A Comparison of Ring and Tree Embedding for Real-time Group Multicast

M. Baldi
Politecnico di Torino
Corso Duca degli Abruzzi, 24
10129 Torino - Italy
mbaldi@polito.it

Y. Ofek
Synchrodyne Networks, Inc.
2600 Netherland Ave.
New York, NY 10463
ofek@synchrodyne.com

Abstract

In general topology networks, routing from one node to another over a tree embedded in the network is intuitively a good strategy, since it typically results in a route length of $O(\log n)$ links, being n the number of nodes in the network. Routing from one node to another over a ring embedded in the network would result in route length of $O(n)$ links. However, in group (many-to-many) multicast, the overall number of links traversed by each packet, i.e., the networks elements on which resources must be possibly reserved, is typically $O(N)$ for both tree and ring embedding, where N is the size of the group. This paper focuses on the tree versus ring embedding for real-time group multicast in which all packets should reach all the nodes in the group with a bounded end-to-end delay. In this work, real-time properties are guaranteed by the deployment of time-driven priority in network nodes.

In order to have a better understanding of the non-trivial problem of ring versus tree embedding, we consider the following group multicast scenarios: (i) static - fixed subset of active nodes, (ii) dynamic - fixed number of active nodes (i.e., the identity of active nodes is changing over time, but its size remains constant), and (iii) adaptive - the number and identity of active nodes change over time.

Tree and ring embedding are compared using the following metrics: (i) end-to-end delay bound, (ii) overall bandwidth allocated to the multicast group, and (iii) signaling overhead for sharing the resources allocated to the group. The results are interesting and counter-intuitive, since, as shown, embedding a tree is not always the best strategy. In particular, dynamic and adaptive multicast on a tree require a protocol for updating state information during operation of the group. Such a protocol is not required on the ring where the circular topology, and implicit token passing mechanisms are sufficient. Moreover, the bandwidth allocation on the ring for the three multicast scenarios is $O(N)$; while on a general tree it is $O(N)$ for the static multicast scenario and $O(N^2)$ for the dynamic and adaptive multicast scenarios.

1 Introduction

There are two commonly used routing methods for broadcast/multicast: (i) in general networks, packets are forwarded to all the destinations over a tree embedded in the network, and (ii) in local area networks, packets are forwarded over a ring or a bus. This work presents a systematic comparison of the two basic approaches for multicast routing in the context of real-time multicast in general networks. We do not deal with how to find the best ring and tree embedding, which has been largely investigated. Instead, given the embedding, our work focuses on determining which embedded structure has the most desirable properties. The objective of this paper is to increase the understanding of the real-time multicast problem, rather than to provide a specific design for a specific network such as the Internet.

The following are some definitions which are used throughout this study.

Network. A connected undirected graph, where all links are bi-directional. Each bi-directional link is considered as the union of two simplex unidirectional links.

Multicast group. A subset of N nodes of the network which are collectively addressed for the reception of packets sent by members of the multicast group.

Group multicast. An operation of many-to-many communications in a multicast group.

Active node. A member of the multicast group which is actually sending packets addressed to the group.

Tree embedding. The N member nodes of a multicast group, connected via an undirected tree that is embedded in the network.

Assumption 1 Ring embedding is an Euler Tour of the same embedded tree. This is also referred to as ring-on-tree.

The objective of the above assumption is to simplify the discussion. In fact, the focus of this comparative study is not on building and maintaining a multicast structure – ring or tree – as the group size or network condition change, but rather on resource allocation and deployment by active group members. Since both tree and ring-on-tree have the same underlying topology, as shown in Figure 1, finding and maintaining them has comparable complexity. Consequently, a ring-on-tree, with $2N-1$ nodes, is considered here, even though a minimum size ring embedding could be made by having a traveling salesman tour of the N nodes. Without loss of generality, in order to simplify the discus-

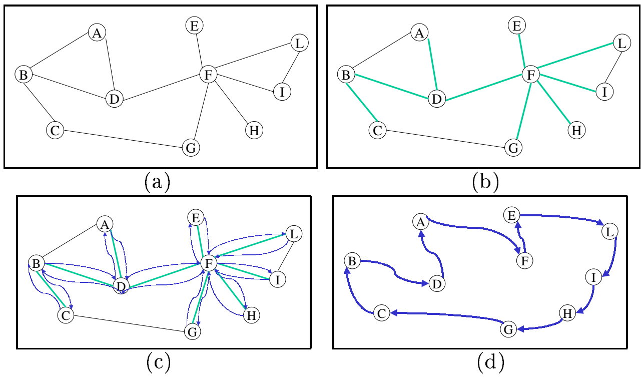


Figure 1: Embedding of a Tree (b) over a General Topology Network (a); Embedding of a ring as Ring-on-Tree or Euler Tour on a Tree (c) and as Traveling Salesman Tour (d).

sion, the network nodes which are not part of the multicast group, in both tree embedding and ring embedding on a tree, are ignored.

This work is further characterized by resources being reserved to a multicast group as a whole, as opposed to performing a specific reservation for each potential active node. Moreover, it is assumed that bandwidth is not a free commodity, and therefore, over allocation of bandwidth is not a desirable solution. This is especially true for real-time applications, such as voice and video, which have better quality with more bandwidth and possibly involve costly long haul links. Accordingly, time-driven priority is taken into consideration as the underlying mechanism to guarantee real-time delivery of packets under full network load.

Section 2 describes the basic operation principles of time-driven priority which is used to control the flow of real-time packets in the network and thus determines the framework for the comparison. Section 3 analyzes real-time multicast scenarios for the embedded ring-on-tree case, while Section 4 analyzes the more complex case of the embedded tree. Finally, Section 5 summarizes and discusses the outcome of this comparative study.

1.1 Related Work on Asynchronous Packet Multicasting

Within the Internet Engineering Task Force (IETF) consensus has been found on multicast routing solutions that involve embedding trees over the Internet. DVMRP (Distance Vector Multicast Routing Protocol) [28], PIM-DM (Protocol Independent Multicast-Dense Mode) [1], and MOSPF (Multicast Open Shortest Path First) [14] route multicast packets on trees from each source to all the members of a multicast group. The three protocols differ in the way source specific trees are built. Both DVMRP and PIM-DM perform selective flooding of packets and deploy a pruning procedure initiated by leaf routers that do not serve any member of the multicast group. MOSPF extends the OSPF routing protocol to carry group membership information so that routers can perform source/destination multicast routing, i.e., compute a tree from the source of a multicast packet to all the members of the multicast group to which the packet is addressed. Being based on an interior gateway protocol (IGP), MOSPF can be used only within an Autonomous System. Consequently, MOSPF deployment over the Internet requires another multicast routing protocol, such as DVMRP and PIM, that can be used across multiple Autonomous Systems.

According to the CBT (Core Based Tree) [3, 4] approach packets are routed over shared trees. Routing based on shared trees requires less computation in routers since they need to calculate only one tree per multicast group, independently of the number of traffic sources. In CBT packets from any source are routed to a node called *core* and then from the core to each of the destinations. Thus, the subtree rooted at the core is shared by multicast packets generated by all the sources of the same multicast group. Also according to PIM-SM (Protocol Independent Multicast-Sparse Mode) [7] routers forward packets to a node called *rendez-vous point* and then from there to each member of the multicast group. However, routers can choose to forward packets from selected sources over a source specific tree in order to optimize routing of the traffic generated by such sources.

The IETF is working on scalable solutions for inter-domain multicast routing. A possible approach consists in connecting intra-domain multicast routing trees rather than building multicast trees across multiple Autonomous Systems.

Notwithstanding IETF's focus on trees, using virtual rings for many-to-many communications is simple and "natural", since a message sent from one member of the group travels around the ring until it comes back to the sender. Virtual rings can be embedded over general topology networks, one virtual ring for each multicast group. In a previous work it has been shown that a virtual ring can be instrumental in designing a *reliable multicast protocol* from *bursty sources* [19]. A novel method for broadcasting and multicasting over packet network was proposed in [20] .

Real-time group multicast with allocation of resources to the group was addressed in depth by the Tenet Group in the framework of the Tenet Project [8]. Multicast communications are implemented through a set of independent trees, each one rooted at a sender [9]. Quality of Service (QoS) is guaranteed by giving priority to real-time traffic over best-effort traffic, by performing rate control and scheduling in network nodes, and by restricting access to network resources through call admission control. When resources are allocated on a link which belongs to multiple trees used by a multicast group, a *shared allocation* is performed [10]. Network nodes, based on the traffic characterization provided by each source, allocate enough shared resource to guarantee the required QoS to each flow. When resource sharing is used, rate control and scheduling can be performed per group (instead of per single flow), which is an important scalable property.

