A Bandwidth Analysis of Reliable Multicast Transport Protocols*

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

ABSTRACT

Multicast is an efficient communication technique to save bandwidth for group communication purposes. A number of protocols have been proposed in the past to provide a reliable multicast service. Briefly classified, they can be distinguished into sender-initiated, receiver-initiated and tree-based approaches.

In this paper, an analytical bandwidth evaluation of generic reliable multicast protocols is presented. Of particular importance are new classes with aggregated acknowledgments. In contrast to other approaches, these classes provide reliability not only in case of message loss but also in case of node failures. Our analysis is based on a realistic system model, including data packet and control packet loss, asynchronous local clocks and imperfect scope-limited local groups.

Our results show that hierarchical approaches are superior. They provide higher throughput as well as lower bandwidth consumption. Relating to protocols with aggregated acknowledgments, the analysis shows only little additional bandwidth overhead and therefore high throughput rates.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.4 [**Performance of Systems**]

General Terms

Bandwidth, Analysis, Reliable Multicast

Keywords

AAK, receiver-initiated, sender-initiated, tree-based

1. INTRODUCTION

A number of reliable multicast transport protocols have been proposed in the literature, which are based on the acknowledgment scheme. Reliability is ensured by replying acknowledgment messages from the receivers to the sender, either to confirm correct data packet delivery or to ask for a retransmission. Reliable multicast protocols are usually classified into sender-initiated, receiver-initiated and tree-based ones. Briefly characterized, in sender-initiated approaches receivers reply positive acknowledgments (ACKs) to confirm correct message delivery in contrast to receiver-initiated protocols, which indicate transmission errors or losses by negative acknowledgments (NAKs). Both classes can result in an overwhelming of the sender and the network around the sender by a large number of ACK or NAK messages. This problem is the well-known acknowledgment implosion problem, which is a vital challenge for the design of reliable multicast protocols, since it limits the scalability for large receiver groups. Tree-based approaches promise to be scalable even for a large number of receivers, since they arrange receivers into a hierarchy, called ACK tree [10]. Leaf node receivers send their positive or negative acknowledgments to their parent node in the ACK tree. Each non-leaf receiver is responsible for collecting ACKs or NAKs only from their direct child nodes in the hierarchy. Since the maximum number of child nodes is limited, no node is overwhelmed with messages and scalability for a large receiver group is ensured. The maximum number of child nodes can be determined according to the processing performance of a node, its available network bandwidth, its memory equipment, and its reliability.

In this paper we present a throughput analysis based on bandwidth requirements as well as the overall bandwidth consumption of all group members, which refer to the data transfer costs. One characteristic of multicast transmissions is that the component with the weakest performance may determine the transmission speed. This means, a group member with a low bandwidth connection, low processing power, high packet loss rate or high packet delay may prevent high transmission rates. Therefore, it is very useful to be able to quantify the necessary requirements for a given multicast protocol.

The remainder of this paper is structured as follows. In Section 2 we discuss the background of our throughput analysis and take a look at related work. In Section 3 we briefly classify the analyzed protocols. Our bandwidth evaluation in Section 4 starts with a definition of the assumed system model before the various protocol classes are analyzed in detail. To illustrate the results, some numerical evaluations are presented in Section 5 before we conclude with a brief

*

summary.

2. RELATED WORK

Reliable multicast protocols were already analyzed in previous work. The first processing requirements analysis of generic reliable multicast protocols was presented by Pingali et al. [8]. They compared the class of sender- and receiver-initiated protocols. Following analytical papers are often based on the model and analytical methods introduced by [8]. Levine et al. [3] have extended the analysis to the class of ring- and tree-based approaches. In Maihöfer et al. [5] protocols with aggregated acknowledgments are considered.

A bandwidth analysis of generic reliable multicast protocols was done by Kasera et al. [2], Nonnenmacher et al. [6] and Poo et al. [9]. In [2], 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 [6] 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 avoidance. In [9], 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 differs from previous work in the following ways. First, we consider the loss of data packets *and* control packets. Second, we assume that local clocks are not synchronized which affects the NAK-avoidance scheme (see Section 3.2). NAK-avoidance works less efficiently with this more realistic assumption. Third, our analysis considers that local groups may not be confined perfectly, so that local data or control packets may reach nodes in other local groups. Finally, our work extends previous analysis by two new tree-based protocol classes. They are based on aggregated ACKs to be able to cope with node failures.

