Optimal Branching Factor for Tree-based Reliable Multicast Protocols

Christian Maihöfer[a] and Kurt Rothermel[b]
E-Mail {christian.maihoefer|kurt.rothermel}@informatik.uni-stuttgart.de
Phone +49 731 5052173 Fax +49 731 5054201

[a]DaimlerChrysler AG, Research and Technology, PO Box 2360, D-89013 Ulm, Germany

[b]University of Stuttgart, IPVR, Breitwiesenstr. 20-22, D-70565 Stuttgart, Germany

In recent years, many reliable multicast protocols on transport layer have been proposed. Previous analysis and simulation studies gave evidence for the superiority of tree-based approaches in terms of throughput and bandwidth requirements.

In many tree-based protocols, the nodes of the tree are formed of multicast group members. In this case, the branching factor, i.e. the maximum number of child nodes is adjustable. In this paper we analyze the influence of the branching factor on a protocol's throughput and bandwidth consumption. This knowledge is important to configure protocols for best performance and to optimize the tree creation process.

Our results show that the optimal branching factor depends mainly on the probability for receiving messages from other local groups. If local groups are assigned to a separate multicast address, the optimal branching factor is small. On the other hand, if TTL scoping is used and therefore the probability for receiving messages from other local groups is greater than zero, larger local groups provide better performance.

Keywords: reliable multicast, branching factor, scope overlapping, TTL, throughput, bandwidth

1. Introduction

Multicast transport protocols use positive or negative acknowledgment schemes to ensure reliable message delivery. A positive acknowledgment returned by a receiver confirms correct message delivery, whereas a negative acknowledgment asks for a message retransmission. It has been shown that tree-based multicast protocols scale better than other multicast schemes suggested in the literature [1–4]. In tree-based protocols, the members of a multicast group are organized in a so-called ACK tree to overcome the well-known acknowledgment implosion problem, i.e., overwhelming of the sender by a large number of positive (ACK) or negative (NAK) acknowledgment messages. Since acknowledgments are propagated along the edges of the ACK tree in a leaf-to-root direction, the implosion problem can be avoided by limiting the branching factor of a node. Note that the ACK tree has to be distinguished from the routing tree. The routing tree is established for reliable multicast protocols as well as for unreliable ones and necessary to deliver initial multicast transmissions. How the routing tree is created is outside the scope of this paper. We assume that the ACK tree is created independent of the routing tree on transport layer, because combining both trees requires special router support that is not given in the current Internet.

Our used terminology is as follows. A node in the ACK tree having children (i.e., a non-leaf node) is defined to be a group leader. A group leader together with its children form a so-called local group. We define a tree's branching factor to be the maximum number of child nodes that can be associated with a group leader. We will use the notion of a global group to denote all members of the multicast group.

The sender of a multicast group represents the root of the corresponding ACK tree, while the other nodes of the tree are the members of the global group. The ACK tree can be created by techniques like expanding ring search (ERS) [5] or the Token Repository Service [6]. Whenever a new member wants to join a multicast group, a node in the corresponding ACK tree has to be selected by one of these techniques to become the group leader of the new member. ERS as well as the Token Repository Service allow choosing the appropriate branching factor. The results reported in this paper will help to find the optimal branching factor.

In order to save network bandwidth and processing power the scope of retransmission messages should be confined to local groups since a group leader is responsible to retransmit messages for its local group members only. If multicast communication is used for retransmissions also, this poses the problem of how to limit the scope. The literature proposes two approaches to deal with this problem. The first one is to assign a separate multicast address to each local group. Retransmissions are sent to the multicast address of the local group and therefore are only received by the members of this group. The other approach is to use TTL scoping [5]. Retransmissions are sent with a TTL value that was measured before and is equal to the maximum distance between the group leader and all of its local group members. Consequently, not only each local group member will receive the retransmitted messages but likely also members of other local groups that are within the corresponding TTL distance.

While attractive at the first glance, the approach to assign multicast addresses to local groups has some serious drawbacks. Most importantly, there may be a large number of additional multicast groups for each of which a network layer (IP) routing tree must be created and maintained, which results in a scalability problem on network layer. That is why we consider the TTL scop-

ing approach in this paper. With TTL scoping retransmissions may be received outside the target local group from members of neighboring local groups. This leads to additional processing and bandwidth overhead that has to be considered in an analysis, which sets our work apart from previous analytical work. A small branching factor, i.e. a small number of directly attached children to a group leader, usually should lead to low load on each group leader. However, if local multicast groups are not perfectly confined, a small branching factor may result in increased load on each group leader because a small branching factor leads to more local groups and therefore more messages received outside the scope of local groups.

Our results of a processing and bandwidth requirements analysis show that the optimal branching factor depends mainly on the used reliable multicast protocol and the probability for receiving retransmissions destined to other local groups, which we will denote as scope overlapping probability. If the scope overlapping probability is low, a small branching factor results in the highest throughput and lowest bandwidth consumption. On the other hand, if the scope overlapping probability grows, the optimal branching factor increases also.

The remainder of this paper is structured as follows. In the next section we discuss related work. Section 3 gives an overview and classification of the considered protocols. Our analysis in Section 4 starts with the definition of the assumed system model followed by detailed formulas for the bandwidth consumption and throughput. To illustrate the influence of the branching factor on the protocols' performance, numerical evaluations are presented in Section 5. Finally, we conclude our work with a brief summary.

2. Related Work

Reliable multicast protocols were already analyzed in previous work. The first work in this area was presented by Pingali et al. [7]. They have compared the processing requirements of sender- and receiver-initiated protocols. Levine et al. [1] have extended this analysis to the class of ring- and tree-based approaches and showed that tree-based approaches are superior. Bandwidth analysis of generic reliable multicast protocols were done by Kasera et al. [8], Nonnenmacher et al. [9] and Poo et al. [10]. In [8], local recovery techniques are analyzed and compared. In Nonnenmacher et al. [9] they studied the performance gain of protocols using parity packets to recover from transmission errors. In [10], non-hierarchical protocols are compared.

Our paper extends previous work in the following ways. The major difference is that we consider overlapping retransmission scopes, which significantly influences the results. Second, we consider the loss of control packets rather than assuming reliable delivery. Third, we assume that local clocks are not synchronized, which influences the NAK suppression scheme. Fourth, in contrast to previous work in this area, we made simulations to confirm the analytical results. Finally, this work is the first one that focuses on the branching factor.