A similar approach to real-time group multicast is proposed by IETF's Integrated Services Working Group in the framework of the Integrated Services Internet [25]. As mentioned above, multicast communications are implemented through both source specific and shared trees. The Integrated Services model encompasses two QoS control models: the *Controlled-Load* [29] service and the *Guaranteed QoS* [24]. The first service model is not relevant to this work, while the second aims at approximating the fluid flow service model in each node by exploiting call admission control, policing at the edge of the network, scheduling of reserved resources, and reshaping within nodes. Applications require the needed QoS using a setup mechanism, e.g., the Resource reSerVation Protocol (RSVP) [30]. Since reservations are soft state, applications must renew their requests periodically and may change their QoS requirements. Reservation on links used by multiple senders of the same multicast group can be made either individually for each source, or globally. In the latter case, no mechanism is foreseen to coordinate senders and avoid that the instantaneous overall demand exceeds the allocated shared amount.

The real-time multicast approaches presented above base QoS assurance on packet scheduling mechanisms, such as Packet Generalized Processor Sharing (PGPS) [21, 22] and (WFQ) [6] that bound the packet delay by approximating the fluid flow service model. Such schemes guarantee a bound on the queuing delay which is inversely proportional to the connection (or session) rate, proportional to the number of nodes traversed by that connection, and proportional to the packet size.

In order to avoid the need for hard QoS guarantees, mechanisms are being devised [23] to allow applications to infer the service actually provided by the network in order to react to its variations [26]. This approach takes into consideration application adaptiveness, but can result in uncontrolled quality.

Even though previous work on real-time group multicast does exist, it did not focus on comparing ring-based and tree-based routing. Analogously, previous work on the comparison of rings and trees was not related to real-time group communications. Moreover, our work assumes time-driven priority as the underlying mechanism to guarantee real-time delivery of packets under full network load. In summary, the comparison of ring and tree embedding for real-time group multicast with time-driven priority is novel.

2 Operation Principles and Comparison Measures

In this sort of study, clear comparison measures are needed to obtain meaningful and insightful results. In order to provide a framework for the comparison, some specific operation principles should be defined. The context of this study is *time-driven priority*, which is appealing since it provides deterministic guarantees on *end-to-end delay bounds* and *bandwidth* with no packet loss due to congestion. Thus, it is possible to determine the quality of service as it will be perceived by the applications.

2.1 Time-driven Priority Operation

Time-driven priority (TDP) [13] can support flows at both constant bit rate (CBR) and variable bit rate (VBR) with a certain degree of periodicity in their traffic[1]. However, in this study it will be assumed that active nodes generate CBR traffic. Note that TDP supports also "best effort" traffic that is transmitted with lower priority and without resource allocation.

A *global common time reference* is required to implement TDP. This time reference can be provided by an external global source like the *global positioning system* (GPS) [16], or can be generated and distributed inside the network with some in-band signaling, as proposed in [18, 17].

Some general concepts related to TDP are reported in the following:

Time Frames. Using the global time reference nodes divide time into *time frames* (TFs) of duration T_f. In each TF one or more packets or ATM cells can be transmitted (for example, if $T_f = 125\mu$sec and the transmission rate is 1Gb/sec, about 290 ATM cells can be transmitted in every TF).

Time Cycle. k TFs are grouped in a *time cycle* which has a duration $k \cdot T_f$. The TFs in a cycle are numbered from 0 to $k-1$.
All arithmetic expressions involving TF numbers are meant to be modulo k, e.g., if i is a frame number, then $(i+1)$ means $(i+1) \bmod k$.

Active node transmission rate. It is determined by the number of data units (e.g., bits, bytes, cells) that can be sent in every time cycle, divided by the time cycle duration $k \cdot T_f$.

TDP pacing conditions. The traffic over a route is said to be *TDP paced* if the following two conditions are true:
Condition 1: All packets that should be sent in TF i by a node are in its output port before the beginning of TF i.
Condition 2: The delay between an output port of one node and the output port of the next node is a constant number of TFs.

Note that this constant delay includes the propagation delay, routing delay, and switching time. Without loss of generality in this work we assume the delay between the output ports of neighboring nodes to be 1 TF.

Definition 1 *Time-driven priority immediate forwarding.* Packets due at an output port by TF i are sent out in TF $i+1$.

Delivery of a packet from the source to the destination traveling across E links takes $2E-1$ TFs, since 1 TF is taken to travel on each of the E links (from an output port to the next output port) and 1 TF is spent in the output port of each of the $E-1$ nodes (since the packet is sent in the TF following the one in which it arrived at the output port). In order to enable TDP immediate for-

[1] Compressed video is an example of such a traffic; see [12] for further details on the transmission of MPEG compressed video using TDP.

warding the number of packets arriving at the output port of a node during each TF must be controlled, i.e., sessions must reserve resources (namely, TF fractions). Resource reservation with TDP requires to find a schedule because the TFs in which transmission time is reserved on each link are uniquely determined by the reservation on the upstream link. The impossibility of reserving resources even though they are available, but not during suitable TFs, is called *blocking*. In order to simplify the exposition, we assume that a TF can be used by only one active node.

Definition 2 *Time frame allocation.* A TF on a link is reserved for transmission by a node or a group of nodes.

Assumption 2 Resource allocation. (i) Only one resource allocation is done for each group multicast operation, which means that *no separate allocation is made for individual active nodes.* (ii) The resource allocation does not change during the multicast session.

Definition 3 *Time frame assignment.* At a certain point in time, a TF reserved to a multicast group is said to be *assigned* to the active node that is actually using it for transmission.

2.2 Performance Measures

The multicast performance over rings and trees is compared according to the following performance measures.

Definition 4 *Network delay bound.* The maximum delay in TFs experienced by a packet sent from any active node to any other member of the multicast group.

Definition 5 *Resource allocated to a multicast group.* The number, B, of allocated TFs per time cycle.

As shown in Section 3, the amount of resources that must be allocated for multicast transmission over a ring is independent of the multicast method (static, dynamic or adaptive - see Section 2.3). Thus, this amount of resources is taken as a reference in the comparison of resource allocation in the various cases. This leads to the following definition.

Definition 6 *Reservation ratio.* The ratio between the amount of TFs that must be allocated for a given embedding (ring or tree) and multicast method, and the amount of TFs needed for the transmission over a ring.

Definition 7 *Updating state information or coordination complexity.* The amount of state information that is exchanged during the group multicast operation to coordinate the usage of TFs among the active nodes.

Another possible comparison means is the schedulability or, dually, the blocking probability achievable with the given configuration. Even though a quantitative analysis of blocking probability is outside the scope of this work, some qualitative observations are made throughout the paper.

2.3 Methods of Real-time Group Multicast

We consider the size of the multicast group and the number and identity of its active sources as parameters in this comparison, since they affect the multicast performance.

Definition 8 *Static Multicast.* The *number*, N_a, and *identity* of active nodes is fixed during the multicast operation.

The broadcasting of an event is a typical case of static multicast in which a number of destinations receive audio and video feeds from one or more broadcasting sources (active nodes). The sources remain the same throughout the event and are constantly transmitting (i.e, active).

Definition 9 *Dynamic Multicast.* The *identity* of the active nodes changes over time, while their *number*, N_a, is fixed.

Dynamic multicast takes place, for example, in a conference call that allows a fixed number of speakers and a large number of listeners. The number of TFs each active node can use is determined during the setup of the multicast communication. Active nodes *dynamically* and fairly share the B TFs allocated to the multicast group: each active node uses $b = B/N_a$ TFs. The identity within a time cycle of the TFs assigned to an active node are not necessarily the same over the duration of the multicast session.

Definition 10 *Adaptive Multicast.*
(i) The *number* and *identity* of active nodes change over time.
(ii) The adaptiveness range is the minimum and maximum number of TFs per time cycle (i.e., capacity) an active node can use.

A typical scenario for adaptive multicast is a videoconference with a fixed number of participants, wherein a variable number of them is actively involved in a discussion. As the number of speaker increases, each source must decrease the rate of its transmission, and vice versa. In this study the adaptiveness range is $[1, B]$, i.e., the number of active nodes, varies in the interval $1 \leq N_a \leq B$.

3 Performance of Ring-on-Tree Embedding

We start the discussion with the analysis of the ring embedding, or virtual ring, since it is simpler and helps in understanding the issues tackled in this work. Packets travel over the virtual ring-on-tree in one direction; since the number of tree nodes is N, the *virtual ring size* (i.e., the number of links constituting the ring) is $2(N-1)$.