3. CLASSIFICATION OF RELIABLE MULTICAST PROTOCOLS

In this section we briefly classify the reliable multicast protocols analyzed in this paper. A more detailed and more general description for some of these classes can be found in [8], [3] and [6].

3.1 Sender-Initiated Protocols

The class of sender-initiated protocols is characterized by positive acknowledgments (ACKs) returned by the receivers to the sender. A missing ACK detects either a lost data packet at the corresponding receiver, a lost ACK packet or a crashed receiver, which cannot be distinguished by the sender. Therefore, a missing ACK packet leads to a data packet retransmission from the sender. We assume that such a retransmission is always sent using multicast. This protocol class will be referred to as (A1). Note that the use of negative acknowledgments, for example to speed up retransmissions, does not necessarily mean that a protocol is not of class (A1). Important is that positive acknowledgments are necessary, for example to release data from the sender's buffer space. An example for a sender-initiated protocol is the Xpress Transport Protocol (XTP) [12].

3.2 Receiver-Initiated Protocols

In contrast to sender-initiated protocols, receiver-initiated protocols return only negative acknowledgments (NAKs) instead of ACKs. As in the sender-initiated protocol class, we assume that retransmissions are sent using multicast. When a receiver detects an error, e.g. by a wrong checksum, a skip in the sequence number or a timeout while waiting for a data packet, a NAK is returned to the sender. Pure receiver-initiated protocols have a non-deterministic characteristic, since the sender is unable to decide when all group members have correctly received a data packet.

Receiver-initiated protocols can either send NAKs using unicast or multicast transmission. The protocol class sending unicast NAKs will be called (N1). An example for (N1) is PGM [11]. The approach using multicast NAKs (N2) is known as NAK-avoidance scheme. A receiver that has detected an error sends a multicast NAK provided that it has not already received a NAK for this data packet from another receiver. Thus, in optimum case, only one NAK is received by the sender for each lost data packet. An example for such a protocol is the scalable reliable multicasting protocol (SRM) [1].

3.3 Tree-Based Protocols

Tree-based approaches organize the receivers into a tree structure called ACK tree, which is responsible for collecting acknowledgments and sending retransmissions. We assume that the sender is the root of the tree. If a receiver needs a retransmission, the parent node in the ACK tree is informed rather than the sender. The parent nodes are called group leaders for their children which form a local group. Note that a group leader may also be a child of another local group. A child which is only a receiver rather than a group leader is called leaf node.

The first considered scheme of this class (H1) is similar to sender-initiated protocols since it uses ACKs sent by the 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 group leader until the root node is reached. If a timeout for an ACK occurs at a group leader or the root, a multicast retransmission is invoked. An example of a protocol similar to our definition of (H1) is RMTP [7]. The second scheme (H2) is based on NAKs with NAK suppression similar to (N2) and selective ACKs (SAKs), which are sent periodically for deciding deterministically when packets can be removed from memory. A SAK is sent to the parent node after a certain number of packets are received or after a certain time period has expired, to propagate the state of a receiver to its group leader. TMTP [14] is an example for class (H2).

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 group leader after all children have acknowledged correct reception. After a group leader or the root node has received an AAK, it can remove the corresponding data from memory because all members in this subhierarchy have already received it correctly. Lorax [4] and RMTP-II [13] are examples for AAK protocols. Our definition of (H3)'s generic behavior is as follows:

1. Group leaders send a local ACK after the data packet is received correctly.
2. Leaf node receivers send an AAK after the data packet is received correctly.
3. The root node and group leaders wait a certain time to receive local ACKs from their 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 local ACKs, a received AAK from a receiver is also allowed to prevent the retransmission.
4. The root node and 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 group leader. If a timeout occurs while waiting for AAKs, a unicast AAK query is sent to the affected nodes.
5. If a group leader or leaf node receiver receives further retransmissions after an AAK has been sent or the prerequisites for sending an AAK are met, these data packets are acknowledged by AAKs rather than ACKs. The same applies for receiving an AAK query that is replied with an AAK if the prerequisites are met.

