Optimal Branching Factor for Tree-based Reliable Multicast Protocols

Christian Maihöfer and Kurt Rothermel
University of Stuttgart, Institute of Parallel and Distributed High Performance Systems (IPVR),
Breitwiesenstr. 20-22, D-70565 Stuttgart, Germany
{christian.maihoefer|kurt.rothermel}@informatik.uni-stuttgart.de

Keywords — reliable multicast, throughput analysis, bandwidth analysis, TTL scoping

Abstract —In recent years, a number of 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 2. 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.

I. 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 [1] that tree-based multicast protocols scale better than other multicast schemes suggested in the literature. 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 ACK or NAK 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.

We will use standard tree terminology throughout the paper. A node 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) [2] or the token repository service [3], [4]. Whenever a new member wants to join a multicast group, a node in the corresponding ACK tree has to be selected to become the group leader of the new member. Both, ERS and the token repository service share the same goal, namely to find a group leader whose number of children is less than the tree's branching factor and that is as close as possible to the new member in terms of network distance. Consequently, the members of a local group are local in the sense that they are close to their group leader. 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.

The scope of retransmission messages should be confined to local groups, i.e., 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. 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. That is why existing protocols like TMTP [2] use TTL scoping for multicasting retransmissions. Therefore, the problem exists that retransmissions may be also 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 our analysis. 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 mainly depends 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 classi-

fication 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. In Section 6 we confirm our analytical results by simulation studies. Finally, we conclude our work with a brief summary.

II. Related Work

Reliable multicast protocols were already analyzed in previous work. The first work in this area was presented by Pingali et al. [5]. 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. [6], Nonnenmacher et al. [7] and Poo et al. [8]. In [6], local recovery techniques are analyzed and compared. The system model is based on a special topology structure consisting of a source link from the sender to the backbone, backbone links and finally tail links from the backbone to the receivers. In Nonnenmacher et al. [7] a similar topology structure is used. They studied the performance gain of protocols using parity packets to recover from transmission errors. The protocols use receiver-based loss detection with multicasted NAKs and NAK suppression. In [8], non-hierarchical protocols are compared. In contrast to previous work, not only stop-and-wait error recovery is considered in the analysis but also go-back-N and selective-repeat schemes.

Our paper extends previous work in the following ways. First, we consider the loss of control packets rather than assuming reliable delivery. In previous work, control packets are assumed to be reliably delivered, which especially favors protocols with multicast NAKs and NAK suppression. NAK suppression works most efficiently if no NAKs are lost at receivers and therefore, only one NAK per lost data packet is sufficient. Second, we assume that local clocks are not synchronized. Again, the NAK suppression scheme is influenced by this assumption since it works less efficiently if local clocks are not synchronized and therefore simultaneous NAKs are sent. Third, our work extends previous analysis by two new protocol classes based on aggregated ACKs. Aggregated ACKs are necessary to guarantee reliable delivery even in case of node failures. Fourth, in contrast to our previous work in this area [9], [10], we present simulations to confirm the analytical results. Finally, this work is the first one that focuses on the branching factor.

III. Classification of Tree-based Multicast Protocols

A. ACK-based Protocol (H1)

The first considered scheme is denoted as (H1). 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 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].

B. 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 [2] is an example for class (H2).

C. ACK and AAK-based Protocol (H3)

Before the next scheme will be introduced, it is necessary to understand that (H1) and (H2) can guarantee reliable delivery only if no group member fails in the system. Assume for example that a group leader G_1 fails after it has acknowledged correct reception of a packet to its group leader G_0 which is the root node. If a receiver of G_1's local group needs a retransmission, neither G_1 nor G_0 can resend the data packet since G_1 has failed and G_0 has removed the packet from memory. This problem is solved by aggregated hierarchical ACKs (AAKs) of the third scheme (H3). A group leader sends an AAK to its parent after all children have acknowledged correct reception. After a group leader has received an AAK, it can remove the corresponding data from memory because all members in this subhierarchy (i.e. the transitive closure of the child relation) have already received it correctly. RMTP II is an example for a protocol that uses AAKs [12]. Our definition of its generic behaviour is as follows:

1. A group leaders sends an ACK to its parent after a data packet has been received correctly.
2. A leaf node receiver sends an AAK to its parent after a data packet has been received correctly.
3. Group leaders wait a certain time to receive ACKs from all children. If a timeout occurs, the packet is retransmitted to all children or selective to those whose ACK is missing. Since leaf node receivers send only AAKs rather than ACKs, a received AAK is also allowed to prevent the retransmission.
4. Group leaders wait to receive AAKs from their children. Upon reception of all AAKs, the corresponding packet can be removed from memory and a group leader sends an AAK to its parent. If a timeout occurs while waiting, a unicast AAK query is sent to the affected nodes.
5. If retransmissions or AAK queries are received by a node after an AAK has been sent or the prerequisites for sending an AAK are met, an AAK is sent to the parent.

Besides AAKs, we consider in our analysis of (H3) a threshold scheme to decide whether a retransmission is performed using unicast or multicast. The group leader compares the number of missing ACKs with a threshold parameter. If the number of missing ACKs is smaller than this threshold, the data packets are retransmitted using unicast. Otherwise, if the number of missing ACKs exceeds the threshold, the overall network and node load is assumed to be lower using multicast retransmission.

D. NAK and AAK-based Protocol (H4)

Our next protocol will be denoted as (H4) and is a combination of the negative acknowledgment with NAK suppression scheme (H2) and aggregated acknowledgments. Similar to (H2), NAKs are used to start a retransmission. Instead of selective periodical ACKs, aggregated ACKs are used to announce the receivers' state and allow group leaders to remove data from memory. Like SAKs, we assume that AAKs are sent periodically. We define the generic behaviour of (H4) as follows:

1. Upon detection of a missing or corrupted data packet, receivers send a NAK to the local group by means of multicast scheduled at a random time in the future and provided that not already a NAK for this data packet is received before the scheduled time. If no retransmission arrives within a certain time period, the NAK sending scheme is repeated.
2. Group leaders retransmit a packet to the local group by means of multicast if a NAK has been received.
3. After a certain number of correctly received data packets, leaf node receivers send an AAK to their group leader in the ACK tree. A group leader forwards an AAK to its parent as soon as the data packets are correctly received and the corresponding AAKs from all child nodes have been received.
4. Group leaders initiate a timer and wait for all AAKs to be received. If the timer expires, an AAK query is sent to those child nodes whose AAK is missing.
5. If an AAK query is received by a node and the prerequisites for sending an AAK are met, the query is acknowledged with an AAK.