Resource allocation is the same for all three multicast methods. Due to the uni-directionality of transmissions over the ring, whenever an active node sends a packet it is forwarded on all links of the ring. In order to allow N_a active nodes to transmit during b TFs, $B = N_a \cdot b$ TFs per time cycle must be allocated on each link, independent

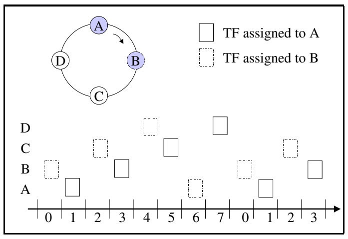
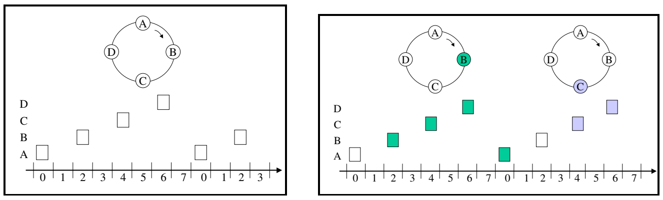


Figure 2: Sample Bandwidth Allocation for Static Multicast.



(a) (b)

Figure 3: Sample Bandwidth Allocation for a Multicast Group (a) and Forwarding of a Packet with the given Allocation (b).

of the instantaneous identity of the active nodes, and of whether they change over time or not. Thus, the allocation of B TFs per link enables both static and dynamic multicast. The same allocation is used also for the adaptive case; resources are allocated to the multicast group as a whole and as the number of active nodes changes, each active node uses from 1 ($N_a = B$) to B ($N_a = 1$) TFs.

Summarizing, the overall resource allocation on the network for all the multicast methods is:

$$R_R = 2B(N-1).$$

R_R is taken as the reference (denominator) for expressing the reservation ratio ρ; $\rho_R = 1$ for all the multicast methods over a ring.

In the following we discuss the delay bound, the distribution of state information and other specific issues for each multicast method.

3.1 Static Multicast

When performing static multicast, the R_R TFs are allocated on the ring as N_a independent allocations of b TFs since the active nodes are always the same. Given one active node, b TFs are chosen on each link starting from the source all around the ring. Figure 2 shows a sample allocation of 1 TFs for each of two active nodes A and B.

In the graphic notation used throughout this paper, a square represents the allocation of the TF identified by the column, on the output link of the node labeling the row. A continuous line square indicates that the TF is assigned to active node A, while a dashed line square indicates that the TF is assigned to active node B.

3.1.1 Network delay bound

Assuming that a schedule enabling TDP immediate forwarding from the N_a active nodes is found, the definition of TDP immediate forwarding given in Section 2.1 provides the delay bound as $2E-1$, where E is the maximum number of links crossed by a packet traveling between any two members of the multicast group. Over a ring, such a number is the ring size—which is $2(N-1)$ for a ring-on-tree—minus 1. Thus, the delay bound is given by

$$D_R^S = [2(2N-2)-1]T_f = (4N-5)T_f$$

If the ring was embedded in the network as a traveling salesman tour of all the multicast group nodes, then the ring would be composed of N links and the delay bound would be only $(2N-3)\cdot T_f$.

3.1.2 Updating State Information or Coordination Complexity

The only state information needed in static multicast is the identity of the TFs assigned to each active node. There is no need for updating such state information during the multicast operation since it remains unchanged.

3.2 Dynamic Multicast

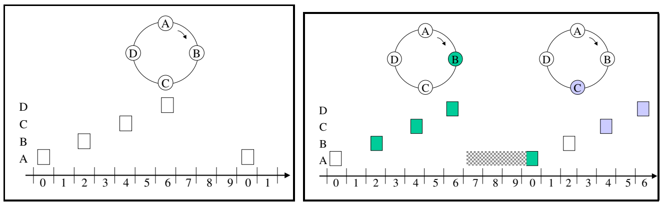
The TF allocation cannot be done for each active node independently since the identity of the active nodes changes over time. On the contrary, B TFs are allocated on each link and any active node shall be able to use any of them to transmit its packets. Figure 3(a) shows a sample reservation of 1 TF on each link to a multicast group which encompasses four members. Figure 3(b) shows an example of dynamic multicast, in which a single (changing over time) active node uses the TF allocated on its outgoing link to send a packet. When B is the active node, it waits until TF 2 and then sends the packet; it reaches the output port of A by TF 7, as shown in Figure 3(b), in order to be available for being forwarded by A during TF 0. The packet then travels on the ring and gets back to node B by TF 1 of the following time cycle. Then, B becomes passive and C becomes active and transmits a packet in TF 4.

Since in general each node uses any of the TFs allocated on its outgoing links, TDP immediate forwarding should be performed for packets sent during any of the allocated TFs on any pair of subsequent links in the virtual ring. As shown in Section 3.2.1, depending on the relationship between the number of links and the dimension of the time cycle, immediate forwarding may not be possible on the whole ring and one of the nodes, the *buffering node*, performs immediate forwarding. This increases the delay bound with respect to the static case as described in Section 3.2.2.

Since when an active node needs to send a packet, it must use one of the B TFs allocated to the multicast group on its outgoing link, a coordination mechanism is needed to ensure fair sharing of the allocated TF among the active nodes. Transmission on a virtual ring enables exploitation of well experimented, simple, and effective mechanisms, as discussed in Section 3.2.3.

3.2.1 Buffering Node

With reference to Figure 3(a), TF 0 is reserved on node A's outgoing link, and in order to enable immediate forwarding TFs must be reserved accordingly on all the other links of the ring. Immediate forwarding is possible along the whole ring because the network delay on all the links (in the following called *ring length*) equals the length of the time cycle. As a consequence, the packets transmitted during the TF allocated on the link between D and A get to node A by the TF before the one allocated on A's outgoing link (i.e., TF 0). If the ring latency (8 TFs) was not an integer multiple of the time cycle (8 TFs), the TF allocated on A's incoming link would not fit to the TF allocated on the outgoing link. As a consequence at least one of the nodes of the ring (A in our example) could not perform immediate forwarding.



(a) (b)

Figure 4: Sample Bandwidth Allocation for a Multicast Group (a) and Forwarding of a Packet Originated from B on a Virtual Ring with Buffering Node A (b).

Figure 4(a) shows the allocation of 1 TF on the same ring with a longer time cycle of $k = 10$ TFs and Figure 4(b) shows the forwarding of packets. Node A receives B's packet by TF 7, but it has no TF reserved until TF 0 of the following time cycle; thus, node A buffers B's packet until the next TF reserved for its multicast group. For this reason such a node is called a buffering node.

Definition 11 *Buffering node.* It is a node on a ring which does not perform immediate forwarding, in order to match the ring length and the time cycle duration.

Each ring may have a buffering node to adapt the length of the ring to the size of the time cycle [2]. Since the buffering node has particular buffer requirements, we assume that

Assumption 3 Only one buffering node is used on a virtual ring.

It must be noted that due to the presence of the buffering node, when more than one TF in each time cycle is reserved for a multicast group, a station does not necessarily receive back a packet by the same TF, (of the following time cycle) in which it transmitted it.

3.2.2 Network Delay Bound

Due to the presence of the buffering node, the delay bound for the multicast communication is larger than in the static case. Namely, it is given by the length of the ring, plus the maximum time spent in the buffering node. The maximum buffering time depends on the distribution of the allocated TFs inside the time cycle and on the relationship between the ring length and the time cycle size. The delay bound is given by:

$$D_R^D = (4N - 5 + D_b)T_f,$$

where $D_b \in [0, k]$ is the maximum buffering time given in terms of number of TFs and k is the size of the time cycle.

3.2.3 Updating State Information or Coordination Complexity

Dynamic multicast requires only the *number* N_a of active nodes to be known to each node in the group, so that it can devise its fair share of TFs it can use. This number does not change during the multicast session.

The dynamic assignment of the TFs to the nodes can be managed through *implicit token passing*. Thus, TF sharing among the active nodes does not require the distribution of state information during the multicast session. Simple mechanisms for TF sharing in a dynamic multicast operation over a virtual ring were proposed in [2]. Each node knows the identity of the TFs reserved to the multicast group and the number of TFs, b, it can use for transmission. When it becomes active, it transmits during the first b free TFs it identifies and continues transmitting during these TFs as far as it remains active. A TF i is said to be a *free TF* if *(i)* it is reserved to the multicast group and *(ii)* no packets addressed to the multicast group get to the output buffer by TF $i - 1$. It is worth noting that there is no need to signal state changes: a node becoming passive simply stops transmitting, while a passive node waits to become active until it "sees" free TFs on the ring.