In summary, ACKs are used for fast error recovery in case of message loss and AAKs to clear buffer space. Besides the AAK scheme, we consider in our analysis of (H3) a threshold scheme to decide whether a retransmission is performed using unicast or multicast. The sender or 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.

Our next protocol will be denoted as (H4) and is a combination of the negative acknowledgment with NAK suppression scheme (H2) and aggregated acknowledgments (H3). 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 and the sender to remove data from memory. Like SAKs, we assume that AAKs are sent periodically. We define the generic behavior of (H4) as follows:

1. Upon detection of a missing or corrupted data packet, receivers send a NAK per 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 and the sender retransmit a packet per multicast if a NAK has been received.
3. After a certain number of correctly received data packets, leaf node receivers send an AAK to its group leader in the ACK tree. A group leader forwards this AAK to its parent group leader or sender, respectively, as soon as the same data packets are correctly received and the corresponding AAKs from all child nodes are received.
4. The sender and group leaders initiate a timer to 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 a group leader or leaf node receiver gets an AAK query and the prerequisites for sending an AAK are met, the query is acknowledged with an AAK.

4. BANDWIDTH ANALYSIS

4.1 Model

Our model is similar to the one used by Pingali et al. [8] and Levine et al. [3]. A single sender is assumed, multicasting to R identical receivers. In case of tree-based protocols, the sender is the root of the ACK tree. We assume that nodes do not fail and the network is not partitioned, i.e. that retransmissions are finally successful. In contrast to previous work, packet loss can occur on both, data packets *and* control packets. Multicast packet loss probability is given by q and unicast packet loss probability by p for any node. Table 1 summarizes the notations for protocol classes (A1), (N1), (N2), (H1) and (H2). In Table 2 the additional notations for the protocol classes (H3) and (H4) are given.

We assume that losses at different nodes are independent events. In fact, since receivers share parts of the multicast routing tree, this assumption does not hold in real networks. However, if all classes have similar trees no protocol class is privileged relative to another one by this assumption.

In the following subsections, the generic protocol classes are analyzed in detail. Although our work considers more protocols and a more general system model, the notations and basic analyzing methods follows [8] and [3].

4.2 Sender-Initiated Protocol (A1)

We determine the bandwidth requirements at the sender W_S^{A1} and each receiver W_R^{A1}, based on the necessary bandwidth 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 bandwidth consumption at the sender is:

$$W_S^{A1} = \text{(initial transmission)} + \text{(retransmissions)} + \text{(receiving ACKs)}$$

$$W_S^{A1} = W_d(1) + \sum_{m=2}^{M^{A1}} W_d(m) + \sum_{i=1}^{\tilde{L}^{A1}} W_a(i) \tag{1}$$

$$E(W_S^{A1}) = E(M^{A1})E(W_d) + E(\tilde{L}^{A1})E(W_a). \tag{2}$$

Table 1: Notations for the analysis of (A1), (N1), (N2), (H1) and (H2)