IV. Analysis

A. Model

Our model is similar to the one used by Pingali et al. [5] 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 assumed to be eventually successful. In contrast to previous work, we assume that packet loss can occur on both, data packets *and* control packets. The multicast packet loss probability is given by q and unicast packet loss probability by p. All parameters used in our analysis are summarized in Table 1 and Table 2. Table 2 lists the additional notations for the protocol classes (H3) and (H4) with aggregated ACKs.

We assume further that losses at different nodes are independent events. In fact, since receivers share parts of the multicast routing tree, this does not hold in real networks. However, this widespread assumption in most analytical work is made to keep the analysis simple. In [13] and [14] the temporal and spatial loss correlation in the Internet and MBone is studied in detail. They concluded from measurements that the timescale for temporal loss correlation is 1 second or less. Beyond this timescale, what happens to a packet is not connected to the behaviour of a former sent packet. Even within the correlation timescale, most losses were solitary losses. With respect to spatial losses, they found only small correlation among the multicast sites except for the loss due to the link next to the source.

We can conclude from these observations that assuming temporal independent losses introduces only a negligible inaccuracy into our model. With respect to spatial losses, an inaccuracy we introduce is the spatial correlation due to loss on the first link from the sender to the backbone. One advantage of our assumption is that the provided results are independent of a concrete network structure and therefore applicable for local networks as well as for global networks like the Internet. We relax this assumption by confirming the analytical results with simulation studies in Section VI.

B. Protocol Independent Methods

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.

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. M_r depends on the probability $\tilde{p}$ that a retransmission is necessary. $\tilde{p}$ 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_r for receiver r is smaller or equal to m (m=1, 2, ...) is:

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

As the packet losses at different receivers are assumed to be independent from each other, the following holds [5]:

$$P(M \leq m) = \prod_{r=1}^{B} P(M_r \leq m) = (1-\tilde{p}^m)^B = \sum_{i=0}^{B} \binom{B}{i} (-1)^i \tilde{p}^{im} \tag{2}$$

$$P(M = m) = P(M \leq m) - P(M \leq m-1)$$

$$E(M) = \sum_{m=1}^{\infty} mP(M=m) = \sum_{i=1}^{B} \binom{B}{i} (-1)^{i+1} \frac{1}{1-\tilde{p}^i}. \tag{3}$$

$E(M)$ is the expected number of necessary transmissions to receive the data packet correctly at all receivers.

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 $(1-\bar{p})$ is:

$$P(X = k) = (1-\bar{p})\bar{p}^{k-1}. \tag{4}$$

The expectation follows to (see [5]):

$$E(M_r) = \frac{1}{1-\bar{p}} \tag{5}$$

$$E(M_r | M_r > x) = \frac{x+1-x\bar{p}}{1-\bar{p}} \tag{6}$$

$$P(M_r > x)[E(M_r | M_r > x) - x] = E(M_r) - x. \tag{7}$$

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:

$$R = \sum_{i=0}^{h-1} B^i = \frac{1-B^h}{1-B} \Rightarrow h = \log_B\Big(R(B-1)+1\Big). \tag{8}$$

G, the number of group leaders follows to:

$$G = \textstyle\sum_{i=0}^{\log_B[R(B-1)+1]-2} B^i. \tag{9}$$

Table 1
NOTATIONS FOR THE ANALYSIS

R	Size of the receiver set.
B	Branching factor of a tree, i.e. the local group size.
X_f, Y_f	Time to feed in a new data packet from a higher protocol layer or to deliver one to a higher layer.
X_d, X_a, X_n, X_Φ	Time to process the data transmission or time to receive and process an ACK, NAK or periodic SAK.
Y_d, Y_a, Y_n, Y_Φ	Time to receive and process a data packet or to process and transmit an ACK, NAK or periodic SAK.
X_t, Y_t	Time to process a timeout at the sender or receiver.
W_d, W_a, W_n, W_ϕ	Bandwidth for a data packet, ACK, NAK and periodic SAK, respectively.
S	Number of periodical SAKs received by the sender in the presence of control message loss.
p_D, q_D	Probability for unicast or multicast data loss at a receiver, respectively.
p_A, p_N	Probability for unicast ACK or NAK loss at a sender.
q_N	Probability for multicast NAK loss at a sender or receiver.
p_s	Probability for simultaneous and therefore unnecessary NAK sending in (H2) and (H4).
p_l	Scope overlapping probability, i.e. probability for receiving a packet from another local group.
$\widetilde{L}_r^w$	Number of ACKs/NAKs per data packet sent by receiver r that reach the sender. $w \in \{H1, H2, H3, H4\}$
$\widetilde{L}^w$	Total number of ACKs or NAKs per data packet received by the sender from all receivers.
$\widetilde{N}_r^w$, $\widetilde{N}_{r,t}^w$, $\widetilde{N}_g^w$	Total number of transmissions per data packet received by receiver r from the parent or total number of received transmissions from all local groups or total number of received transmissions at group leader g.
M_r^w, M^w	Number of necessary transmissions for receiver r and total number of transmission for all receivers.
O_r^w, O^w	Number of necessary rounds to correctly deliver a packet to receiver r or to all receivers, respectively.
$O_{e,r}^w$, O_e^w	Total number of empty rounds for receiver r or empty rounds for all receivers, respectively.
N_k	Number of NAKs sent in round k.
P_S^w, P_R^w, P_G^w	Processing time per packet at sender, receiver, or group leader respectively.
Λ_S^w, Λ_R^w, Λ_G^w, Λ^w	Throughput for protocols w at the sender, receiver, group leader and overall system throughput.
W_S^w, W_R^w, W_G^w, W^w	Bandwidth requirements for protocols w at the sender, receiver, group leader and overall bandwidth consumption.

C. 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.

C.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 1.

At the sender we have:

$$P_S^{H1} = X_f + X_d(1) + \sum_{m=2}^{M^{H1}} \Big(X_t(m-1) + X_d(m)\Big) + \sum_{i=1}^{\widetilde{L}^{H1}} X_a(i). \quad (10)$$

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:

$$E(P_S^{H1}) = E(X_f) + E(M^{H1})E(X_d) + \Big(E(M^{H1}) - 1\Big)E(X_t) + E(\widetilde{L}^{H1})E(X_a). \quad (11)$$

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

$$E(W_S^{H1}) = E(M^{H1})E(W_d) + E(\widetilde{L}^{H1})E(W_a), \quad (12)$$

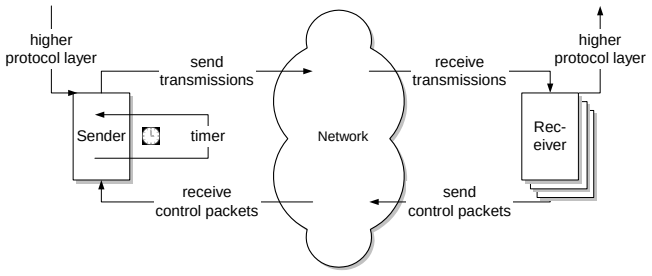

Fig. 1. CPU processing load

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 $E(\tilde{L}^{H1})$. $E(M^{H1})$, the expected number of necessary transmissions, is determined by the probability for a retransmission:

$$\tilde{p} = q_D + (1 - q_D)p_A, \tag{13}$$

i. e. either a data packet is lost (q_D) or the data packet is received correctly and the ACK is lost ($(1 - q_D)p_A$). Now, $E(M^{H1})$ can be determined with Eq. 3:

$$E(M^{H1}) = \sum_{i=1}^{B} \binom{B}{i} (-1)^{i+1} \frac{1}{1 - \tilde{p}^i}. \tag{14}$$