3. Classification of Tree-based Multicast Protocols

3.1. ACK-based Protocol (H1)

The first considered scheme is denoted as (H1) in conformity with [1]. As in all other protocol classes we assume that the initial sender is the root of the ACK tree and that the initial transmission is multicasted to the global group. (H1) uses unicast ACKs sent by receivers to their group leaders to indicate correctly received packets. Each group leader that is not the root node also sends an ACK to its parent as soon as a data packet has been received. If a timeout for an ACK occurs at a group leader, a multicast retransmission is invoked for this local group. As explained in the introduction such a retransmission can be sent to a separate multicast address for this local group or sent to the global group address and limited in scope by the TTL value. An example of a protocol similar to our definition of (H1) is RMTP [11].[1]

3.2. NAK-based Protocol (H2)

The second scheme (H2) is based on NAKs with NAK suppression. NAKs are sent by means of multicast to the group leader and other nodes of this local group. A receiver that misses a data packet sends a NAK provided that it has not already received a NAK from another receiver that also misses the data packet. NAKs alone do not allow a deterministic decision when packets can be removed from memory. Therefore, selective ACKs (SAKs) are sent after a certain number of packets has been received or after a certain time period has been expired, to propagate the state of a receiver to its group leader. TMTP [5] is an example for class (H2).

3.3. Protocols with Aggregated Acknowledgments (H3) and (H4)

Protocol (H3) and protocol (H4) are based on protocol (H1) or (H2), respectively. Additionally, they implement aggregated ACKs, so called AAKs. In contrast to normal ACKs, they are sent to confirm the correct message delivery for a whole subhierarchy of the control tree. A group leader sends an AAK after it has received an AAK from each child node. AAKs are necessary to guarantee reliable delivery even in case of node failures. A detailed discussion of these protocol classes is given in [12].

4. Analysis

4.1. Model

Our model is similar to the one used by Pingali et al. [7] and Levine et al. [1]. This means, that our analysis is based on the per-packet processing and bandwidth requirements. A single sender is assumed, multicasting to R identical receivers. We assume that nodes do not fail, i.e. transmissions are eventually successful. In contrast to previous work, packet loss can occur on both data packets *and* control packets. The multicast packet

[1] However, please note that RMTP uses router support to create an ACK tree similar to the routing tree.

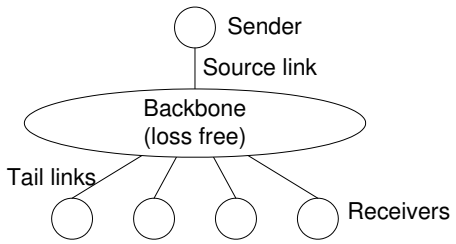

Figure 1. System Model

loss probability is given by q and unicast packet loss probability by p.

In contrast to Pingali et al. [7] and Levine et al. [1] we assume spatially dependent losses. This assumption is reasonable, since receivers share parts of the multicast routing tree. In [13] the spatial loss correlation in the Internet and MBone is studied in detail. They found only small correlation among the multicast sites except for the loss due to the link next to the source, which is highly correlated. Therefore, we use the system model shown in Figure 1, which considers spatial correlation due to loss on the first link from the sender to the backbone. The sender is connected with an error-prone link to the backbone. An error on this link will be seen by all receivers. According to the observations of Yajnik et al., which have revealed only small loss probability in the backbone, we consider the backbone as error free. Finally, each receiver is connected to the backbone with an error-prone link. Errors on this tail links are assumed to be mutually independent. Note, that this model is similar to [8] and [9].

4.2. General Analytical Methods

In this section we present the used analytical methods without considering the protocol classes and without considering local groups. The main issue for our analysis is to obtain the number of necessary transmissions M to deliver a data packet correctly to all receivers. Many other quantities, like the number of ACK or NAK packets and the number of timeouts that have to be processed depend on M. The expected total number of necessary transmissions $E(M)$ to receive the data packet correctly at all receivers is the sum of the retransmits due to loss on the source link, $E(M_S)$, and retransmits due to loss on the tail link, $E(M_T)$, (or ACK loss in case of protocol (H1) and (H3)) and the initial transmission:

$$E(M) = E(M_S) + E(M_T) + 1. \tag{1}$$

To obtain the values for $E(M_S)$ and $E(M_T)$ some preceding steps have to be performed. We assume q_D to be the multicast end-to-end loss probability perceived by a random receiver. According to the system model described above and shown in Figure 1, the end-to-end loss probability is supposed to be equally split between the source link loss $q_D\prime$ and tail link loss $q_D\prime$:

$$q_D\prime = 1 - \sqrt{1 - q_D}. \tag{2}$$

Analogous to M, which is the total number of data packet transmissions for all receivers, M_r denotes the number of necessary data packet transmissions for a single receiver r. To obtain the number of retransmissions due to loss on the source link, M_S, this part of the system model can be assumed as consisting of a sender and a single receiver, which is the backbone. The necessary number of transmissions for a single receiver follows from the Bernoulli distribution. This means, M_r counts the number of trials until the first success occurs. The probability for the first success in a Bernoulli experiment at trial k with probability for success $(1-q)$ is:

$$P(X = k) = (1-q)q^{k-1}. \tag{3}$$

The expectation follows to (see [7]):

$$E(M_r) = \frac{1}{1-q} \tag{4}$$

$$E(M_r | M_r > x) = \frac{x + 1 - xq}{1-q} \tag{5}$$

$$P(M_r > x)[E(M_r | M_r > x) - x] = E(M_r) - x. \tag{6}$$

The number of retransmissions due to loss on the source link is now with $q_D\prime$ and excluding the initial transmission:

$$E(M_S) = \frac{1}{1 - q_D\prime} - 1. \tag{7}$$

Now we have to obtain the value for M_T. M_{single} is the necessary number of transmissions for a single receiver due to loss on the tail link. M_{single} depends on the probability $\tilde{p}$ that a retransmission is made. $\tilde{p}$ is the failure or retransmission probability for a single receiver and is made up of the data and control packet loss probabilities (see following sections). With $\tilde{p}$, the probability that the number of necessary transmissions M_{single} for a certain receiver is smaller or equal to m (m=1, 2, ...) is:

$$P(M_{single} \leq m) = 1 - \tilde{p}^m. \tag{8}$$

On the tail links, where the packet losses at different receivers are assumed to be independent from each other, the following for $E(M_T)$, the expected number of necessary transmissions to receive the data packet correctly at all receivers, holds [7]:

$$\begin{aligned} P(M_T \leq m) &= \prod_{r=1}^{B} P(M_{single} \leq m) \\ &= (1 - \tilde{p}^m)^B = \sum_{i=0}^{B} \binom{B}{i} (-1)^i \tilde{p}^{im} \end{aligned} \tag{9}$$

$$\begin{aligned} P(M_T = m) &= P(M_T \leq m) - P(M_T \leq m-1) \\ E(M_T) &= \sum_{m=1}^{\infty} m P(M_T = m) \\ &= \sum_{i=1}^{B} \binom{B}{i} (-1)^{i+1} \frac{1}{1 - \tilde{p}^i}. \end{aligned} \tag{10}$$