R	Size of the receiver set.
B	Branching factor of a tree or the local group size.
W_d, W_a, W_n, W_ϕ	Bandwidth for a data packet, ACK, NAK and SAK, respectively.
W_S^w, W_R^w, W_H^w, W^w	Bandwidth requirements for protocols w at the sender, receiver, group leader and overall bandwidth consumption. $w \in \{A1, N1, N2, H1, H2, H3, H4\}$
$\Lambda_S^w, \Lambda_R^w, \Lambda_H^w, \Lambda^w$	Throughput respectively relative bandwidth efficiency for protocols w at the sender, receiver, group leader and overall system throughput.
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, q_N	Probability for unicast ACK, NAK or multicast NAK loss.
$\tilde{p}, \bar{p}$	Probability that a retransmission is necessary for protocol (A1) or (N2), respectively.
p_s	Probability for simultaneous and therefore unnecessary NAK sending in (N2), (H2) and (H4).
p_l	Probability for receiving a data or control packet from another local group.
$\widetilde{L}_r^w, \widetilde{L}^w$	Number of ACKs or NAKs per data packet sent by receiver r that reach the sender or total number of ACKs or NAKs per data packet received from all receivers.
N_r^w, $N_{r,t}^w$	Total number of transmissions per data packet received by receiver r from the parent node or total number of received data packets from all local groups, respectively.
N_g^w	Total number of transmissions per data packet received by a group leader from all local groups.
M_r^w, M^w	Number of necessary transmissions for receiver r, to receive a data packet correctly in the presence of data and ACK or NAK loss or total number of transmissions for all receivers.
O^w, O_r^w	Number of necessary rounds to correctly deliver a packet to all receivers or to receiver r.
$O_e^w, O_{e,r}^w$	Total number of empty rounds or empty rounds for receiver r, respectively.
N_k	Number of NAKs sent in round k.

$W_d(m)$ and $W_a(i)$ are the bandwidths required for a data packet or ACK packet for the m-th or i-th transmission, respectively. M^{A1} 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 $\widetilde{L}^{A1}$ is the total number of ACKs received for this data packet. $E(W_S^{A1})$ is the expectation of the bandwidth requirement at the sender. The only unknowns are $E(M^{A1})$ and $E(\widetilde{L}^{A1})$:

$$E(\widetilde{L}^{A1}) = RE(M^{A1})(1-q_D)(1-p_A). \tag{3}$$

This means, the sender gets one ACK per data packet transmission $E(M^{A1})$ from every receiver R, provided that the data packet is not lost with probability $(1-q_D)$ and the ACK is not lost with probability $(1-p_A)$.

Now, the number of transmissions have to be analyzed. The probability for a retransmission is:

$$\tilde{p} = q_D + (1-q_D)p_A, \tag{4}$$

i. e. either a data packet is lost (q_D) or the data packet is received correctly and the ACK is lost $((1-q_D)p_A)$. So, the probability that the number of necessary transmissions M_r^{A1} for receiver r is smaller or equal to m (m=1, 2, ...) is:

$$P(M_r^{A1} \le m) = 1 - \tilde{p}^m. \tag{5}$$

As the packet losses at different receivers are independent from each other:

$$\begin{aligned} P(M^{A1} \le m) &= \prod_{r=1}^{R} P(M_r^{A1} \le m) = (1-\tilde{p}^m)^R \\ &= \sum_{i=0}^{R} \binom{R}{i} (-1)^i \tilde{p}^{im} \qquad (6) \\ P(M^{A1} = m) &= P(M^{A1} \le m) - P(M^{A1} \le m-1) \\ &= \sum_{i=0}^{R} \binom{R}{i} (-1)^i \tilde{p}^{i(m-1)} (\tilde{p}^i - 1) \qquad (7) \\ E(M^{A1}) &= \sum_{m=1}^{\infty} m P(M^{A1} = m) \\ &= \sum_{m=1}^{\infty} m \sum_{i=0}^{R} \binom{R}{i} (-1)^i \tilde{p}^{im} (1-\tilde{p}^{-i}) \\ &= \sum_{i=0}^{R} \binom{R}{i} (-1)^i (1-\tilde{p}^{-i}) \sum_{m=1}^{\infty} m \tilde{p}^{im} \\ &= \sum_{i=1}^{R} \binom{R}{i} (-1)^i (1-\tilde{p}^{-i}) \frac{\tilde{p}^i}{(1-\tilde{p}^i)^2} \\ &= \sum_{i=1}^{R} \binom{R}{i} (-1)^{i+1} \frac{1}{1-\tilde{p}^i}. \qquad (8) \end{aligned}$$

$E(M^{A1})$ is the expected total number of necessary transmissions to receive the data packet correctly at all receivers.