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 $E(\tilde{L}^{H1})$ is therefore:

$$E(\tilde{L}^{H1}) = BE(M^{H1})(1 - q_D)(1 - p_A). \tag{15}$$

C.2 Receiver (Leaf Node)

$E(\tilde{N}_{r,t}^{H1})$ is the total number of received transmissions at receiver r, which are mainly the messages sent by r's parent $E(\tilde{N}_r^{H1})$, 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 [2], 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 $E(\tilde{N}_r^{H1})$ from the parent node at receiver r is:

$$E(\tilde{N}_r^{H1}) = E(M^{H1})(1 - q_D). \tag{16}$$

The total number of received transmissions $E(\tilde{N}_{r,t}^{H1})$ at receiver r is the sum of transmissions from r's parent plus those received from other local groups (for G see Eq. 9):

$$E(\tilde{N}_{r,t}^{H1}) = E(M^{H1})(1 - q_D) + (G - 1)\Big(E(M^{H1}) - 1\Big)p_l. \tag{17}$$

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

$$E(P_R^{H1}) = E(Y_f) + E(\tilde{N}_{r,t}^{H1})E(Y_d) + E(\tilde{N}_r^{H1})E(Y_a) \tag{18}$$

$$E(W_R^{H1}) = E(\tilde{N}_{r,t}^{H1})E(W_d) + E(\tilde{N}_r^{H1})E(W_a). \tag{19}$$

C.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.

$$E(\tilde{N}_g^{H1}) = E(M^{H1})(1 - q_D) + (G - 2)\Big(E(M^{H1}) - 1\Big)p_l \tag{20}$$

$$\begin{aligned} E(P_G^{H1}) &= \Big(E(M^{H1}) - 1\Big)\Big(E(X_d) + E(X_l)\Big) + E(\tilde{L}^{H1})E(X_a) \\ &\quad + E(\tilde{N}_g^{H1})E(Y_d) + E(\tilde{N}_r^{H1})E(Y_a) + E(Y_f) \qquad (21) \\ &= E(P_S^{H1}) + E(P_R^{H1}) - E(X_f) - E(X_d(1)) \\ &\quad -\Big(E(M^{H1}) - 1\Big)p_l E(Y_d) \qquad (22) \end{aligned}$$

$$\begin{aligned} E(W_G^{H1}) &= E(W_S^{H1}) + E(W_R^{H1}) \\ &\quad - E(W_d(1)) - \Big(E(M^{H1}) - 1\Big)p_l E(W_d). \qquad (23) \end{aligned}$$

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

$$\Lambda_S^{H1} = \frac{1}{E(P_S^{H1})}, \quad \Lambda_R^{H1} = \frac{1}{E(P_R^{H1})}, \quad \Lambda_G^{H1} = \frac{1}{E(P_G^{H1})}. \tag{24}$$

Analogous, the maximum rates limited by bandwidth requirements are:

$$\Lambda_S^{H1} = \frac{1}{E(W_S^{H1})}, \quad \Lambda_R^{H1} = \frac{1}{E(W_R^{H1})}, \quad \Lambda_G^{H1} = \frac{1}{E(W_G^{H1})}. \tag{25}$$

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

$$\Lambda^{H1} = min\{\Lambda_S^{H1}, \Lambda_G^{H1}, \Lambda_R^{H1}\}. \tag{26}$$

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:

$$\begin{aligned} E(W^{H1}) &= E(W_S^{H1}) + (R - G + 1)E(W_R^{H1}) \\ &\quad + (G - 1)E(W_G^{H1}). \qquad (27) \end{aligned}$$

D. NAK-based Protocol (H2)

(H2) uses selective periodical ACKs (SAKs) and NAKs with NAK suppression scheme. 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 $E(S) = (1 - p_A)B$.

D.1 Sender (Root Node)

The processing and bandwidth requirements are:

$$E(P_S^{H2}) = E(X_f) + E(M^{H2})E(X_d) + E(\tilde{L}^{H2})E(X_n) + E(O^{H2})E(X_t) + E(S)E(X_\Phi) \quad (28)$$

$$E(W_S^{H2}) = E(M^{H2})E(W_d) + E(\tilde{L}^{H2})E(W_n) + E(S)E(W_\Phi). \quad (29)$$

$E(M^{H2})$ is determined by Eq. 3 with loss probability $\tilde{p} = q_D$. 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:

$$O_r^{H2} = M_r^{H2} + O_{e,r}^{H2}. \quad (30)$$

$E(M_r^{H2})$ is given in Eq. 5 with failure probability $\bar{p} = q_D$. 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:

$$E(O_{e,r}^{H2}) = \sum_{k=1}^{E(M_r^{H2})-1} \left(\frac{1}{1-p_k} - 1\right). \quad (31)$$

$(1/1 - p_k)$ is the expectation for the number of empty rounds plus the last successful NAK reception at the sender, which is subtracted (see Eq. 5). 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:

$$p_k = q_N^{N_k}. \quad (32)$$

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:

$$N_k = \sum_{i=1}^{B} N_{k,i} \quad (33)$$

$$N_{k,1} = {q_D}^k \quad (34)$$

$$N_{k,2} = {q_D}^k(1 - N_{k,1} + N_{k,1}(q_N + p_s - q_N p_s)) = N_{k,1} - N_{k,1}^2 + N_{k,1}^2(q_N + p_s - q_N p_s) \quad (35)$$

$$N_{k,n} = N_{k,n-1} - N_{k,n-1}^2 + N_{k,n-1}^2(q_N + p_s - q_N p_s), n > 1. \quad (36)$$

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

$$O^{H2} = M^{H2} + O_e^{H2} \quad (37)$$

$$E(O_e^{H2}) = \sum_{k=1}^{E(M^{H2})-1} \left(\frac{1}{1-p_k} - 1\right). \quad (38)$$

To determine $E(\tilde{L}^{H2})$ 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. 17):

$$E(\tilde{L}^{H2}) = \vartheta_1(1 - q_N) + (G - 1)\vartheta_1 p_l \quad (39)$$

$$\vartheta_1 = \sum_{k=1}^{E(M^{H2})} N_k \frac{1}{1-p_k}. \quad (40)$$

ϑ_1 is the total number of NAKs sent within a local group. The number of group leaders (G), is obtained with Eq. 9.