$\tilde{p}$ for the various protocol classes will be determined later in the analysis.

Finally, we have to obtain the number of group leaders. The number of nodes R in a complete tree with

branching factor B and height h is:

$$R = \sum_{i=0}^{h-1} B^i = \frac{1-B^h}{1-B} \Rightarrow h = \log_B\Big(R(B-1)+1\Big). \quad (11)$$

G, the number of group leaders follows to:

$$G = \sum_{i=0}^{\log_B[R(B-1)+1]-2} B^i. \quad (12)$$

4.3. ACK-based Protocol (H1)

Our analysis distinguishes among the three different kinds of nodes in the ACK tree, the initial sender at the root of the tree, the receivers that form the leaves of the ACK tree and the group leaders, which are inner nodes. A group leader is a sender and receiver as well.

The analysis is based on the assumption that each local group consists of exactly B members and one group leader. We assume further, that when a group leader has to retransmit a message, the group leader has already received this packet correctly. The following subsections analyze the processing requirements at the sender, receivers and group leaders.

4.3.1. Sender (Root Node)

Protocol (H1) uses unicast ACKs for controlling the reliable message delivery. To obtain the maximum throughput we analyze the processing times at the sender P_S^{H1}, at a receiver P_R^{H1} and at a group leader P_G^{H1}. The throughput is then limited by the maximum processing requirements at the sender, receivers or group leaders.

The analysis is based on the necessary requirements for sending a single data packet correctly to all receivers. We assume that the sender waits until all ACKs are received and then sends a retransmission if necessary. The CPU processing load is illustrated in Figure 2.

At the sender we have:

$$P_S^{H1} = X_f + X_d(1) + \sum_{m=2}^{M^{H1}} \Big(X_t(m-1) + X_d(m)\Big) + \sum_{i=1}^{\widetilde{L}^{H1}} X_a(i). \quad (13)$$

X_f is the processing time required to feed in a new data packet from a higher protocol layer. $X_a(i)$ denotes the processing requirement to receive an ACK packet for the i-th transmission. Analogous, $X_t(m)$ and $X_d(m)$ are the processing requirements for a timer interrupt or data packet transmission for the m-th transmission. M^{H1} is the total number of transmissions necessary to transmit a packet correctly to all receivers in the presence of data packet and ACK loss and $\widetilde{L}^{H1}$ is the total number of ACKs received for this packet. Timer interrupts must be processed only if not all ACKs are received, i.e. when a retransmission is necessary. Therefore, for the last, successful transmission no timer processing is considered.

In the following equations we consider only expectations, since we are always interested in the mean results. $E(P_S^{H1})$ is the expectation of the processing requirement at the sender:

$$E(P_S^{H1}) = E(X_f) + E(M^{H1})E(X_d) + \Big(E(M^{H1})-1\Big)E(X_t) + E(\widetilde{L}^{H1})E(X_a). \quad (14)$$

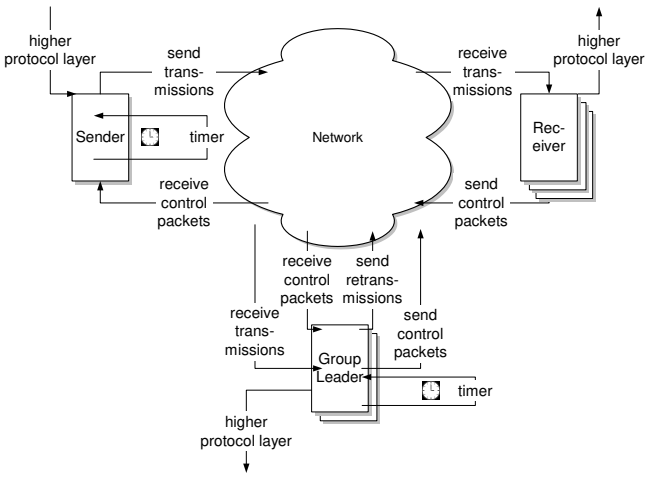

Figure 2. CPU processing load of ACK-based protocols

The bandwidth requirement is given by W_S^{H1}:

$$E(W_S^{H1}) = E(M^{H1})E(W_d) + E(\widetilde{L}^{H1})E(W_a), \quad (15)$$

where W_d and W_a are the necessary bandwidths for a data packet or ACK packet, respectively. The only unknowns are $E(M^{H1})$ and $E(\widetilde{L}^{H1})$. $E(M^{H1})$, the expected number of necessary transmissions, is determined by the probability for a retransmission:

$$\tilde{p} = q_D\prime + (1-q_D\prime)p_A, \quad (16)$$

i. e. either a data packet is lost ($q_D\prime$) or the data packet is received correctly and the ACK is lost ($(1-q_D\prime)p_A$).

The number of retransmissions due to loss on the tail link or ACK loss is (see Eq. 10):

$$E(M_T^{H1}) = \sum_{i=1}^{B} \binom{B}{i} (-1)^{i+1} \frac{1}{1-\tilde{p}^i} - 1. \quad (17)$$

$E(M^{H1})$ as the sum of the number of retransmissions on the source link, the tail link and the initial transmission can be obtained according to Eq. 1. Group leaders receive an ACK from each child node for every data transmission provided that the data packet and ACK packet was not lost. The mean number of ACKs $E(\widetilde{L}^{H1})$ is therefore:

$$E(\widetilde{L}^{H1}) = BE(M^{H1})(1-q_D)(1-p_A), \quad (18)$$

where B is the branching factor of a group leader.

