send a sparse-mode graft packet to join the multicast, this solution does not scale for a larger number of QoS parameters. Our novel protocol family also compares favorably to popular sparse-mode solutions such as PIM-SM, reacting quickly to network changes while incurring low bandwidth and computational overhead. The choice of a particular QuORRUM protocol must be based on the bandwidth available in a network and the tolerable repair delay. With large available bandwidth, QuORRUM variations 1 or 2 would be the better choice, while QuORRUM 3 would be ideal in limited-bandwidth networks.

Future work targets the interplay with higher layers, for example performance correlations with reliable multicasting and particular application types. In addition, we plan to investigate scalability issues in conjunction with a hierarchical version of QuORRUM, and a hybrid version

of the protocol which employs dense-mode tree fixes together with occasional sparse-mode QuORRUM-style or MAODV-style repairs. Our general objective is to provide a solid framework and practical toolbox for implementing and deploying qualification-based multicast with robust and efficient protocols.

References

- [1] Network simulator ns-2. <http://www.isi.edu/nsnam/ns/>, 2002.
- [2] K. Almeroth. The evolution of multicast: From the Mbone to inter-domain multicast to Internet2 deployment. *IEEE Network*, 14:10–20, Jan./Feb. 2000.
- [3] K. C. Almeroth and M. H. Ammar. The use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE Journal of Selected Areas in Communications*, 14(6):1110–1122, 1996.
- [4] W. Arbaugh, J. R. Davin, D. J. Farber, and J. M. Smith. Security for Virtual Private Intranets. *IEEE Computer*, 9:48–54, September 1998.
- [5] S. Chen and K. Nahrstedt. An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions. *IEEE Network Magazine, Special Issue on Transmission and Distribution of Digital Video*, 1998.
- [6] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, D. Meyer, and L. Wei. Protocol Independent Multicast (PIM), Dense Mode Protocol Specification. *draft-ietf-idmr-pim-dm-06.txt*, August 1997.
- [7] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment Issues for the IP Multicast Service and Architecture. *IEEE Network*, 14:88–98, January 2000.
- [8] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol independent multicast-sparse mode (PIM-SM): Protocol specification. *IETF RFC 2117*, June 1997.
- [9] X. Li, Y. Yang, M. Gouda, and S. Lam. Batch Rekeying for Secure Group Communications. In *Proceedings of the ACM SIGCOMM 2001*, pages 525–534, August 2001.
- [10] P. Rajvaidya and K. Almeroth. Analysis of routing characteristics in the multicast infrastructure, 2003.
- [11] E. Royer and C. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing. *draft-ietf-manet-maodv-00.txt*, July 2000.
- [12] K. Sarac and K. Almeroth. Supporting multicast deployment efforts: A survey of tools for multicast monitoring. *Journal of High Speed Networking—Special Issue on Management of Multimedia Networking*, Mar. 2001.
- [13] B. Wang and J. Hou. Multicast routing and its QoS extension: Problems, algorithms, and protocols. *IEEE Network*, 14, Jan. 2000.
- [14] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal of Selected Areas in Communications*, 14(7):1228–1234, 1996.
- [15] A. Zegura, K. Calvert, and S. Bhattacharjee. How to Model an Internetwork. In *Proceedings of IEEE Infocom*, pages 594–602, March 1996.