3.3 Adaptive Multicast

Packet forwarding is performed in a way similar to dynamic multicast and a buffering node is used when needed to match the ring length with the time cycle size. Thus, the delay bound for multicast transmission depends on the ring latency and on the buffering node delay (see Section 3.2.2), i.e.

$$D_R^A = D_R^D = (4N - 5 + D_b)T_f.$$

Updating State Information or Coordination Complexity

As it is in dynamic multicast, adaptive multicast requires each active node to know the number B/N_a and the identity of the TFs it can use for transmission. The number N_a of active nodes changes during the operation of the multicast group and a protocol is needed to signal to all the active nodes whenever a node becomes active or passive[2]. Then, each source knows that it is allowed to transmit

[2]Actually, through the implementation of an elaborate round-robin TF allocation algorithm, the protocol for signaling node state changes can be avoided. Nevertheless, as discussed in [2], this can result in decreased efficiency in the usage of network resources.

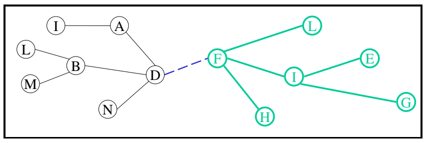


Figure 5: Subtree Induced by Link DF.

during B/N_a TFs and can identify the TFs assigned to itself by the implicit token passing method described in Section 3.2.3.

In adaptive multicast, the share of allocated TFs B/N_a cannot be always an integer number; [2] proposes a simple algorithm to fairly and effectively assign the $B \bmod N_a$ spare TFs. Each active node takes static possession of $\lfloor B/N_a \rfloor$ TFs; the remaining TFs are periodically allocated to all the active nodes through an implicit token passing protocol that exploits circular nature of the ring.

4 Performance of Tree Embedding

4.1 Transmission on a Tree using Time-driven Priority

The following definitions are used in the evaluation of tree embedding performance:

Definition 12 *Induced subtree.* Given a unidirectional link l, the subtree consisting of all nodes reachable through l as the first hop is the subtree induced by l and it is indicated as T_l.

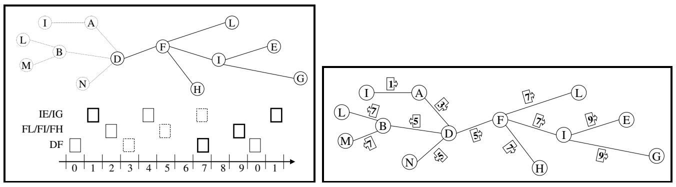
Figure 5 shows the subtree (thick line) induced by link DF (dashed line).

Deployment of TDP immediate forwarding requires a specific relationship between the TFs allocated on the incoming link of a node and the choice of the TFs allocated on all its outgoing links. If the set of b TFs $T = \{t_1, \ldots t_b\}$ is allocated on the incoming link of a node, the TFs $T' = \{t_1+2, \ldots t_b+2\}$ should be reserved on each outgoing link[3]. Figure 6(a) shows a sample allocation of TFs on some of the links of the tree depicted in that picture; the same line pattern is used to draw the TFs correlated by the deployment of TDP immediate forwarding. A set of 3 TFs {0,3,7} is allocated on the link between D and F; TFs {2,5,9} are consequently allocated on F's outgoing links, namely, those directed to H, I, and L.

Definition 13 *Tree schedule.* Given a TF allocated for transmission of node n over its outgoing link l, its tree schedule is the collection of the TFs allocated on each link of the subtree induced by l, chosen in a way that a packet sent by node n in the given TF is delivered with TDP immediate forwarding to all the nodes of the induced subtree.

Figure 6(b) shows the tree schedule of TF 1 on the outgoing link of node I; the arrows wrapping numbers indicate the direction of the link in which the TF allocation has been

[3] In Section 2.1 it is assumed that the delay from the output buffer of the upstream node to the output buffers of the given node is 1 TF, thus, the delay between two successive forwarding of a packet is 2 TFs.



(a) (b)

Figure 6: Scheduling of TFs on a Tree (a) and Sample Tree Schedule Starting from Node I (b).

performed. Active nodes transmit during TFs for which a tree schedule has been found, and the real-time properties of the communication are guaranteed by TDP immediate forwarding.

Network Delay Bound

For all three multicast methods the maximum delay over the tree when TDP immediate forwarding is performed is given by:

$$D_T = (2 \cdot H - 1)T_f,$$

where H is the diameter of the tree (in terms of number of links) which is on the order of $\log N$ (N is the size of the multicast group).

4.2 Static Multicast

4.2.1 Resource Allocation

There are N_a active nodes, and each requires b TFs to be allocated on its outgoing link(s) and a tree schedule to be found for each of the b TFs. Thus, $N_a \cdot b$ TFs are reserved to the multicast group on each link and the total amount of allocated TFs is given by:

$$R_T^S = N_a \cdot b(N-1) = B \cdot (N-1). \quad (1)$$

4.2.2 Reservation Ratio

The reservation ratio of static multicast over a tree is given by:

$$\rho_T^S = \frac{R_T^S}{R_R} = \frac{B(N-1)}{2B(N-1)} = \frac{1}{2},$$

i.e., static multicast over a tree requires half the resources required over a ring embedded on the same tree. This is due to the fact that with ring-on-tree embedding each packet is forwarded in both directions of each tree link.

4.2.3 Updating State Information or Coordination Complexity

The only state information needed by an active (source) node for static multicast operation is the identity of the TFs it has been assigned on its outgoing link(s). This state information is set when the node begins operation and does not change over time; thus, no specific protocol or mechanism is needed to update state information.

4.3 Dynamic Multicast

The total amount of communications resources needed depends on the tree topology and the locations of the currently active nodes. Therefore, it is not possible to devise a closed form expression for the minimum resource allocation needed for dynamic multicast operation over a general tree - the state space is much too large, and the problem is made harder by conflicting scheduling requirements. We try to give an insight of the constraints that drive such a resource reservation and provide lower and upper bounds for the minimum resource allocation.

A lower bound for the allocation is first devised (Section 4.3.1) as the minimum allocation that allows any possible set of N_a active nodes to simultaneously transmit using reserved resources (i.e., during TFs allocated to the multicast group).

Then, the requirement for TDP immediate forwarding is brought in, thus raising the issue of scheduling over the tree (Section 4.3.2). It is proven that dynamic multicast with TDP immediate forwarding is not possible on every tree with the lower bound allocation.

The dynamic assignment of allocated TFs to active nodes is taken into consideration (Section 4.3.3); the lower bound allocation proves not to allow dynamic assignment. Moreover, we show that it is not always possible to find a schedule on every tree for any TF allocation which allows dynamic allocation. An upper bound on resource reservation is given and it is proven to both allow dynamic assignment and be schedulable on any tree. Then, the minimum allocation enabling dynamic multicast over a core based tree is given (Section 4.3.4). The section concludes by providing the reservation ratio (Section 4.3.5) and the control complexity (Section 4.3.6) for dynamic multicast over a tree.

4.3.1 Resource Allocation

A minimum number of TFs is allocated on each link to enable the nodes to transmit using reserved resources, thus obtaining a guaranteed service. b TFs are reserved on the outgoing link of each leaf so that when the node becomes active it can use them to transmit its packets. The TFs reserved on the links departing from passive nodes are not used, and the corresponding capacity can be exploited for carrying best-effort traffic.

A generic node of the tree receives multicast packets from its incoming links and forwards them on all the outgoing links. Besides forwarding packets received on its incoming links, a node must transmit its own ones with a guaranteed service when it is active. Thus, depending on the number of upstream nodes, more TFs are possibly allocated on its outgoing links.

Theorem 1 Given a unidirectional link l, the minimum number of TFs that must be reserved to the multicast group M on l to provide guaranteed transmission of packets sent by N_a active nodes during b TFs is given by b times the minimum between (1) the total number of multicast members not contained in the subtree induced by l and (2) N_a, i.e.,

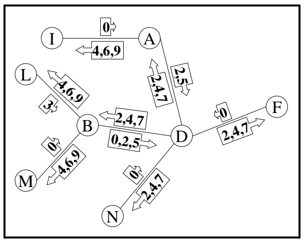


Figure 7: Lower Bound on TFs Allocation on a Tree.

$$R^D_{T,l} = \min\{|M \backslash T_l|, N_a\} b$$

Proof Long queuing delays and drop of packets to be forwarded on link l can be avoided if the number of packets that the transmitting node n can send on l during each time cycle is equal to or greater than the number of packets addressed to the multicast group it receives from all the incoming links during the time cycle. The delay experienced by packets in each node and the buffer space required to store the packets from when they are received, until they can be forwarded, depends on the scheduling.