4.3.2. Receiver (Leaf Node)

$E(\widetilde{N}_{r,t}^{H1})$ is the total number of received transmissions at receiver r, which are mainly the messages sent by r's parent $E(\widetilde{N}_r^{H1})$, provided that each local group has its own multicast address. However, if the multicast group has only one multicast address as e.g. in TMTP [5], retransmissions may reach members outside of a local group. The scope overlapping probability for receiving

a retransmission from another local group is assumed to be p_l for any receiver. Such transmissions received from other local groups obviously increase the load of a node. In our analysis, we assume that transmissions from other local groups do not decrease the necessary number of local retransmissions, since in most cases they are received after a local retransmission has already been triggered.

The mean number of received transmissions $E(\widetilde{N}_r^{H1})$ from the parent node at receiver r is:

$$E(\widetilde{N}_r^{H1}) = E(M^{H1})(1-q_D). \tag{19}$$

The total number of received transmissions $E(\widetilde{N}_{r,t}^{H1})$ at receiver r is the sum of transmissions from r's parent plus those received from other local groups (for G see Eq. 12):

$$E(\widetilde{N}_{r,t}^{H1}) = E(M^{H1})(1-q_D) + (G-1)\Big(E(M^{H1})-1\Big)p_l. \tag{20}$$

Finally, the processing requirements P_R^{H1} and bandwidth requirements W_R^{H1} for a receiver are:

$$E(P_R^{H1}) = E(Y_f) + E(\widetilde{N}_{r,t}^{H1})E(Y_d) + E(\widetilde{N}_r^{H1})E(Y_a) \tag{21}$$

$$E(W_R^{H1}) = E(\widetilde{N}_{r,t}^{H1})E(W_d) + E(\widetilde{N}_r^{H1})E(W_a). \tag{22}$$

4.3.3. Group Leader (Inner Node)

Since a group leader is a sender and receiver as well, the processing requirement is basically the sum of the sender and receiver processing requirements. However, $X_d(1)$ and X_f are not considered here, since the initial transmission is sent using the multicast routing tree rather than the ACK tree and the group leader does not feed in a packet from a higher layer. Furthermore, a group leader may receive additional retransmissions only from $G-2$ other group leaders since this group leader and its parent group leader have to be subtracted.

$$E(\widetilde{N}_g^{H1}) = E(M^{H1})(1-q_D) + (G-2)\Big(E(M^{H1})-1\Big)p_l \tag{23}$$

$$\begin{aligned} E(P_G^{H1}) &= \Big(E(M^{H1})-1\Big)\Big(E(X_d)+E(X_t)\Big) \\ &\quad + E(\widetilde{L}^{H1})E(X_a) + E(\widetilde{N}_g^{H1})E(Y_d) \\ &\quad + E(\widetilde{N}_r^{H1})E(Y_a) + E(Y_f) \qquad (24) \\ &= E(P_S^{H1}) + E(P_R^{H1}) - E(X_f) - E(X_d(1)) \\ &\quad - \Big(E(M^{H1})-1\Big)p_l E(Y_d) \qquad (25) \end{aligned}$$

$$E(W_G^{H1}) = E(W_S^{H1}) + E(W_R^{H1}) - E(W_d(1)) - \Big(E(M^{H1})-1\Big)p_l E(W_d). \tag{26}$$

The maximum rates limited by processing requirements for the sender Λ_S^{H1}, receiver Λ_R^{H1} and group leader Λ_G^{H1} are:

$$\Lambda_S^{H1} = \frac{1}{E(P_S^{H1})},\ \Lambda_R^{H1} = \frac{1}{E(P_R^{H1})},\ \Lambda_G^{H1} = \frac{1}{E(P_G^{H1})}. \tag{27}$$

Analogous, the maximum rates limited by bandwidth requirements are:

$$\Lambda_S^{H1} = \frac{1}{E(W_S^{H1})},\ \Lambda_R^{H1} = \frac{1}{E(W_R^{H1})},\ \Lambda_G^{H1} = \frac{1}{E(W_G^{H1})}. \tag{28}$$

Overall system throughput Λ^{H1} is given by the minimum of the packet processing rates for the sender, receiver and group leader:

$$\Lambda^{H1} = min\{\Lambda_S^{H1}, \Lambda_G^{H1}, \Lambda_R^{H1}\}. \tag{29}$$

Our definition of total bandwidth consumption encompasses the total costs at the communication endpoints, i.e. the costs for the sender and receivers but not the internal network costs, i.e. costs for the routers and links. The total bandwidth consumption of protocol (H1) is the sum of the sender's, leaf node receivers' and group leaders' bandwidth consumption:

$$E(W^{H1}) = E(W_S^{H1}) + (R-G+1)E(W_R^{H1}) + (G-1)E(W_G^{H1}). \tag{30}$$

4.4. NAK-based Protocol (H2)

(H2) uses selective periodical ACKs (SAKs) and NAKs with NAK suppression scheme. We have to consider timer processing at the sender, other group leaders and receivers, since receivers have to detect NAK loss. Group leaders collect all NAKs belonging to one round and retransmit a message if the timer expires and at least one NAK has been received. We distinguish between the number of rounds and the number of transmissions. Due to NAK loss at the sender, it may happen that no retransmission occurs within a round. Then a further round is started until the sender receives at least one NAK and triggers a retransmission.

A SAK is sent by the receiver to announce its state, which specifies its received and missed packets. We assume that a SAK is sent after a certain number of packet transmissions. Therefore, when analyzing the requirements for a *single* packet, only the proportionate processing requirements for sending Y_Φ and receiving X_Φ a SAK are considered. W_Φ is the proportionate bandwidth requirement. S is assumed to be the number of SAKs received by the sender in the presence of SAK losses, where $E(S) = (1-p_A)B$.