D.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:

$$\begin{aligned} E(P_R^{H2}) &= E(Y_f) + E(M^{H2})(1 - q_D)E(Y_d) + E(Y_\Phi) \\ &+ [E(O_r^{H2}) - 1]\frac{\vartheta_2}{\vartheta_3}E(Y_n) + [E(O_r^{H2}) - 2]E(Y_t) \\ &\underbrace{+\left[[E(O^{H2}) - 1]\vartheta_2 - [E(O_r^{H2} - 1]\frac{\vartheta_2}{\vartheta_3}\right](1 - q_N)E(X_n)}_{\text{from this local group}} \\ &+(G - 1)p_l \\ &\underbrace{\left[\left(E(M^{H2}) - 1\right)E(Y_d) + [E(O^{H2}) - 1]\vartheta_2 E(X_n)\right]}_{\text{from other local groups}} \end{aligned} \quad (41)$$

$$\begin{aligned} E(W_R^{H2}) &= E(M^{H2})(1 - q_D)E(W_d) + E(W_\Phi) \\ &+ [E(O_r^{H2}) - 1]\frac{\vartheta_2}{\vartheta_3}E(W_n) \\ &+ \left[[E(O^{H2}) - 1]\vartheta_2 - [E(O_r^{H2}) - 1]\frac{\vartheta_2}{\vartheta_3}\right](1 - q_N)E(W_n) \\ &+ (G - 1)p_l \\ &\left[\left(E(M^{H2}) - 1\right)E(W_d) + [E(O^{H2}) - 1]\vartheta_2 E(W_n)\right]. \end{aligned} \quad (42)$$

$(E(O^{H2}) - 1)$ is used as an abbreviation for $P(O^{H2} > 1)[E(O^{H2}|O^{H2} > 1) - 1]$ (see Eq. 7). 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:

$$\vartheta_2 = \frac{1}{E(O^{H2})}\sum_{k=1}^{E(M^{H2})} N_k \frac{1}{1-p_k} \quad (43)$$

$$\vartheta_3 = \frac{1}{E(O^{H2})}\sum_{k=1}^{E(M^{H2})} {q_D}^k B \frac{1}{1-p_k}, \quad (44)$$

where $(1/1 - p_k)$ is the number of empty rounds plus the last successful NAK sent (see Eq. 5 and 40).

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.

D.3 Group Leader (Inner Node)

As the group leader role contains the sender and receiver role as well, the processing and bandwidth requirements are:

$$\begin{aligned} E(P_G^{H2}) &= E(P_S^{H2}) + E(P_R^{H2}) - E(X_f) - E(X_d(1)) \\ &\quad - p_l\Big[\Big(E(M^{H2}) - 1\Big)E(Y_d) + [E(O^{H2}) - 1]\vartheta_2 E(Y_n)\Big] \qquad (45) \\ E(W_G^{H2}) &= E(W_S^{H2}) + E(W_R^{H2}) - E(W_d(1)) \\ &\quad - p_l\Big[\Big(E(M^{H2}) - 1\Big)E(W_d) + [E(O^{H2}) - 1]\vartheta_2 E(W_n)\Big]. \qquad (46) \end{aligned}$$

In the above equations the processing requirements for one other local group are subtracted, which results in $G-2$ other group leaders (see protocol (H1)). The rate for sender, receiver and group leader as well as the overall system throughput and bandwidth consumption for (H2) can be obtained analogous to (H1).

E. ACK and AAK-based Protocol (H3)

We assume that the correct transmission of a data packet consists of two phases. In the first phase, the data is transmitted and ACKs are collected until all ACKs are received. Then the second phase starts, in which the missing AAKs are collected. Note that most AAKs are already received in phase one, since AAKs are sent from group leaders as soon as all children have sent their AAKs (see Section III). So, only nodes whose AAK is missing must be queried in phase two.

E.1 Sender (Root Node)

The additional notations for the analysis of (H3) and (H4) are given in Table 2. The processing and bandwidth requirements are:

$$\begin{aligned} E(P_S^{H3}) &= E(X_f) + E(M_u^{H3})N_u E(X_{d,u}) + E(M_m^{H3})E(X_{d,m}) \\ &\quad + E(M^{H3} - 1)E(X_t) \\ &\quad + E(\tilde{L}_a^{H3})E(X_a) + E(O_q^{H3})E(X_t) \\ &\quad + E(L_{aaq}^{H3})E(X_{aaq}) + E(\tilde{L}_{aa}^{H3})E(X_{aa}) \qquad (47) \\ E(W_S^{H3}) &= E(M_u^{H3})N_u E(W_{d,u}) + E(M_m^{H3})E(W_{d,m}) \\ &\quad + E(\tilde{L}_a^{H3})E(W_a) + E(L_{aaq}^{H3})E(W_{aaq}) \\ &\quad + E(\tilde{L}_{aa}^{H3})E(W_{aa}). \qquad (48) \end{aligned}$$

M_m^{H3} and M_u^{H3} are the number of necessary multicast or unicast transmissions, respectively. M^{H3} is the total number of transmissions. $X_{d,m}$ and $X_{d,u}$ determine the processing requirements and $W_{d,m}$ and $W_{d,u}$ the bandwidth requirements for a multicast or unicast packet transmission. $\tilde{L}_{aa}^{H3}$ is the number of received AAKs. The processing of AAKs is similar to the processing of data packets and ACKs. If AAKs are missing after a timeout has occurred, the sender or group leader sends unicast AAK query messages to the corresponding child nodes. Note that this processing is started after all ACKs have been received and no further retransmissions due to lost data packets are necessary. O_q^{H3} is the number of necessary query rounds and L_{aaq}^{H3} is the number of necessary unicast AAK queries in the presence of message loss.

With p_t, the probability that unicast is used for retransmissions, the number of unicast and multicast transmissions are:

$$M_u^{H3} = p_t(M^{H3} - 1) \qquad (49)$$

$$M_m^{H3} = (1 - p_t)(M^{H3} - 1) + 1. \qquad (50)$$

Please note that the first transmission is always sent with multicast. $E(M^{H3})$ is determined with Eq. 3 and probability $\tilde{p}$ for a retransmission due to data or ACK loss:

$$\tilde{p} = \underbrace{p_t p_D + (1 - p_t)q_D}_{\text{data loss}} + \underbrace{\Big[1 - \Big(p_t p_D + (1 - p_t)q_D\Big)\Big]p_A}_{\text{no data loss but ACK loss}}. \qquad (51)$$

ϕ is the threshold for unicast or multicast retransmissions. If the current number of nodes n_k, which need a retransmission is smaller than the threshold ϕ, then unicast is used for the retransmission. p_t is the probability that the current number of nodes n_k is smaller than the threshold ϕ:

$$p_t = \frac{1}{M^{H3}} \sum_{k=1}^{M^{H3}} \begin{cases} 1, n_k < \phi \\ 0, n_k \geq \phi \end{cases} \qquad (52)$$

Since p_t is used to obtain M^{H3}, p_t can only be determined if $q_D = p_D$. In this case, parameter p_t is unnecessary to determine M^{H3}.