The number of multicast packets that n is expected to forward on l depends on the number of active nodes not contained in the subtree induced by l. Since the set of active nodes changes over time and any member of the group can be active, the maximum number of active nodes not contained in the subtree induced by l is $|M \backslash T_l|$. In any case, the total number of active nodes sending packets that traverse link l cannot be larger than N_a, the number of active nodes in the multicast group.

Since each node transmits during at most b TFs in each time cycle, the maximum number of packets[4] that n can forward to link l during a time cycle is the minimum between $b \cdot N_a$ and $b|M \backslash T_l|$. □

The total TF allocation is given by:

$$R^D_{T,min} = \sum_{l \in L} \min\{|M \backslash T_l| \cdot b, B\}\text{TFs}, \tag{2}$$

where L is the set of unidirectional links of the tree. $R^D_{T,min}$ is the lower bound on the allocation to perform dynamic multicast over a tree with controlled delay and loss. Figure 7 shows a simple allocation of TFs assuming $b = 3$, $N_a = 3$, $N = 8$, and a time cycle of 10 TFs.

It is worth noting that if $N >> N_a$ (as it is likely in large scale videoconferences), $\hat{R}^D_{T,l} \simeq N_a \cdot b = B$ TFs, and the total TF allocation on the tree is

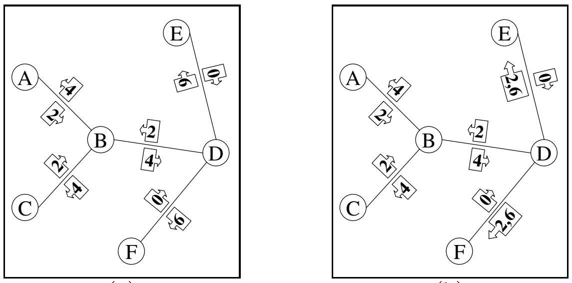
$$\hat{R}^D_{T,min} \simeq N \cdot N_a \cdot b = N \cdot B \text{ TFs}. \tag{3}$$

4.3.2 Scheduling

In order to enable TDP immediate forwarding for the multicast delivery, the order and identity of the allocated TFs, i.e., the schedule, must be properly chosen along the time axis.

Definition 14 *Tree schedulable.* A TF allocation to a multicast group is *tree schedulable* if it is large enough to

[4] It is possible that during the time cycle in which the set of active nodes changes both the node becoming passive and the new active node transmit during the same b TFs. This results in more than $b \cdot N_a$ packets sent to the multicast group. This situation is not considered in this proof since it is exceptional and can be avoided by properly delaying the beginning of the transmission of a newly active node.



(a) (b)

Figure 8: Example of Tree on which the Lower Bound on the TF Allocation is not *Tree Schedulable* (a); Tree Schedulable TF Allocation on the same Topology (b).

allow, for each TF in the allocation, its tree schedule to be also included in the allocation.

Note that a TF can be part of more than one tree schedule. Moreover, tree schedulability is related to the size of the allocation, i.e., the number of allocated TFs, and not to their identity.

Tree schedulability is *necessary* to guarantee that when an allocated TF is used for transmission of a packet, the network will deliver the packet to all the members of the multicast group using TDP immediate forwarding. However, tree schedulability is *not enough* to enable multicast TDP immediate forwarding: the TFs must be chosen properly on the links. Note that the allocation given by Equation (1) is tree schedulable since it enables static multicast.

Theorem 2 The lower bound on the TF allocation $R^D_{T,min}$ (as given by Theorem 1) is not tree schedulable on every tree.

Proof The proof is given by reduction ad absurdum, i.e., by showing a negative example. Considering a multicast group with a single active node ($N_a = 1$) and a single TF per time cycle needed by the active node for transmission (i.e., $b = 1$), only one TF shall be allocated to the multicast group on each link. Figure 8(a) shows a simple tree topology on which the minimum TF allocation of 1 TF per link is not tree schedulable. A packet sent during the TF allocated on link ED (i.e., TF 0) cannot be forwarded on link DF using TDP immediate forwarding since TF number 2 is not reserved to the multicast group on link DF. In fact, TF 6 is allocated on link DF in order to allow TDP immediate forwarding of packets sent during TF 4 on link BD and TF 0 is reserved on link ED to enable packets to be TDP immediate forwarded during TF 2 on link DB. □

In order for the allocation to be tree schedulable, two more TFs should be reserved to the multicast group: TF 2 on link DE and TF 2 on link DF, as shown in Figure 8(b).

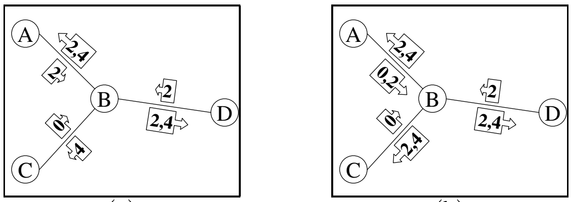
4.3.3 Dynamic Assignment of TFs

In order to limit the amount of resources reserved to the multicast group, the TFs allocated are dynamically assigned to the active nodes. Dynamic assignment is effective if it is possible to assign the same TFs to different nodes as the set of active nodes changes.

Definition 15 *Dynamically assignable.* A tree schedulable allocation is *dynamically assignable* if (i) for any possible set of N_a active nodes and (ii) independently of the tree topology, it is possible to assign to each active node (1) b TFs on its outgoing links and (2) the tree schedule of these b TFs, avoiding the same TF to be assigned to more than one active node at the same time.

It should be noted that dynamic assignability, as well as tree schedulability, is concerned only with the size of the allocation, i.e., with the number of allocated TFs, and not with the identity of TFs within each time cycle.

Dynamic assignability is necessary to enable dynamic multicast and can require a larger TF allocation than tree schedulability. Consider as an example the tree schedulable allocation shown in Figure 9(a), which is not dynamically assignable to two active nodes ($N_a = 2$), for each one is transmitted during one TF ($b = 1$). In fact, if nodes A and D are active simultaneously, they should be both assigned TF 4 on link BC since it is the tree schedule of the only TF allocated on their outgoing links. Figure 9(b) shows a dynamically assignable allocation for the same multicast group which requires reserving two more TFs.



(a) (b)

Figure 9: Example of Tree Schedulable, but not Dynamically Assignable Allocation (a); Sample Dynamically Assignable Allocation (b).

The following corollary provides an upper bound on the minimum allocation enabling dynamic multicast over a tree.

Corollary 1 The allocation

$$R^D_T = N \cdot b(N-1) \text{ TFs} \tag{4}$$

is tree schedulable and dynamically assignable.

Proof The corollary can be trivially proven by reducing the dynamic multicast of N_a active nodes among N group members, to static multicast of N active nodes (since each node can be potentially active). R^D_T is thus obtained from Equation (1) with N active nodes. □

Whenever the allocation R^D_T is used, no advantage is taken of the nature of the dynamic multicast, namely that no more than N_a nodes are active concurrently. The allocation given by Corollary 1 is a loose bound on a tree schedulable allocation. A tighter bound is devised in Section 4.3.4 for a Core Based Tree.

4.3.4 Core Based Tree

A Core Base Tree (CBT) [5, 3, 4] is a shared, bi-directional multicast tree on which each packet travels from the source to a node called the *core*, and from the core to the destinations. This makes tree schedulability easier to study and less demanding from the resource allocation viewpoint. In fact, the minimum TF allocation avoiding congestion in

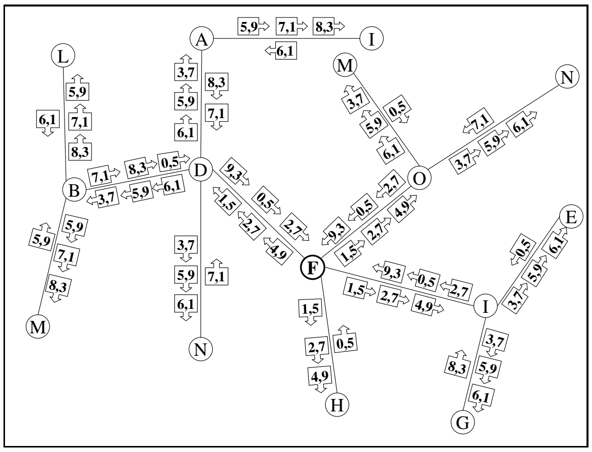


Figure 10: Minimum TF Allocation on a Core Base Tree.