4.4.1. Sender (Root Node)

The processing and bandwidth requirements are:

$$E(P_S^{H2}) = E(X_f) + E(M^{H2})E(X_d) + E(\widetilde{L}^{H2})E(X_n) + E(O^{H2})E(X_t) + E(S)E(X_\Phi) \tag{31}$$

$$E(W_S^{H2}) = E(M^{H2})E(W_d) + E(\widetilde{L}^{H2})E(W_n) + E(S)E(W_\Phi). \tag{32}$$

$E(M^{H2})$ is determined by Eq. 10 with loss probability $\tilde{p} = q_D\prime$. A round starts with the sending of a data packet and ends with the expiration of a timeout at the sender. Usually, there will be one data transmission in each round. However, if the sender receives no NAKs due to NAK losses, no retransmission is made and new NAKs must be sent by the receivers in the next round. O_r^{H2} is the number of rounds for receiver r. The number of rounds is the sum of the number of necessary rounds for sending transmissions M_r^{H2} and the number of empty rounds $O_{e,r}^{H2}$ in which all NAKs are lost and therefore no retransmission is made:

$$O_r^{H2} = M_r^{H2} + O_{e,r}^{H2}. \tag{33}$$

$E(M_r^{H2})$ is given in Eq. 4 with failure probability $q = q_D$. The expected number of empty rounds $E(O_{e,r}^{H2})$ is the expected number of empty rounds after the first transmission plus the expected number of empty rounds after the second transmission and so on:

$$E(O_{e,r}^{H2}) = \sum_{k=1}^{E(M_r^{H2})-1} \left(\frac{1}{1-p_k} - 1\right). \quad (34)$$

$(1/1-p_k)$ is the expectation for the number of empty rounds plus the last successful NAK reception at the sender, which is subtracted (see Eq. 4). The number of empty rounds after transmission k is determined by the failure probability p_k, i.e. the probability that all sent NAKs in round k are lost:

$$p_k = {q_N}^{N_k}. \quad (35)$$

N_k, the number of NAKs sent in round k, is obtained as follows. The first receiver that did not receive the data packet sends a NAK. The probability for packet loss in round k is ${q_D}^k$, which is equal to $N_{k,1}$, the probability for the first receiver to send a NAK. Then a second receiver sends a NAK provided that it has received no data packet and no NAK packet. Either the first receiver has sent no NAK (with probability $1-N_{k,1}$) or the NAK was lost or sent simultaneously (with probability $N_{k,1}(q_N + p_s - q_N p_s)$). As we assume a system model in which local clocks are not synchronized, it is possible that NAKs are sent simultaneously. This probability is given by p_s. Now, N_k can be expressed as follows:

$$N_k = \sum_{i=1}^{B} N_{k,i} \quad (36)$$

$$N_{k,1} = {q_D}^k \quad (37)$$

$$\begin{aligned} N_{k,2} &= {q_D}^k(1 - N_{k,1} + N_{k,1}(q_N + p_s - q_N p_s)) \\ &= N_{k,1} - N_{k,1}^2 + N_{k,1}^2(q_N + p_s - q_N p_s) \end{aligned} \quad (38)$$

$$\begin{aligned} N_{k,n} &= N_{k,n-1} - N_{k,n-1}^2 \\ &\quad + N_{k,n-1}^2(q_N + p_s - q_N p_s), n > 1. \end{aligned} \quad (39)$$

The total number of rounds O^{H2} for all receivers can be defined analogous to O_r^{H2}:

$$O^{H2} = M^{H2} + O_e^{H2} \quad (40)$$

$$E(O_e^{H2}) = \sum_{k=1}^{E(M^{H2})-1} \left(\frac{1}{1-p_k} - 1\right). \quad (41)$$

To determine $E(\widetilde{L}^{H2})$ we must take into account that NAKs are not only received from members of this local group but may also be received from other local groups with scope overlapping probability p_l (see Eq. 20):

$$E(\widetilde{L}^{H2}) = \vartheta_1(1 - q_N) + (G-1)\vartheta_1 p_l \quad (42)$$

$$\vartheta_1 = \sum_{k=1}^{E(M^{H2})} N_k \frac{1}{1-p_k}. \quad (43)$$

ϑ_1 is the total number of NAKs sent within a local group. The number of group leaders (G), is obtained with Eq. 12.

4.4.2. Receiver (Leaf Node)

Retransmissions are received mainly from its group leader, but may also be received from leaders of other local groups. Analogous, NAKs are mainly received from other receivers of this local group but may also be received from members of other local groups. The processing and bandwidth requirement for a receiver are:

$$\begin{aligned} E(P_R^{H2}) &= E(Y_f) + E(M^{H2})(1-q_D)E(Y_d) + E(Y_\Phi) \\ &\quad + [E(O_r^{H2}) - 1]\frac{\vartheta_2}{\vartheta_3}E(Y_n) + [E(O_r^{H2}) - 2]E(Y_t) \\ &\quad \underbrace{+\Big[[E(O^{H2}) - 1]\vartheta_2 - [E(O_r^{H2} - 1]\frac{\vartheta_2}{\vartheta_3}\Big](1-q_N)E(X_n)}_{\text{from this local group}} \\ &\quad +(G-1)p_l \\ &\quad \underbrace{\Big[\Big(E(M^{H2}) - 1\Big)E(Y_d) + [E(O^{H2}) - 1]\vartheta_2 E(X_n)\Big]}_{\text{from other local groups}} \end{aligned} \quad (44)$$

$$\begin{aligned} E(W_R^{H2}) &= E(M^{H2})(1-q_D)E(W_d) + E(W_\Phi) \\ &\quad + [E(O_r^{H2}) - 1]\frac{\vartheta_2}{\vartheta_3}E(W_n) \\ &\quad + \Big[[E(O^{H2}) - 1]\vartheta_2 - [E(O_r^{H2}) - 1]\frac{\vartheta_2}{\vartheta_3}\Big](1-q_N)E(W_n) \\ &\quad + (G-1)p_l \\ &\quad \Big[\Big(E(M^{H2}) - 1\Big)E(W_d) + [E(O^{H2}) - 1]\vartheta_2 E(W_n)\Big]. \end{aligned} \quad (45)$$

$(E(O^{H2}) - 1)$ is used as an abbreviation for $P(O^{H2} > 1)[E(O^{H2}|O^{H2} > 1) - 1]$ (see Eq. 6). Accordingly, $(E(O_r^{H2}) - 1)$ is also an analogous abbreviation. ϑ_2 is the average number of NAKs sent in each round and ϑ_3 is the mean number of receivers that did not receive a data packet and therefore are supposed to send a NAK:

$$\vartheta_2 = \frac{1}{E(O^{H2})} \sum_{k=1}^{E(M^{H2})} N_k \frac{1}{1-p_k} \quad (46)$$

$$\vartheta_3 = \frac{1}{E(O^{H2})} \sum_{k=1}^{E(M^{H2})} {q_D}^k B \frac{1}{1-p_k}, \quad (47)$$

where $(1/1-p_k)$ is the number of empty rounds plus the last successful NAK sent (see Eq. 4 and 43).