N_u is the mean number of receivers per round for which a unicast retransmission is invoked:

$$N_u = \frac{1}{M_u^{H3}} \sum_{k=1}^{M^{H3}} \begin{cases} n_k, n_k < \phi \\ 0, n_k \geq \phi \end{cases} \qquad (53)$$

$E(\tilde{N}_r^{H3})$ is the total number of transmissions that reach receiver r with unicast and multicast from its parent node in the ACK tree:

$$E(\tilde{N}_r^{H3}) = \frac{N_u}{B}E(M_u^{H3})(1 - p_D) + E(M_m^{H3})(1 - q_D). \qquad (54)$$

The mean number of ACKs that reach the sender or group leader in the presence of ACK loss is given by:

$$E(\tilde{L}_a^{H3}) = BE(\tilde{N}_r^{H3})(1 - p_A)p_c. \qquad (55)$$

Here we assume that only receivers of the same local group acknowledge transmissions. p_c is the probability that no AAK can be sent due to missing AAKs of child nodes.

The number of AAK query rounds O_q^{H3}, is determined by the probability $\hat{p}$ that a query fails:

$$\hat{p} = p_q + (1 - p_q)p_{AA}. \qquad (56)$$

$E(O_q^{H3})$ can be obtained with Eq. 3 and $\hat{p}$ instead of $\tilde{p}$. B_{aa} is the number of receivers, the sender has to query when the first AAK timeout occurs, which is equal to the number of receivers that have not already successfully sent an AAK in the first phase:

$$E(O_q^{H3}) = \sum_{i=1}^{B_{aa}} \binom{B_{aa}}{i}(-1)^{i+1}\frac{1}{1 - \hat{p}^i} \qquad (57)$$

$$B_{aa} = B\Big(p_c + (1 - p_c)p_{AA}\Big)^{E(\tilde{N}_r^{H3})}. \qquad (58)$$

Table 2
ADDITIONAL NOTATIONS FOR THE ANALYSIS OF (H3) AND (H4)

X_{aa}, Y_{aa}	Time to receive and process an AAK at the sender, or process the transmission of an AAK at the receiver.
X_{aaq}, Y_{aaq}	Time to send an AAK query at the sender, or receive and process an AAK query at the receiver.
$X_{d,u}$, $X_{d,m}$	Time to send a data packet by means of unicast or multicast, respectively.
W_{aa}, W_{aaq}	Bandwidth for an AAK or AAK query packet, respectively.
$W_{aa,\phi}$, $W_{aaq,\phi}$	Proportionate bandwidth for a periodical AAK or AAK query packet, respectively.
$W_{d,u}$, $W_{d,m}$	Bandwidth to send a data packet with unicast or multicast, respectively.
p_q, p_{AA}	Probability for AAK query loss at the receiver or AAK loss at the sender, respectively.
n_k	Current number of receivers that need a retransmission.
ϕ	Threshold for unicast retransmission. If $n_k < \phi$, unicast is used for a retransmission and multicast otherwise.
p_t	Probability that n_k is smaller than the threshold ϕ for unicast retransmissions and therefore unicast is used.
$\hat{p}$	Probability that an AAK query fails.
N_u	Mean number of sent unicast messages per packet retransmission.
M_u^{H3}, M_m^{H3}	Number of necessary unicast or multicast transmissions in the presence of failures.
O_q^w	Number of necessary AAK query rounds.
L_a^w, L_{aa}^w	Number of ACKs or AAKs sent by a receiver.
$\widetilde{L}_a^w$, $\widetilde{L}_{aa}^w$	Number of ACKs or AAKs received by the sender.
L_{aaq}^w, $\widetilde{L}_{aaq}^w$	Number of AAK queries sent by the sender or received by a receiver, respectively.
B_{aa}	Number of receivers from which the AAK is missing when phase two starts.
p_c	Probability that no AAK can be sent due to missing AAKs of child nodes.

$p_c + (1-p_c)p_{AA}$ is the probability that no AAK can be sent in a round or that the AAK is lost.

Queries are sent with unicast to the nodes whose AAK is missing. The total number of queries in all rounds are:

$$E(L_{aaq}^{H3}) = \sum_{k=1}^{E(O_q^{H3})} B_{aa}\hat{p}^{(k-1)}. \tag{59}$$

The number of AAKs received at the sender is the number of AAKs in the retransmission phase plus the number of AAKs in the AAK query phase, which is exactly one AAK from every receiver in B_{aa} (see Eq. 55):

$$E(\widetilde{L}_{aa}^{H3}) = BE(\widetilde{N}_r^{H3})(1-p_{AA})(1-p_c) + B_{aa}. \tag{60}$$

E.2 Receiver (Leaf Node)

The processing and bandwidth requirements at the receiver are given by:

$$\begin{aligned} E(P_R^{H3}) &= E(Y_f) + E(\widetilde{N}_{r,t}^{H3})E(Y_d) + E(L_a^{H3})E(Y_a) \\ &\quad + E(L_{aa}^{H3})E(Y_{aa}) + E(\widetilde{L}_{aaq}^{H3})\Big(E(Y_{aa}) + E(Y_{aaq})\Big) \end{aligned} \tag{61}$$

$$\begin{aligned} E(W_R^{H3}) &= E(\widetilde{N}_{r,t}^{H3})E(W_d) + E(L_a^{H3})E(W_a) \\ &\quad + E(L_{aa}^{H3})E(W_{aa}) + E(\widetilde{L}_{aaq}^{H3})\Big(E(W_{aa}) + E(W_{aaq})\Big). \end{aligned} \tag{62}$$

$\widetilde{N}_{r,t}^{H3}$ is the total number of transmissions that reach receiver r. In contrast to the already obtained $\widetilde{N}_r^{H3}$, additional data retransmissions are considered from other local groups that may be received with probability p_l:

$$\begin{aligned} E(\widetilde{N}_{r,t}^{H3}) &= \frac{N_u}{B}E(M_u^{H3})(1-p_D) + E(M_m^{H3})(1-q_D) \\ &\quad + \Big(E(M_m^{H3}) - 1\Big)(G-1)p_l. \end{aligned} \tag{63}$$

The number of transmissions that are acknowledged with an ACK, L_a^{H3}, or with an AAK, L_{aa}^{H3} are:

$$L_a^{H3} = p_c E(\widetilde{N}_r^{H3}) \tag{64}$$

$$L_{aa}^{H3} = (1-p_c)E(\widetilde{N}_r^{H3}). \tag{65}$$

$\widetilde{L}_{aaq}^{H3}$, the number of AAK queries received by an receiver is now:

$$\widetilde{L}_{aaq}^{H3} = \frac{1}{B_{aa}}E(L_{aaq}^{H3})(1-p_q), \tag{66}$$

where $1/B_{aa}$ is the probability to be a receiver that gets a unicast AAK query.