Now $E(W_S^{A1})$ is entirely determined. The bandwidth efficiency respectively maximum throughput for the sender Λ_S^{A1}, to send data packets successfully to a receiver is:

$$\Lambda_S^{A1} = \frac{1}{E(W_S^{A1})}. \tag{9}$$

Accordingly, the processing requirement for a packet at the receiver is:

$$W_R^{A1} = (\text{receiving packets}) + (\text{sending ACKs}) \tag{10}$$

$$E(W_R^{A1}) = E(M^{A1})(1-q_D)(E(W_d) + E(W_a)), \tag{11}$$

where a packet is received with probability $(1-q_D)$ and each received packet is acknowledged. The maximum throughput Λ_r^{A1} of a receiver is:

$$\Lambda_R^{A1} = \frac{1}{E(W_R^{A1})}. \tag{12}$$

Overall system throughput Λ^{A1} is determined by the minimum of the throughput rates at the sender and receivers:

$$\Lambda^{A1} = min\{\Lambda_S^{A1}, \Lambda_R^{A1}\}. \tag{13}$$

Now we are able to determine the total bandwidth consumption. In contrast to previous work, our definition of total bandwidth consumption is the bandwidth that is necessary at the sender and receivers to send and receive messages. This means, we assume that the internal network structure is not known and therefore not considered in the analysis. In [2] and [6], total bandwidth is defined on a per link basis. Such a definition encompasses the total costs within the network but has the disadvantage, that a network topology has to be defined with routers and links between routers. Here, we want to determine the total costs at the communication endpoints, i.e. the costs for the sender and receivers.

The total bandwidth consumption of protocol (A1) is then the sum of the sender's and receivers' bandwidth consumptions:

$$E(W^{A1}) = E(W_S^{A1}) + RE(W_R^{A1}). \tag{14}$$

4.3 Receiver-Initiated Protocol (N1)

As in the sender-initiated protocol, data packets are always transmitted using multicast. In (N1), error control is realized by unicast NAKs. The sender collects all NAKs received within a certain timeout period and sends only one retransmission independent of the number of received or lost control packets during that round.

The bandwidth requirement at the sender is:

$$W_S^{N1} = (\text{transmissions}) + (\text{receiving NAKs})$$

$$W_S^{N1} = \sum_{m=1}^{M^{N1}} W_d(m) + \sum_{i=1}^{\widetilde{L}^{N1}} W_n(i) \tag{15}$$

$$E(W_S^{N1}) = E(M^{N1})E(W_d) + E(\widetilde{L}^{N1})E(W_n). \tag{16}$$

The only unknowns are $E(M^{N1})$ and $E(\widetilde{L}^{N1})$. The number of transmissions, M^{N1}, until all receivers correctly receive a packet is only determined by the probability for data packet loss analogous to Eq. 8 of (A1) with q_D instead of $\tilde{p}$.

To determine $E(\widetilde{L}^{N1})$, some intermediate steps have to be done. First, the number of transmissions, M_r^{N1}, for a single receiver is given by the probability q_D. This means, M_r^{N1} 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 $(1 - q_D)$ is:

$$P(X = k) = (1 - q_D){q_D}^{k-1}. \tag{17}$$

The necessary number of transmissions for a single receiver M_r^{N1} follows from the Bernoulli distribution and [8]:

$$E(M_r^{N1}) = \frac{1}{1 - q_D} \tag{18}$$

$$E(M_r^{N1}|M_r^{N1} > 1) = \frac{2 - q_D}{1 - q_D} \tag{19}$$

$$E(M_r^{N1}|M_r^{N1} > 2) = \frac{3 - 2q_D}{1 - q_D} \tag{20}$$

$$P(M_r^{N1} > 1)[E(M_r^{N1}|M_r^{N1} > 1) - 1] = E(M_r^{N1}) - 1. \tag{21}$$

Besides the necessary number of transmissions, we have to introduce the number of rounds, necessary to correctly deliver a data packet. A round starts with the sending of a data packet and ends with the expiration of a timeout at the sender. Normally, there will be one data transmission in each round. However, if the sender receives no NAKs due to NAK loss, no retransmission is made and new NAKs must be sent by the receivers in the next round. O_r^{N1} is the number of necessary rounds for receiver r. The number of rounds is the sum of the number of necessary rounds for sending transmissions M_r^{N1} and the number of empty rounds $O_{e,r}^{N1}$ in which all NAKs are lost and therefore no retransmission is made:

$$O_r^{N1} = M_r^{N1} + O_{e,r}^{N1}. \tag{22}$$

$E(M_r^{N1})$ is given in Eq. 18. $E(O_{e,r}^{N1})$ can be determined analogous to $E(M_r^{N1})$, with probability p_k for the loss of all sent NAKs in round k (see Eq. 25). The expected number of empty rounds $E(O_{e,r}^{N1})$ 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:

$$E(O_{e,r}^{N1}) = \sum_{k=1}^{E(M_r^{N1})-1} \left(\frac{1}{1-p_k} - 1\right). \tag{23}$$

$(1/1-p_k)$ is the expectation for the number of empty rounds plus the last successful NAK reception at the sender which is subtracted. N_k, the number of NAKs sent in round k is given by:

$$N_k = {q_D}^k R, \tag{24}$$

where ${q_D}^k$ is the probability for a single receiver that until round k all data packets are lost. The number of empty rounds after transmission k is determined by the failure probability:

$$p_k = {p_N}^{N_k} = {p_N}^{{q_D}^k R}. \tag{25}$$

p_k is the probability that all sent NAKs in round k are lost. The number of sent NAKs is equal to the number of receivers ${q_D}^k R$ that need a retransmission in round k (see Eq. 24).

$\widetilde{L}^{N1}$ is the number of NAKs received by the sender and ϑ_1 is the total number of NAKs sent in all rounds:

$$E(\widetilde{L}^{N1}) = \vartheta_1(1 - p_N) \tag{26}$$

$$\vartheta_1 = \sum_{k=1}^{E(M^{N1})} N_k \frac{1}{1-p_k}. \tag{27}$$

Finally, at the receiver we have:

$$E(W_R^{N1}) = E(M^{N1})(1 - q_D)E(W_d) + P(O_r^{N1} > 1)[E(O_r^{N1}|O_r^{N1} > 1) - 1]E(W_n). \tag{28}$$

Note that the last, successful transmission is not replied with a NAK.

The throughput rates are analogous to (A1):

$$\Lambda_S^{N1} = \frac{1}{E(W_S^{N1})}, \quad \Lambda_R^{N1} = \frac{1}{E(W_R^{N1})}, \quad \Lambda^{N1} = min\{\Lambda_S^{N1}, \Lambda_R^{N1}\}. \tag{29}$$

The total bandwidth consumption is:

$$E(W^{N1}) = E(W_S^{N1}) + RE(W_R^{N1}). \tag{30}$$

4.4 Receiver-Initiated Protocol (N2)

In contrast to (N1), this protocol class sends NAKs to all group members using multicast. Ideally, NAK suppression ensures that only one NAK is received by the sender. As in the previous protocol, the sender collects all NAKs belonging to one round and then starts a retransmission:

$$W_S^{N2} = (\text{transmissions}) + (\text{receiving NAKs})$$

$$W_S^{N2} = \sum_{m=1}^{M^{N2}} W_d(m) + \sum_{i=1}^{\widetilde{L}^{N2}} W_n(i) \tag{31}$$

$$E(W_S^{N2}) = E(M^{N2})E(W_d) + E(\widetilde{L}^{N2})E(W_n). \tag{32}$$

$E(M^{N2})$ is determined analogous to (A1) and (N1) with loss probability q_D (see Eq. 8). $\widetilde{L}^{N2}$ contains the number of necessary and additional NAKs received at the sender:

$$E(\widetilde{L}^{N2}) = \vartheta_1(1-q_N) \quad (33)$$

$$\vartheta_1 = \sum_{k=1}^{E(M^{N2})} N_k \frac{1}{1-p_k}. \quad (34)$$

N_k, the number of NAKs sent in round k, is the sum of: NAK of the first receiver that did not receive the data packet plus NAK of another unsuccessful receiver that did not receive the first NAK packet and sends a second NAK and so on:

$$N_k = \sum_{i=1}^{R} N_{k,i} \quad (35)$$

$$N_{k,1} = {q_D}^k \quad (36)$$

$$\begin{aligned} N_{k,2} &= {q_D}^k(1 - N_{k,1} + N_{k,1}q_N) \\ &= N_{k,1}(1 - N_{k,1} + N_{k,1}q_N) \\ &= N_{k,1} - N_{k,1}^2 + N_{k,1}^2 q_N \end{aligned} \quad (37)$$

$$N_{k,n} = N_{k,n-1} - N_{k,n-1}^2 + N_{k,n-1}^2 q_N \quad , n > 1. \quad (38)$$