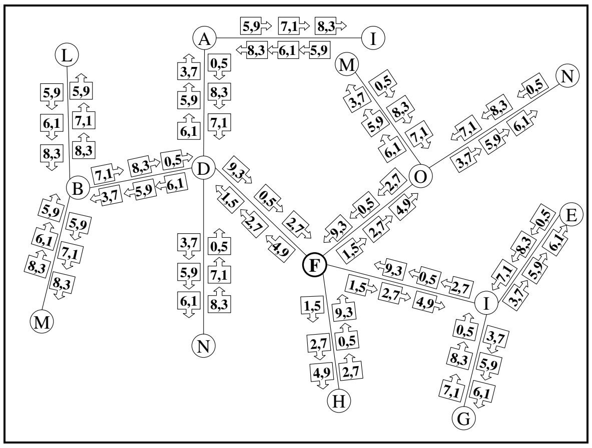


Figure 11: Tree Schedulable and Dynamically Assignable Allocation on a Core Base Tree.

dynamic multicast on a CBT is tree schedulable, as proven by Theorem 3 below. Intuitively, the CBT is simpler to schedule since there is a single time reference point, which is the core node, for packet forwarding from any source. In other words, the tree schedules for TFs allocated on the outgoing link of any source can be easily found starting from the core node.

The minimum TF allocation on each link l enabling guaranteed transmission from N_a active nodes is the one given by Theorem 1 for a generic tree. However, when applied to a CBT, the number of active nodes not contained in the subtree induced by each link directed from the core to the leaves is 0 and Equation (2) can be re-written as

$$R^D_{CBT,min} = \sum_{l \in L,\ \mathrm{core} \notin T_l} \min\{|M \backslash T_l| b, B\} + B(N-1) \text{ TFs},$$

where M is the multicast group, L is the set of unidirectional links in the tree, and T_l is the set of nodes in the subtree induced by the generic link l. Figure 10 shows the minimum TF allocation on a CBT with core F, 3 active nodes, and $b = 2$.

Theorem 3 The minimum TF allocation $R^D_{CBT,min}$ is tree schedulable on any CBT.

Proof Since each packet reaches the core before being forwarded to the members of the multicast group, the core can be seen as a static source of all the traffic addressed to the group. The total amount of bandwidth allocated from the core for transmission to the group is B TFs; hence the overall TF allocation is given by $B(N-1)$, in accordance to Equation (1), which guarantees tree schedulability from the core to the leaves. There is the additional constraint that identity within the time cycle of the B TFs must be the same on all the core's outgoing links.
$\min\{|M \backslash T_l| b, B\}$ TFs are allocated on each link on the path from the leaves to the core in a way such that their tree schedule includes the B TFs reserved on the core's outgoing links. It is always possible to build this tree schedulable allocation starting from the core (assuming that resources are available on the links) because the path from the leaves to the core is a linear one. Thus, given the set of TFs reserved on the core's outgoing link $\{t_0, \ldots, t_{B-1}\}$, the TFs $\{t_0 - 2, \ldots, t_{B-1} - 2\}$ are reserved on the upstream link. In general, given a link which is h hops from the core, the TFs $\{t_0 - 2(h+1), \ldots, t_{B-1} - 2(h+1)\}$ must be allocated on it[5]. □

[5] The operations on TF numbers are made modulo the time cycle length k.

Theorem 4 The minimum TF allocation $R^D_{CBT,min}$ is not dynamically assignable on any possible CBT.

Proof The proof can be given in a way similar to what is done for general trees in Section 4.3.3 by showing a sample minimum allocation which is not dynamically assignable. Figure 10 depicts a minimum TF allocation for $b = 2$ and 3 active nodes; node N and node M cannot be active at the same time, i.e., then the allocation is not dynamically assignable. □

Theorem 5 The allocation

$$R^D_{CBT,upper} = 2B(N-1) \text{ TFs}$$

is tree schedulable and dynamically assignable on any CBT.

Figure 11 shows a tree schedulable and dynamically assignable TF allocation on a CBT with core F, $b = 2$, $N_a = 3$, and $k = 10$.

Proof Tree schedulability is assured by Theorem 3.
Dynamic assignability is guaranteed by the fact that given any set of N_a active nodes, each one can be uniquely assigned b TFs, say $\{\tau_0, \ldots, \tau_{b-1}\}$, out of the B allocated on the core's outgoing links. The TFs assigned to an active node on the other links are those which are either contained in the tree schedule of $\{\tau_0, \ldots, \tau_{b-1}\}$, or contain $\{\tau_0, \ldots, \tau_{b-1}\}$ in their tree schedule. □

Theorem 6 The allocation

$$R^D_{CBT} = B(2N - N_a - 1) \text{ TFs} \tag{5}$$

is tree schedulable and dynamically assignable on any Core Based Tree having at least N_a leaves.

Proof B TFs are allocated on each link from the core to the leaves leading to an overall allocation of $B(N-1)$ TFs. Dynamic assignability can be obtained by having N_a leaves which are assigned in a fixed manner N_a different sets of b TFs. Given each of these N_a leaves, the TFs $\{\tau_0 - 2(h+1), \ldots, \tau_{b-1} - 2(h+1)\}$ are allocated on each link on the path from the leave to the core, where h is the number of hops from the considered link to the core. The allocation considered so far guarantees that the N_a leaves can be active concurrently.
In order to allow any other node n to become active at the same time as $N_a - 1$ of the above leaves, it must be possible to assign to n any subset of the B TFs allocated on the core's outgoing links. Thus, B TFs must be reserved on the path from n to the core in the way described in the proof of Theorem 3. As a consequence, B TFs are allocated on each of the $(N-1)$ links towards the core, with the exception of the outgoing links of the

N_a given above; on those links only b TFs are reserved. Thus, the total allocation is given by:

$$N_a \cdot b + B(N - 1 - N_a) + B(N - 1) = B(2N - N_a - 1).$$

□

Figure 12 shows a minimum tree schedulable and dynamically assignable TF allocation on a CBT with core F, $b = 2$, $N_a = 3$, and $k = 10$.

4.3.5 Reservation Ratio

The reservation ratio is devised for the upper and lower bounds of the resource allocation that enable TDP immediate forwarding over a tree. The upper bound of the reservation ratio is obtained from Equation (4) as

$$\rho_T^D = \frac{N \cdot b(N-1)}{2B(N-1)} = \frac{N}{2N_a}.$$

The upper bound of the reservation ratio is directly proportional to the number of members in the multicast group. If more than half of the members are active ($N_a > N/2$), the resource allocation on a tree is smaller than on a ring ($\rho_T^D < 1$). Otherwise, when active nodes are less than half of the members ($N_a > N/2$), the ring may possibly require less resources than the tree ($\rho_T^D > 1$). In essence, the tree is the most effective structure if the subset of active nodes is large.

ρ_T^D shows that the upper bound on the reservation ratio increases with the dimension of the group and suggests that dynamic multicast is less expensive over a ring when the multicast group is large. The lower bound on the allocation for large groups $\hat{R}_{T,min}^D$ given in Equation (3) is used to devise a lower bound on the reservation ratio

$$\hat{\rho}_{T,min}^D = \frac{N_a \cdot b(N-1)}{2B(N-1)} = \frac{1}{2}.$$

Thus, we can conclude that the relative cost (in terms of allocated resources) of tree versus ring embedding for large scale dynamic multicast depends on the topology of the structure (and thus, finally, of the underlying network).

Considering dynamic multicast over a CBT, the upper bound of the reservation ratio is devised from Equation (5) as

$$\rho_{CBT}^D = \frac{B(2N - N_a - 1)}{2B(N-1)} = \frac{2N - N_a - 1}{2(N-1)}$$

Thus, dynamic multicast with $N = N_a$ over a CBT requires half the resources required over a ring, for any possible embedding of the ring as an Euler tour of the ring. As the size of the group increases with respect to the number of active nodes ($N >> N_a$), ρ_{CBT}^D asymptotically approaches 1.

It is worth noticing that if the ring is embedded as a traveling salesman tour among the members of the multicast group, the size of the ring is half and thus the reservation ratio is always greater than 1 (i.e., the ring requires less resources than the CBT).

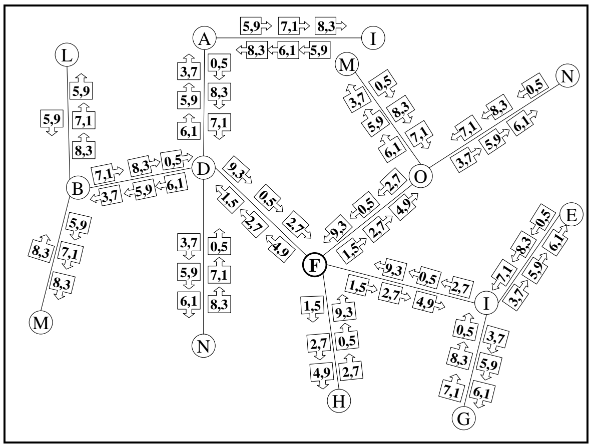


Figure 12: Minimum Tree Schedulable and Dynamically Assignable Allocation on a Core Base Tree.