E.3 Group Leader (Inner Node)

The requirements for a group leader consist of the sender and receiver requirements (see Eq. 22):

$$\begin{aligned} E(P_G^{H3}) &= E(P_S^{H3}) + E(P_R^{H3}) - E(X_f) - E(X_{d,m}(1)) \\ &\quad - \Big(E(M_m^{H3}) - 1\Big)p_l E(Y_d) \end{aligned} \tag{67}$$

$$\begin{aligned} E(W_G^{H3}) &= E(W_S^{H3}) + E(W_R^{H3}) - E(W_{d,m}(1)) \\ &\quad - \Big(E(M_m^{H3}) - 1\Big)p_l E(W_d). \end{aligned} \tag{68}$$

The rate for sender, receiver and group leader as well as the overall system throughput and bandwidth consumption for (H3) can be obtained analogous to (H1).

F. NAK and AAK-based Protocol (H4)

Analogous to protocol (H3), the correct transmission of a data packet consists of two phases. In the first phase, the data is transmitted. If NAKs are received by the sender or group leaders, retransmissions are invoked. We assume that the retransmission phase has been finished before the second phase starts. In this phase AAKs are sent from receivers to their parent in the ACK tree. Missing AAKs are queried with unicast messages by the sender and group leaders. In a NAK-based protocol this is only reasonable if it is done after a certain number of correct data packet transmissions rather than after every transmission. Therefore, the costs for sending ($Y_{aa,\phi}$) and receiving AAKs ($X_{aa,\phi}$) as well as the costs for querying AAKs ($X_{aaq,\phi}$, $Y_{aaq,\phi}$) can be set to a proportionate cost of the other costs. The same applies for the bandwidth costs ($W_{aa,\phi}$ and $W_{aaq,\phi}$).

F.1 Sender (Root Node)

At the sender, the processing and bandwidth requirements can be obtained analogous to (H2) and (H3):

$$\begin{aligned} E(P_S^{H4}) &= E(X_f) + E(M^{H4})E(X_d) + E(\tilde{L}^{H4})E(X_n) \\ &\quad + E(O^{H4})E(X_t) + E(O_q^{H4})E(X_t) \\ &\quad + E(L_{aaq}^{H4})E(X_{aaq,\phi}) + E(\tilde{L}_{aa}^{H4})E(X_{aa,\phi}) \qquad (69) \\ E(W_S^{H4}) &= E(M^{H4})E(W_d) + E(\tilde{L}^{H4})E(W_n) \\ &\quad + E(L_{aaq}^{H4})E(W_{aaq,\phi}) + E(\tilde{L}_{aa}^{H4})E(W_{aa,\phi}). \qquad (70) \end{aligned}$$

$E(M^{H4})$, $E(\tilde{L}^{H4})$ and $E(O^{H4})$ are determined analogous to protocol (H2).

The mean number of AAK query rounds $E(O_q^{H4})$ is obtained analogous to Eq. 57 of protocol (H3) with failure probability $\hat{p} = p_q + (1 - p_q)p_{AA}$ and a modified B_{aa}. B_{aa} is the number of receivers, the sender has to query when the first AAK timeout at the sender occurs. Since receivers send one AAK autonomously after a certain number of successful receptions, the number of nodes to query in phase two is the number of lost AAKs, so $B_{aa} = Bp_{AA}$.

The total number of unicast query messages in all rounds L_{aaq}^{H4} is obtained analogous to Eq. 59 of protocol (H3). Using unicast, only those nodes are queried whose AAK is missing. So finally, the mean number of received AAKs at the sender is equal to the number of child nodes in the ACK tree:

$$E(L_{aa}^{H4}) = B. \qquad (71)$$

F.2 Receiver (Leaf Node)

The processing and bandwidth requirements are analogous to (H2) and (H3):

$$\begin{aligned} E(P_R^{H4}) &= E(Y_f) + E(M^{H4})(1 - q_D)E(Y_d) \\ &\quad + [E(O_r^{H4} - 1]\frac{\vartheta_2}{\vartheta_3}E(Y_n) + [E(O_r^{H4} - 2]E(Y_t) \\ &\quad + E(Y_{aa,\phi}) + E(\tilde{L}_{aaq}^{H4})\Big(E(Y_{aaq,\phi}) + E(Y_{aa,\phi})\Big) \\ &\quad \underbrace{+\Big[[E(O^{H4}) - 1]\vartheta_2 - [E(O_r^{H4}) - 1]\frac{\vartheta_2}{\vartheta_3}\Big](1 - q_N)E(X_n)}_{\text{from this local group}} \\ &\quad \underbrace{+ (G-1)p_l\Big[\Big(E(M^{H4}) - 1\Big)E(Y_d) + [E(O^{H4}) - 1]\vartheta_2 E(X_n)\Big]}_{\text{from other local groups}} \qquad (72) \end{aligned}$$

$$\begin{aligned} E(W_R^{H4}) &= E(M^{H4})(1 - q_D)E(W_d) + [E(O_r^{H4}) - 1]\frac{\vartheta_2}{\vartheta_3}E(W_n) \\ &\quad + E(W_{aa,\phi}) + E(\tilde{L}_{aaq}^{H4})\Big(E(W_{aaq,\phi}) + E(W_{aa,\phi})\Big) \\ &\quad + \Big[[E(O^{H4}) - 1]\vartheta_2 - [E(O_r^{H4}) - 1]\frac{\vartheta_2}{\vartheta_3}\Big](1 - q_N)E(W_n) \\ &\quad + (G-1)p_l\Big[\Big(E(M^{H4}) - 1\Big)E(W_d) \\ &\qquad + [E(O^{H4}) - 1]\vartheta_2 E(W_n)\Big]. \qquad (73) \end{aligned}$$

ϑ_2 and ϑ_3 can be obtained analogous to (H2). For $E(\tilde{L}_{aaq}^{H4})$, the mean number of received AAK queries and replied AAKs see Eq. 66.