The first receiver sends a NAK provided that the data packet was lost with probability ${q_D}^k$. The second receiver sends a NAK provided that the data packet was lost and the NAK of the first receiver was lost $(N_{k,1}q_N)$ or the first receiver sends no NAK $(1 - N_{k,1})$, and so on.

In Eq. 38, a perfect system model is assumed in which additional NAKs are only sent due to NAK loss at receivers. This means, receivers must have synchronized local clocks and a defined sending order for NAKs. However, since receivers are usually not synchronized in real systems it can occur that NAKs are sent simultaneously. Therefore, we extend Equations 36-38 with the probability for simultaneous NAK sending (p_s) to:

$$N_{k,1} = {q_D}^k \quad (39)$$

$$N_{k,n} = N_{k,n-1} - N_{k,n-1}^2 + N_{k,n-1}^2(q_N + p_s - q_N p_s) \ , n > 1. \quad (40)$$

The number of rounds O_r^{N2} for receiver r is obtained analogous to protocol (N1). It is the sum of the number of necessary rounds for sending transmissions M_r^{N2} and the number of empty rounds $O_{e,r}^{N2}$ in which all NAKs are lost and therefore no retransmission is made. The total number of rounds O^{N2} for all receivers can be defined analogous to O_r^{N2}:

$$O_r^{N2} = M_r^{N2} + O_{e,r}^{N2} \quad (41)$$

$$O^{N2} = M^{N2} + O_e^{N2}. \quad (42)$$

The number of necessary transmissions, M_r^{N2}, for a single receiver r is given by the probability q_D. Analogous to Eq. 18 of protocol (N1) the expectation is:

$$E(M_r^{N2}) = \frac{1}{1-q_D}. \quad (43)$$

The number of empty rounds after transmission k is determined by the failure probability:

$$p_k = q_N^{N_k}. \quad (44)$$

p_k is the probability that all sent NAKs in round k are lost. The expected number of empty rounds $E(O_e^{N2})$ is equal to the expected number of empty rounds after the first transmission plus the expected number of empty rounds after the second transmission and so on. Now, $E(O_e^{N2})$ and $E(O_{e,r}^{N2})$ can be determined analogous to M_r^{N2} (see Eq. 18):

$$E(O_e^{N2}) = \sum_{k=1}^{E(M^{N2})-1} \left(\frac{1}{1-p_k} - 1\right) \quad (45)$$

$$E(O_{e,r}^{N2}) = \sum_{k=1}^{E(M_r^{N2})-1} \left(\frac{1}{1-p_k} - 1\right). \quad (46)$$

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

At the receiver we have:

$$\begin{aligned} E(W_R^{N2}) &= E(M^{N2})(1-q_D)E(W_d) \\ &+ P(O_r^{N2} > 1)[E(O_r^{N2}|O_r^{N2} > 1) - 1]\frac{\vartheta_2}{\vartheta_3}E(W_n) \\ &+ [P(O^{N2} > 1)[E(O^{N2}|O^{N2} > 1) - 1]\vartheta_2 \\ &- P(O_r^{N2} > 1)[E(O_r^{N2}|O_r^{N2} > 1) - 1]\frac{\vartheta_2}{\vartheta_3}]\,(1-q_N)E(W_n). \end{aligned} \quad (47)$$

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 want to send a NAK:

$$\vartheta_2 = \frac{1}{E(O^{N2})}\sum_{k=1}^{E(M^{N2})} N_k \frac{1}{1-p_k} \quad (48)$$

$$\vartheta_3 = \frac{1}{E(O^{N2})}\sum_{k=1}^{E(M^{N2})} {q_D}^k R \frac{1}{1-p_k}, \quad (49)$$

where $(1/1-p_k)$ is the number of empty rounds plus the last successful NAK sending (see Eq. 18, 45 and 46).