ϑ_2/ϑ_3 in $E(P_R^{H2})$ obtains the probability for the considered receiver r to be the one that sends a NAK. The term with X_n obtains the processing requirements to receive NAKs from other nodes. The number of sent NAKs is subtracted from the number of total NAKs to get the number of received NAKs.

4.4.3. Group Leader (Inner Node)

The group leader role contains both the sender and receiver role. The processing and bandwidth requirements for a group leader as well as the overall system throughput and bandwidth consumption can be obtained analogous to protocol (H1)

The analysis of protocol classes (H3) and (H4) is similar to the analysis of classes (H1) and (H2). The detailed formulas can be found in [12].

5. Numerical Results

In the following we will show the impact of the branching factor on the protocols' performance by means of numerical examples. For all results, the mean processing costs are set equal to 1, except for the periodic costs, which are set equal to 0.1. Also bandwidth costs for a data packet are set equal to 1. Since control packets are usually smaller, their costs are set to 0.1. Therefore,

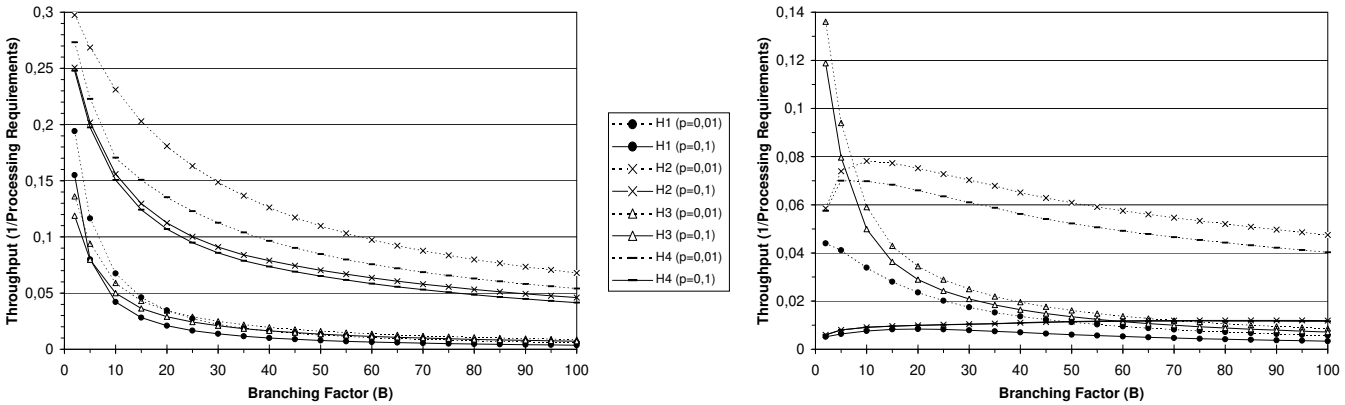

Figure 3. Throughput limited by processing requirements (a) scope overlapping $p_l = 0$ (left side) and (b) $p_l = 0.1$ (right side)

the periodic control packet costs are set equal to 0.01. With this costs, the graphs show the throughput of the various protocol classes relative to the normalized maximum throughput of 1. Data packet as well as control packet loss probability is set to 0.1 or 0.01. The dotted curves are the result for loss probability 0.01 and the solid ones for loss probability 0.1. (H3) is configured to use always unicast for retransmissions. All displayed results assume a group size of 10000 receivers. We have also evaluated the results for 1000 and 100000 receivers.

As there are no measurements from protocols in the Internet available, we can obtain a reasonable scope overlapping probability p_l only by simulations. Our used probability is obtained due to simulation results for TMTP [5] with group sizes of 25 to 100 nodes in networks of 1000 to 2000 nodes. Unfortunately, it was not possible to simulate a sparse multicast group, e.g. 100 receivers in a network of 100000 nodes, as the used simulator NS2 does not provide scalability for large networks. We have measured overlapping probabilities (p_l) between 0.2 and 0.6. We expect that for sparse groups, i.e. for large networks, the overlapping probability will be lower since in this case TTL scoping works more efficiently. Therefore, we have used $p_l = 0.1$ for the numerical results.

Figure 3 shows the throughput of all analyzed protocol classes with respect to the processing requirements. In Figure 3.a it is assumed that local groups are perfectly confined, i.e. messages sent by a group leader are only received by the leader's local group. This can be achieved by assigning a multicast address for each local group. As shown in this figure, small local groups reach the highest throughput with respect to processing requirements. The reason for this result is that less packets must be sent or received at a single inner node if the local group size is small. Although not depicted in the figure, a group size of 1 would reach the best results. However, such a local group size is not reasonable for real world protocol implementations since this would result in large path lengths and therefore high delays within the ACK tree.

In Figure 3.b it is assumed that local groups are not perfectly confined with a scope overlapping probability of $p_l = 0.1$. As the results show, this assumption leads to larger optimal group sizes for most protocols. However, (H1)'s optimal branching factor with loss probability 0.01 is still two child nodes per group leader. As protocols (H2) and (H4) send not only retransmissions by means of multicast but also NAKs, more messages are received outside the scope of a local group. So, they react more sensitive to not perfectly confined local groups than (H1) and therefore, a larger branching factor and a smaller number of local groups provide better performance.

If the scope overlapping probability p_l is increased, the optimal branching factor increases also for all protocol classes. For example, with $p_l = 0.4$, the optimal branching factor for (H1) with loss probability 0.01 is then 5-10 and for (H2) 30 child nodes per group leader. The more local groups exist, the more independent message retransmissions are triggered. If local groups are not perfectly confined in scope, the number of local groups determine the number of received messages from other local groups. Because if more local groups exist, more message retransmissions are triggered and more messages are received outside the scope of the local group. This results in less local groups for maximum throughput and therefore in a larger optimal branching factor. If the scope overlapping probability p_l is decreased, the optimal branching factor decreases also. In the extreme case of $p_l = 0$, the optimal branching factor is two for all protocols as Figure 3.a shows.