F.3 Group Leader (Inner Node)

As the group leader role contains the sender role and the receiver role as well, the processing and bandwidth requirements are (see protocol (H2):

$$\begin{aligned} E(P_G^{H4}) &= E(P_S^{H4}) + E(P_R^{H4}) - E(X_f) - E(X_d(1)) \\ &\quad - p_l\Big[\Big(E(M^{H4}) - 1\Big)E(Y_d) + [E(O^{H4}) - 1]\vartheta_2 E(Y_n)\Big] \qquad (74) \\ E(W_G^{H4}) &= E(W_S^{H4}) + E(W_R^{H4}) - E(W_d(1)) \\ &\quad - p_l\Big[\Big(E(M^{H4}) - 1\Big)E(W_d) + [E(O^{H4}) - 1]\vartheta_2 E(W_n)\Big]. \qquad (75) \end{aligned}$$

The rate for sender, receiver and group leader as well as the overall system throughput and bandwidth consumption for (H4) can be obtained analogous to (H1).

V. 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, 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 [2] with group sizes of 25 to 100 nodes in networks of 1000 to 2000 nodes. In section VI, the simulations are introduced in more detail. Unfortunately, it was not possible to simulate a sparse multicast group, e.g. 100 receivers in a network of 100000 nodes, since the used simulator NS2 [15] 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 2 shows the throughput of all analyzed protocol classes with respect to the processing requirements. In Figure 2.a it is assumed

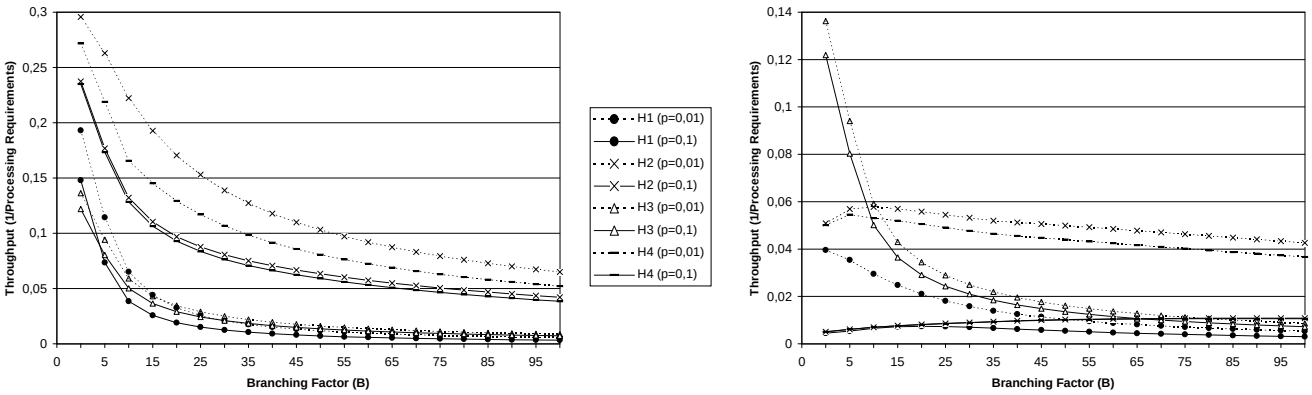

Fig. 2. Throughput limited by processing requirements with scope overlapping (a) $p_l = 0$ (left side) and (b) $p_l = 0.1$ (right side)

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 2.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 2 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 2 for all protocols as Figure 2.a shows.

The performance of protocol (H3) is independent of the scope overlapping probability always constant, since retransmissions are always sent with unicast. 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 optimal branching factor for protocol (H2) with respect to processing requirements is 5 whereas with 100000 receivers it is 80.

Figure 3 shows the throughput with respect to bandwidth requirements. The results are similar to Figure 2, 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 2.b and Figure 3.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.

Finally, Figure 4 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 imperfect confined local groups, larger local group sizes are preferable. In contrast to the throughput results, we cannot identify in Figure 4.b an optimal value within the displayed range of up to 100 child nodes per group leader. In fact, total bandwidth consumption reacts very sensitive to imperfect confined local groups, so that 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.

VI. Simulation Results

We have implemented the TMTP [2] reliable multicast protocol in the NS2 [15] network simulator environment to compare the analytical results with simulated results. We have used two networks generated with Tiers [16] and GT-ITM [17]. Both networks consist of 1000 nodes. All nodes in the network use DVMRP [18] routing. To simulate message loss, each link in the network is configured with probability 0.1% for message loss. We have measured an average end-to-end message loss probability for data packets of about 5%.

During the simulation, 100 nodes join the multicast group and therefore the ACK tree. The ACK tree is created by TRS [3] with a branching factor in the range from 2 to 30. After all nodes have joined the

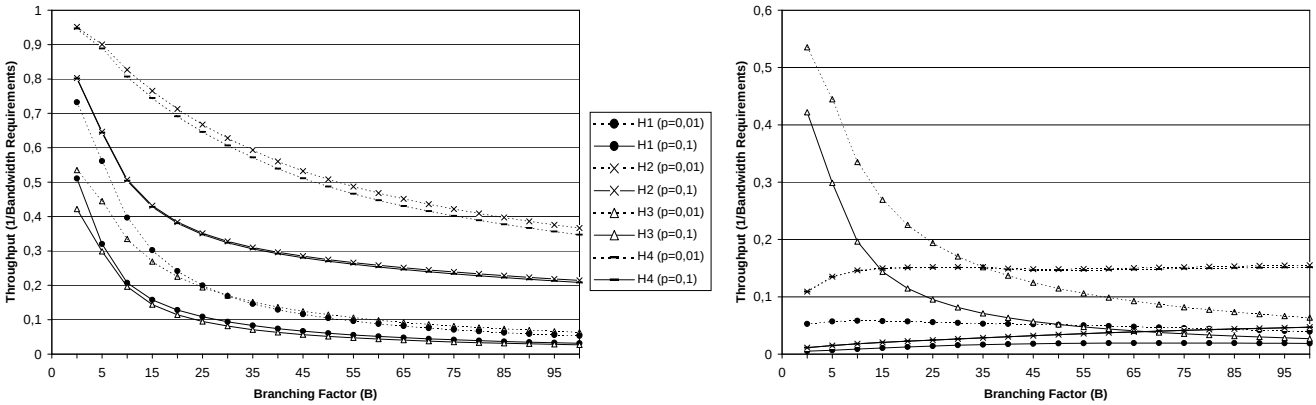

Fig. 3. Throughput limited by bandwidth requirements with scope overlapping (a) $p_l = 0$ (left side) and (b) $p_l = 0.1$ (right side)