4.3.6 Updating State Information or Coordination Complexity

The state information needed to operate dynamic multicast over a tree encompasses the identity of (1) the TFs allocated to the multicast group on the outgoing links of each node, and (2) the b TFs each active node is assigned for the transmission of its data. The first piece of information is distributed at the beginning of the operation of the multicast group and is not changed over time. The second one has to be updated each time the identity of the active nodes changes. A signaling protocol must be used to allow nodes which change state (from active to passive and vice versa) to notify all the other nodes and to enable the distributed assignment of TFs.

In principle, the active nodes which do not change their state do not need to change their TF assignment. In fact, when a node becomes active, there is always another node turning passive; the former could take over the TFs previously assigned to the latter. However, the TF allocation that enables this way of operation may be larger than one that requires the reconsideration of TF assignments to all the nodes whenever two of them modify their status.

4.4 Adaptive Multicast

Since it is possible that only one node is active, each group member must be enabled to transmit up to B TFs. Thus, B TFs are allocated on every outgoing link of each group member. The same number of TFs must be allocated also on every incoming link and the total TF allocation is given by $R_{T,min}^A = 2B(N-1)$. $R_{T,min}^A$ is a lower bound on the TF allocation enabling adaptive multicast. A TF allocation must be both dynamically assignable and tree schedulable on any tree to enable adaptive multicast with guaranteed quality of service.

Corollary 2 The allocation

$$R_{T,min}^A = 2B(N-1) \text{ TFs}$$

is not tree schedulable on every tree.

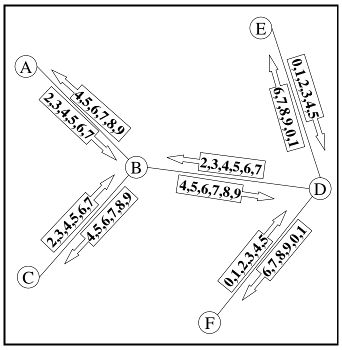


Figure 13: Minimum TF Allocation for Adaptive Multicast.

Proof Figure 13 shows the minimum TF allocation on a sample tree where $N = 6$ and $B = 6$; thus, 6 TFs are allocated on each link for a total of $2 \cdot 6(6-1) = 60$ TFs. The allocation is not tree schedulable because not all the TFs allocated on link FD have their tree schedules included in the allocation. The portion of tree schedule on link DE is missing since the TFs reserved on link DE are chosen in order to accommodate the tree schedule of the TFs allocated on the links BD, AB and CB. □

A larger TF allocation (12 TFs more) is needed to guarantee tree schedulability on the tree given in Figure 13.

Theorem 7 The allocation

$$R^A_{CBT} = 2 \cdot B \cdot (N-1)\text{TFs} \tag{6}$$

is tree schedulable on any CBT and dynamically assignable for a variable number of active nodes, i.e., it enables adaptive multicast.

Proof Theorem 5 states that the above allocation (called $R^D_{CBT,upper}$ in that theorem) is dynamically assignable for a dynamic group of N_a active nodes. Since the number of allocated TFs does not depend on the actual number of active nodes, the allocation is dynamically assignable for any number of active nodes $N_a \leq N$. □

Corollary 3 The allocation

$$R^A_T = N \cdot B \cdot (N-1)\text{TFs} \tag{7}$$

is tree schedulable on any tree and dynamically assignable for a variable number of active nodes.

Proof Equation (7) is derived from Equation (4) assuming $b = B$, i.e., $N_a = 1$. Thus, Corollary 1 assures tree schedulability and dynamic assignability for dynamic multicast with 1 active node. If $N_a > 1$, each active node is using less then B TFs. Since each source could use any of the B reserved TFs on its outgoing link, it chooses any $b = B/N_a$ of them to transmit its data. Tree schedulability and dynamic assignability are maintained since they do not depend on whether the number of active nodes changes over time. □

4.4.1 Reservation Efficiency

The upper bound on the reservation ratio to perform adaptive multicast over a general topology tree can be devised from Equation (7) as

$$\rho^A_T = \frac{N \cdot B(N-1)}{2B(N-1)} = \frac{N}{2}$$

ρ^A_T shows that the upper bound of the reservation ratio for adaptive multicast is directly proportional to the size of group. Thus, the ring is more convenient than the tree in terms of required resources, especially for large multicast groups.

The reservation ratio over a CBT is $\rho^A_{CBT} = 1$, as devised from Equation (6): the amount of resources required to perform adaptive multicast over a CBT and a ring is the same.

4.4.2 Updating State Information or Coordination Complexity

Adaptive multicast requires the same state information as dynamic multicast. The number of active nodes is changing over time, and thus, the TF assignment of each active node changes whenever a node changes its status. A signaling protocol must be used to allow nodes which change status (from active to passive and vice versa) to notify all the other active nodes and to coordinate the distributed assignment of TFs. Such a protocol is critical since the correct set of TFs must be assigned to all the active nodes; if any of the active nodes fails to get the state information update, the operation of the whole multicast group may be disrupted.

Moreover, as the number of active nodes N_a changes over time, the fair share of resources reserved to the multicast group (B/N_a) can be a non-integer number of TFs. As a consequence, to guarantee maximum fairness, the protocol should be able to assign $\lfloor B/N_a \rfloor$ TFs to each node and alternately assign the remaining TFs to all the active nodes. This requires the active nodes to exchange control information when the set of active nodes is not changing. Instead, as it was discussed in Section 3.3, sharing one TF among multiple active nodes over a ring is simple.

5 Conclusions

This section first summarizes the main results and then discusses some open issues.

5.1 Comparison Summary and Discussion

Real-time group multicast (many-to-many) with deterministic quality of service guarantees is a challenging problem. In this paper we study this problem in the context of time-driven priority with immediate forwarding. Specifically, the main objective of this manuscript is to increase the understanding of the real-time multicast problem in two basic network configurations: (i) tree embedding - the approach popular on general networks, and (ii) ring embedding - the approach traditionally used in local area networks [EG METANET]. The work focused on increasing the understanding of the tree versus ring embedding, rather than attempting to provide a specific design for a specific network, such as the Internet.

In order to provide a comprehensive evaluation of ring versus tree embedding, the following multicast scenarios

Table 1: Sharing of State Information among Group Members.

	Ring	Tree
Static Mcast	None	None
Dynamic Mcast	None	Identity of assigned TFs
Adaptive Mcast	Number of active nodes	- Number of active nodes - Identity of assigned TFs

were investigated: (*i*) static - fixed subset of active nodes, (*ii*) dynamic - fixed number of active nodes (i.e., the subset of active nodes is changing over time, but its size remains constant), and (*iii*) adaptive - the number and identity of active nodes change over time. The results are interesting and often counter-intuitive, since, despite of its lower network delay, embedding a tree is not always the best strategy. In particular, dynamic and adaptive group multicast on a tree requires a signaling protocol for continuously updating state information and coordinating the operation of the group during the communication session and not only during the setup phase. Such signaling protocol is not required on a ring where the circular topology with simple implicit token passing is sufficient; these results are summarized in Table 1. Moreover, as summarized in Table 2, the bandwidth allocation on the ring for the above three traffic scenario is $O(N)$; on the tree it is $O(N)$ only for the static traffic scenario, while for dynamic and adaptive multicast it is $O(N^2)$. Only if a core based tree is used, then dynamic and adaptive multicast requires a bandwidth allocation of $O(N)$.

5.2 Open Issues and Challenges

As was mentioned in Section 2 a global common time reference is required to implement time-driven priority (TDP). This used to be a major hurdle. However, today a global common time reference has been established by the time-of-day international standard that is called Coordinated Universal Time or UTC (a.k.a. Greenwich Mean Time or GMT). Specifically, time is measured by counting the oscillations of the cesium atom **in multiple locations**; in fact, 9,192,631,770 oscillations of the cesium atom define one UTC second. UTC is available everywhere around the globe from several distribution systems, such as, GPS [16] (USA satellites system), GLONASS [15] (Russian Federation satellites system), and in the future by Galileo [27] (European Union and Japanese satellites system). There are other means for distribution of UTC, such as, CDMA cellular phone system and TWTFT (Two-Way Satellite Time and Frequency Transfer) [11] technique based on communications satellites. UTC receivers from GPS are available from many vendors for a low price – for example, the price of a one PPS (pulse per second) UTC clock, with accuracy of 10-20 nanoseconds, is about $200. By combining UTC from GPS with local Rubidium or Cesium clocks it is possible to have a correct UTC (within 1 micro-second) without an external time reference from GPS/GLONASS/Galileo for days (with Rubidium clocks) and months (with Cesium clocks).