The performance of protocol (H3) is independent of the scope overlapping probability always constant, since it is configured to use unicast for retransmissions. If the scope overlapping probability p_l exceeds 0.02, (H3) outperforms all other protocol classes.

The results for other group sizes show similar behaviour but differ in the exact quantity of the branching factor. Generally speaking, the more receivers in the multicast group are, the larger is the optimal branching factor. For example with $p_l = 0.1$ and 1000 receivers the

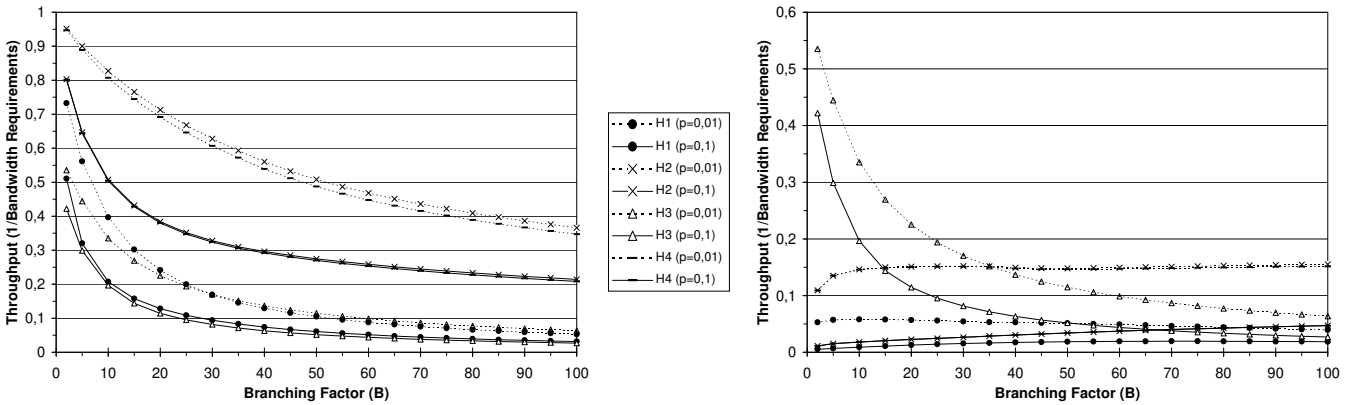

Figure 4. Throughput limited by bandwidth requirements with scope overlapping (a) $p_l = 0$ (left side) and (b) $p_l = 0.1$ (right side)

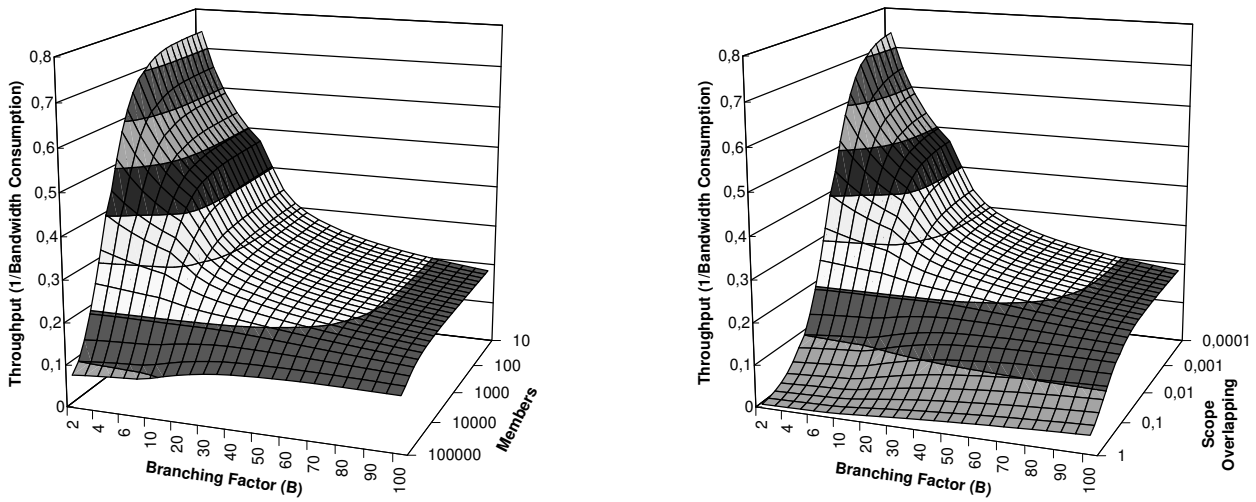

Figure 5. Bandwidth limited throughput of protocol class (H2) (a) scope overlapping $p_l = 0.001$ (left side) and (b) number of receivers $R = 1000$ (right side)

optimal branching factor for protocol (H2) with respect to processing requirements is 5 whereas with 100000 receivers it is 80.

Figure 4 shows the throughput with respect to bandwidth requirements. The results are similar to Figure 3, i.e. a low scope overlapping probability results in a small optimal branching factor whereas a high scope overlapping probability results in a larger optimal branching factor. By comparing Figure 3.b and Figure 4.b we can see, that the optimal branching factor with respect to bandwidth requirements is larger than with respect to processing requirements, since in the latter case also timeout processing is considered, which is independent of the scope overlapping probability.

In Figure 5 the throughput of protocol class (H2) with respect to bandwidth requirements is shown with varying branching factor, number of members of the multicast group and scope overlapping probability. Figure a) assumes a scope overlapping probability of $p_l = 0.001$ and Figure b) assumes 1000 receivers. In both figures the packet loss probability is 0.1. As Figure 5.a shows, the optimal branching factor depends also on the number of group members. With a small number of group members, the optimal branching factor is also small. However, with the given scope overlapping probability and a large number of group members, a larger branching factor results in a higher throughput. Figure 5.b shows the dependence on the scope overlapping probability in more detail. If a higher scope overlapping probability is given, the branching factor should be also large for optimal performance.