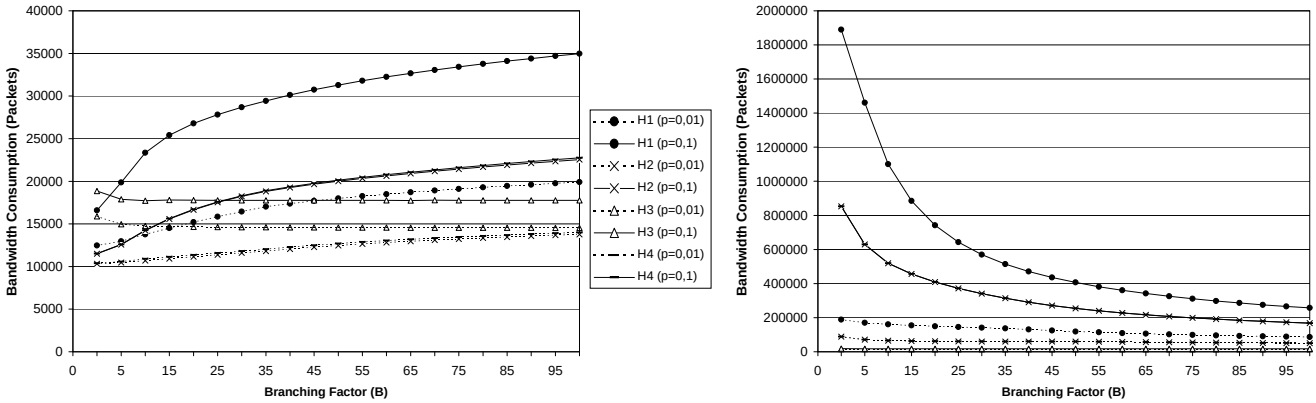

Fig. 4. Bandwidth consumption with scope overlapping (a) $p_l = 0$ (left side) and (b) $p_l = 0.1$ (right side)

ACK tree, data packets are transmitted to the group members using the TMTP protocol. The sending rate was 1 packet per second during the simulation time of 40 seconds. Additionally, we have assumed a selective repeat retransmission scheme with a window size of 2. Since 100 multicast members in a network consisting of 1000 nodes is a very dense group, the measured scope overlapping probability was rather high with $p_l = 0.5$.

Figure 5.a shows the throughput limited by bandwidth requirements for the analytical protocol class (H2) and the TMTP simulation. Figure 5.b shows the total bandwidth consumption. Since the measured end-to-end message loss probabilities were not constant during the simulations, we have indicated the measured loss probabilities in the figures. The displayed analytical results are swaying, since they are based on the measured loss probabilities, too.

The measured results for TMTP are the normalized average throughput and bandwidth consumption per reliable data packet transmission of 10 simulation runs with randomly selected receivers joining the group.

As expected, the results for the analytical model and TMTP are not completely identical; however, both show identical behaviour with varying loss probabilities and varying branching factors. In this scenario, an increased branching factor leads to increased throughput and decreased bandwidth consumption. Note that if the message loss probability is higher for a certain measurement, for example for branching factor 30 as displayed in the figure, this results in an decreased throughput and increased bandwidth consumption. This must be taken into account when assessing the effect of the branching factor on the protocol's performance. We have performed further simulation studies with other link loss probabilities, and other network sizes. All simulations show similar results.

In summary, we have shown that the simulation results of a realistic reliable multicast protocol are very close to the predicted results by the analysis, which confirms the suitability of our assumed system model and the analysis.

VII. SUMMARY

We have analyzed the processing and bandwidth requirements of four different classes 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 results of our analysis are confirmed by simulation studies.

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 2. On the other hand, if

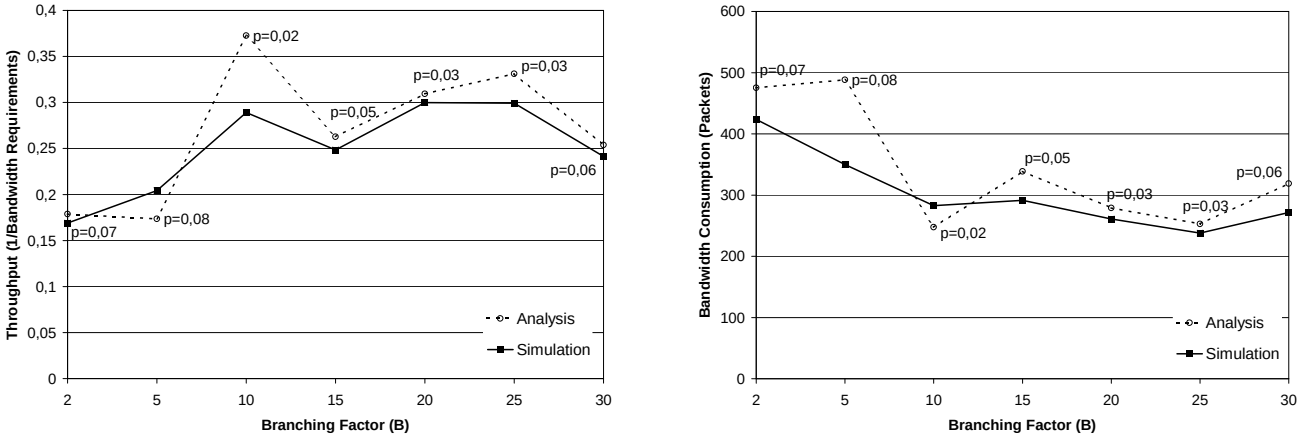

Fig. 5. Comparison of analytical and simulation results for (a) throughput and (b) bandwidth consumption

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] 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, Nov. 1996, pp. 333–344, ACM Press.

[3] 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, Oct. 1999, pp. 139–145, IEEE.

[4] C. Maihöfer and K. Rothermel, "Constructing height-balanced multicast acknowledgment trees with the Token Repository Service," Technical Report TR 1999/15, University of Stuttgart, 1999.

[5] 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, May 1994, pp. 221–230, ACM Press.

[6] 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'98*, New York, Apr. 1998, pp. 988–995, IEEE.

[7] J. Nonnenmacher, M. Lacher, M. Jung, G. Carl, and E. Biersack, "How bad is reliable multicast without local recovery," in *Proceedings of IEEE INFOCOM'98*, New York, Apr. 1998, pp. 972–979, IEEE.

[8] 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.

[9] 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, Oct. 2000, pp. 250–257, IEEE.

[10] 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, Nov. 2000, pp. 15–26, ACM.

[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] B. Whetten and G. Taskale, "An overview of the reliable multicast transport protocol II," *IEEE Network*, vol. 14, no. 1, pp. 37–47, Feb. 2000.

[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.

[14] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proceedings of IEEE INFOCOM'99*, New York, 1999, pp. 345–352.

[15] S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, M. Handley P. Haldar, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, and D. Zappala, "Improving simulation for network research," Technical Report 99-702, University of Southern California, 1999.

[16] K. Calvert, M.B. Doar, and E.W. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, June 1997.

[17] E. Zegura, K. Calvert, and M. Donahoo, "A quantitative comparison of graph-based models for internet topology," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 770–783, Dec. 1997.

[18] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol," RFC 1075, 1988.