Table 2: Resource Allocation to Multicast Groups.

	Ring	Tree	
		Generic	Core Based
Static Mcast	$2N_a \cdot b(N-1)$	$N_a \cdot b(N-1)$	$N_a \cdot b(N-1)$
Dynamic Mcast	$2N_a \cdot b(N-1)$	$N \cdot b(N-1)$	$2N_a \cdot b(N-1)$
Adaptive Mcast	$2N_a \cdot b(N-1)$	$N \cdot N_a \cdot b(N-1)$	$2N_a \cdot b(N-1)$

There are still a number of important open issues for further study:

- The analysis in this paper focused on TDP with immediate forwarding, which implies a strict scheduling in the next time frame, namely, a data packet received in time frame i should be scheduled for transmission on time frame $i+1$. However, a more flexible scheduling called non-immediate forwarding can be used, see the discussion and analysis in [12]. Non-immediate forwarding is interesting since it may change the analysis and results presented in this manuscript.
- How to handle network failures in order to minimize the disruption to real-time group multicast. Specifically, to study network fault detection, diagnostic, and recovery, while maintaining the integrity of each one of the multicast groups.
- How to dynamically change the membership in the real-time group multicast, namely, how to realize run-time join and leave operations and how they affect system performance. Such operations are not trivial since they may require changes to the group topology.
- How to combine real-time group multicast with reliable group (many-to-many) multicat (e.g., with the scheme described in [19]).

References

[1] A. Adams, J. Nicholas, and W. Siadak. Protocol independent multicast-dense mode (PIM-DM): Protocol specification (revised). Internet Draft - Protocol Independent Multicast (PIM) Working Group, Internet Engineering Task Force, February 2002.

[2] M. Baldi, Y. Ofek, and B. Yener. Adaptive group multicast with time-driven priority. *IEEE/ACM Transactions on Networking*, 8(1):31–43, February 2000.

[3] A. Ballardie. Core based trees (CBT) multicast routing architecture. Experimental RFC 2201, Internet Engineering Task Force, September 1997.

[4] A. Ballardie. Core based trees (CBT version 2) multicast routing – protocol specification. Experimental RFC 2189, Internet Engineering Task Force, September 1997.

[5] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees (CBT). In Deepinder P. Sidhu, editor, *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 85–95, San Francisco, California, September 1993. ACM.

[6] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queuing algorithm. *ACM Computer Communication Review (SIGCOMM'89)*, pages 3–12, 1989.

[7] D. Estrin et al. Protocol independent multicast-sparse mode (PIM-SM): Protocol specification. Experimental RFC 2362, Internet Engineering Task Force, June 1998.

[8] D. Ferrrari, A. Banerjea, and H. Zhang. Network support for multimedia. a discussion of the tenet approach. *Computer Networks and ISDN Systems*, 26:1267 – 1280, 1994.

[9] A. Gupta, W. Heffner, M. Moran, and C. Szyperski. Network support for realtime multi-party applications. In *Fourth Interna-*

tional Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'93), Lancaster, England, November 1993.

[10] A. Gupta, W. Howe, M. Moran, and Q. Nguyen. Resource sharing for multi-party real-time communication. In *IEEE INFOCOM'95*, Boston, MA, April 1995.

[11] National Physical Laboratory. Two-way Satellite Time and Frequency Transfer (TWTFT). http://www.npl.co.uk/-npl/ctm/twstft.html.

[12] C-S. Li, Y. Ofek, A. Segall, and K. Sohraby. Pseudo-isochronous cell forwarding. *Computer Networks and ISDN Systems*, 30:2359 – 2372, 1998.

[13] C-S Li, Y. Ofek, and M. Yung. "Time-Driven Priority" flow control for real-time heterogeneous internetworking. In *IEEE INFOCOM '96*, 1996.

[14] J. Moy. Multicast extensions to OSPF. Standards Track RFC 1584, Internet Engineering Task Force, March 1994.

[15] Russian Federation Ministry of Defense Coordination Scientific Information Center. Global Navigation Satellite System GLONASS. http://www.rssi.ru/SFCSIC/english.html.

[16] National Institute of Standards and Technology (NIST). Global Positioning System data archive. http://www.boulder.nist.gov/timefreq/service/gpstrace.htm.

[17] Y. Ofek. Generating a fault tolerant global clock using high-speed control signals for the MetaNet architecture. *IEEE Transactions on Communications*, pages 2179–2188, May 1994.

[18] Y. Ofek and M. Faiman. Distributed global event synchronization in a fiber optic hypergraph network. In *The 7th Intl. Conference on Distributed Computing Systems*, pages 307–314, 1987.

[19] Y. Ofek and B. Yener. Reliable concurrent multicast from bursty sources. *IEEE Journal on Selected Areas in Communications*, 15(3):434–444, April 1997.

[20] Y. Ofek, B. Yener, and M. Yung. Concurrent asynchronous broadcast on the MetaNet. *IEEE Transactions on Computers*, 46(7):737–748, July 1997.

[21] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The sigle-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[22] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[23] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. Standard Track RFC 1889, IETF's Audio-Video Transport Working Group, January 1996.

[24] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. Standard Track RFC 2212, Internet Engineering Task Force, September 1997.

[25] S. Shenker and J. Wroclawski. General characterization parameters for Integrated Service network elements. Standard Track RFC 2215, Internet Engineering Task Force, September 1997.

[26] T. Turletti and C. Huitema. Videoconferencing on the Internet. *IEEE/ACM Transactions on Networking*, 4(3):340 – 351, June 1996.

[27] European Union. Transport-satellite navigation. http://europa.eu.int/scadplus/leg/en/lvb/l24205.htm.

[28] D. Waitzman, C. Partridge, and S. Deering. Distance Vector Multicast Routing Protocol. Experimental RFC 1075, Internet Engineering Task Force, November 1988.

[29] J. Wroclawski. Specification of the controlled-load network element service. Standard Track RFC 2211, Internet Engineering Task Force, September 1997.

[30] J. Wroclawski. The use of RSVP with IETF Integrated Services. Standard Track RFC 2210, Internet Engineering Task Force, September 1997.

A Notation

The following notation has been used throughout the paper.

CBT core based tree.

D^{α}_{β} network delay bound over the structure β implementing the multicast method α. The structure can be either a ring ($\beta = R$), or a general topology tree ($\beta = T$), or a core based tree ($\beta = CBT$). The multicast method can be one of static ($\alpha = S$), dynamic ($\alpha = D$), or adaptive ($\alpha = A$).

D_b maximum buffering time, expressed in number of time frames, experienced by packets into the buffering node on a ring used for dynamic and adaptive multicast.

$h(n)$ the height, measured in number of links, of the subtree rooted by n. In general $h(n) = o(\log N)$.

$H = \max_{n \in M} h(n)$ is the diameter of the tree.

k number of TFs in a time cycle.

L set of unidirectional links of a tree.

M as the set of nodes participating to the multicast group.

N as the number of members of the multicast group, i.e., the cardinality of M.

N_a as the number of sources in the multicast group; these are also said *active nodes*.

M_a as the set of active nodes participating to the multicast group.

B as the total number of TFs per time cycle that active nodes of the multicast group M use for transmitting their packets.

b as the number of TFs per time cycle assigned to a single active node. Without loss of generality we consider that active nodes fairly share the bandwidth allocated to the multicast group and thus $b = B/A$.

R^{α}_{β} Resource allocation over the structure β implementing the multicast method α. The structure can be either a ring ($\beta = R$), or a general topology tree ($\beta = T$), or a core based tree ($\beta = CBT$). The multicast method can be one of static ($\alpha = S$), dynamic ($\alpha = D$), or adaptive ($\alpha = A$).

ρ^{α}_{β} Reservation ratio over the structure β implementing the multicast method α. The structure can be either a ring ($\beta = R$), or a general topology tree ($\beta = T$), or a core based tree ($\beta = CBT$). The multicast method can be one of static ($\alpha = S$), dynamic ($\alpha = D$), or adaptive ($\alpha = A$).

TDP Time-Driven Priority.

TF Time Frame.

T_f as the duration of a TF.

T_l the set of links constituting the subtree induced by link l.