Figure 6 shows the total bandwidth consumption of all analyzed protocols in terms of weighted sent and received messages. The results for total bandwidth consumption are similar to the throughput results. With perfectly confined local groups, small local groups result in the lowest bandwidth consumption. In case of imperfectly confined local groups, larger local group sizes are preferable. In contrast to the throughput results, we cannot identify in Figure 6.b an optimal value

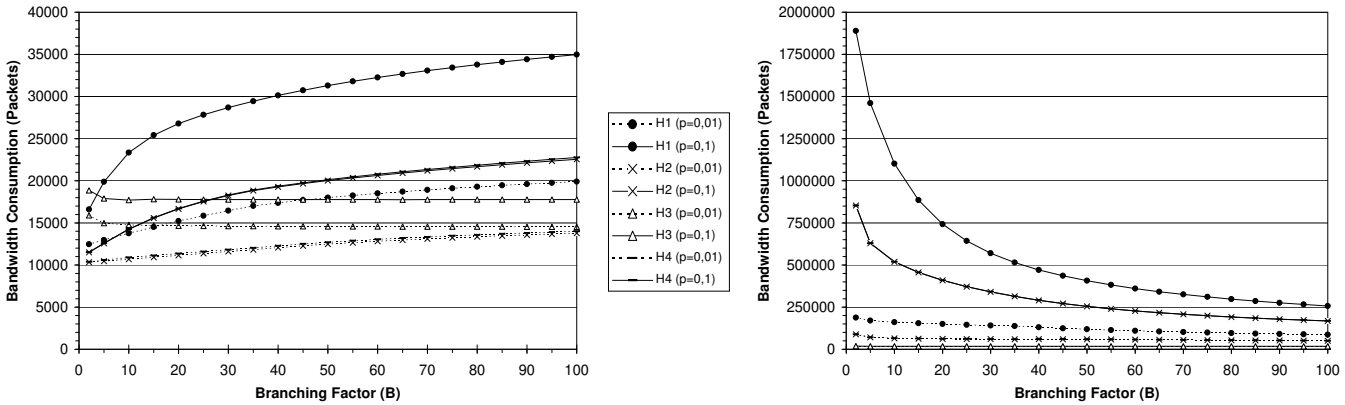

Figure 6. Bandwidth consumption with scope overlapping (a) $p_l = 0$ (left side) and (b) $p_l = 0.1$ (right side)

within the displayed range of up to 100 child nodes per group leader. In fact, total bandwidth consumption reacts very sensitive to imperfectly confined local groups, hence the optimal group size is larger than 100 nodes. However, we can see for loss probability 0.1 that after an initial decrease, the bandwidth consumption does not decrease significantly as the branching factor is increased. So, a branching factor of 30 or more child nodes would be a reasonable value in this scenario.

In [12] NS2 simulation results of TMTP are shown that prove the correctness of the analysis.

6. Summary

We have analyzed the processing and bandwidth requirements of reliable tree-based multicast protocols. Our work allows to determine the maximum throughput rates and bandwidth consumption with respect to the branching factor. The assumed system model considers data and control packet loss, asynchronous local clocks and local groups that are not perfectly confined in scope.

The numerical evaluations have shown the impact of the branching factor on the protocols' throughput and bandwidth consumption. The most important parameter is the probability for receiving messages from other local groups. If local groups are assigned to a separate multicast address and therefore messages are strictly confined to a local group, the optimal branching factor is two. On the other hand, if TTL scoping is used it can be assumed that messages are not strictly confined to the local group's scope. In this case, larger local groups provide better performance and less bandwidth consumption for most protocols.

Our future work will be to analyze the impact of the branching factor on end-to-end delay. A small branching factor leads to large path lengths within the ACK tree. It would be interesting to analyze whether this results in higher retransmission delays.

REFERENCES

1. B. Levine and J. Garcia-Luna-Aceves, "A comparison of reliable multicast protocols," *Multimedia Systems*, vol. 6, no. 5, pp. 334–348, Sept. 1998.
2. C. Maihöfer, K. Rothermel, and N. Mantei, "A throughput analysis of reliable multicast transport protocols," in *Proceedings of the Ninth International Conference on Computer Communications and Networks*, Las Vegas, USA, Oct. 2000, pp. 250–257, IEEE Press.
3. C. Maihöfer, "A bandwidth analysis of reliable multicast transport protocols," in *Proceedings of the Second International Workshop on Networked Group Communication (NGC 2000)*, Palo Alto, USA, Nov. 2000, pp. 15–26, ACM Press.
4. C. Maihöfer and K. Rothermel, "A delay analysis of generic multicast transport protocols," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2001)*, Tokyo, Japan, Aug. 2001, IEEE Press.
5. R. Yavatkar, J. Griffioen, and M. Sudan, "A reliable dissemination protocol for interactive collaborative applications," in *The Third ACM International Multimedia Conference and Exhibition (MULTIMEDIA '95)*, New York, USA, Nov. 1996, pp. 333–344, ACM Press.
6. K. Rothermel and C. Maihöfer, "A robust and efficient mechanism for constructing multicast acknowledgment trees," in *Proceedings of the Eight International Conference on Computer Communications and Networks*, Boston, USA, Oct. 1999, pp. 139–145, IEEE Press.
7. S. Pingali, D. Towsley, and J. F. Kurose, "A comparison of sender-initiated and receiver-initiated reliable multicast protocols," in *Proceedings of the Sigmetrics Conference on Measurement and Modeling of Computer Systems*, New York, USA, May 1994, pp. 221–230, ACM Press.
8. S. Kasera, J. Kurose, and D. Towsley, "A comparison of server-based and receiver-based local recovery approaches for scalable reliable multicast," in *Proceedings of IEEE INFOCOM Conference on Computer Communications*, New York, USA, Apr. 1998, pp. 988–995, IEEE Press.
9. J. Nonnenmacher, M. Lacher, M. Jung, G. Carl, and E. Biersack, "How bad is reliable multicast without local recovery," in *Proceedings of IEEE INFOCOM Conference on Computer Communications*, New York, USA, Apr. 1998, pp. 972–979, IEEE Press.
10. G. Poo and A. Goscinski, "Performance comparison of sender-based and receiver-based reliable multicast protocols," *Computer Communications*, vol. 21, no. 7, pp. 597–605, June 1998.
11. S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," *IEEE Journal on Selected Areas in Communications, special issue on Network Support for Multipoint Communication*, vol. 15, no. 3, pp. 407–421, Apr. 1997.
12. C. Maihöfer and K. Rothermel, "Optimal branching factor for tree-based reliable multicast protocols," in *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2001)*, Orlando, USA, July 2001, pp. 10–21, SCS.
13. M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the mbone multicast network," in *Proceedings of IEEE Global Internet*, London, UK, Nov. 1996, pp. 94–99, IEEE Press.