The second term in $E(W_R^{N2})$ is the processing requirement to send NAKs, where the considered receiver r is only with probability ϑ_2/ϑ_3 the one that sends a NAK. In the third term the number of sent NAKs is subtracted from the number of total NAKs to get the number of received NAKs.

The throughput rates are:

$$\Lambda_S^{N2} = \frac{1}{E(W_S^{N2})}, \quad \Lambda_R^{N2} = \frac{1}{E(W_R^{N2})}, \quad \Lambda^{N2} = min\{\Lambda_S^{N2}, \Lambda_R^{N2}\}. \quad (50)$$

The total bandwidth consumption is:

$$E(W^{N2}) = E(W_S^{N2}) + RE(W_R^{N2}). \quad (51)$$

4.5 Tree-Based Protocol (H1)

Our analysis distinguishes between the three different kinds of nodes in the ACK tree, the sender at the root of the tree, the receivers that form the leaves of the ACK tree and the receivers that are inner nodes. We will call these inner receivers group leaders. Group leaders are sender and receiver as well.

Our analysis of all tree-based protocols 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 sent a retransmission, the group leader has already received this packet correctly. The following subsections analyze the bandwidth requirements at the sender W_S^{H1}, receivers W_R^{H1} and group leaders W_H^{H1}.

4.5.1 Sender (root node)

$$W_S^{H1} = W_d(1) + \sum_{m=2}^{M^{H1}} W_d(m) + \sum_{i=1}^{\widetilde{L}^{H1}} W_a(i) \quad (52)$$

$$E(W_S^{H1}) = E(M^{H1})E(W_d) + E(\widetilde{L}^{H1})E(W_a) \quad (53)$$

M^{H1} is the number of necessary transmissions until all members of a local group have received a packet correctly. $E(M^{H1})$ is determined analogous (B instead of R) to Eq. 8 of protocol (A1), since every local group is like a sender-based system. Furthermore, the number of ACKs received by the sender and group leaders in the presence of possible

ACK loss $E(\widetilde{L}^{H1})$ is similar to $E(\widetilde{L}^{A1})$, with B instead of R:

$$E(\widetilde{L}^{H1}) = BE(M^{H1})(1-q_D)(1-p_A). \tag{54}$$

4.5.2 Receiver (leaf node)

$E(N_{r,t}^{H1})$ is the total number of received transmissions at receiver r and consists mainly of the sent messages from the parent $E(N_r^{H1})$, provided that each local group has its own multicast address. However, if the whole multicast group has only one multicast address, retransmissions may reach members outside of this local group. The probability for receiving a retransmission from another local group is assumed to be p_l for any receiver. Such received transmissions from other local groups 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 many cases they are received after a local retransmission have already been triggered.

First we want to determine the number of group leaders. The number of nodes R in a complete tree with branching factor B and height h is:

$$\begin{aligned} R &= \sum_{i=0}^{h-1} B^i = B^0 + B^1 + \cdots + B^{h-2} + B^{h-1} \\ &= \frac{(1-B)B^0}{1-B} + \frac{(1-B)B^1}{1-B} + \cdots + \frac{(1-B)B^{h-2}}{1-B} + \frac{(1-B)B^{h-1}}{1-B} \\ &= \frac{B^0 - B^1 + B^1 - B^2 + \cdots + B^{h-2} - B^{h-1} + B^{h-1} - B^h}{1-B} \\ &= \frac{1-B^h}{1-B}, \end{aligned} \tag{55}$$

and the tree height follows to:

$$h = \log_B (R(B-1)+1). \tag{56}$$

The number of group leaders plus the sender is therefore:

$$G = \sum_{i=0}^{h-2} B^i. \tag{57}$$

The number of received transmissions $E(N_r^{H1})$ from the parent node at receiver r is:

$$E(N_r^{H1}) = E(M^{H1})(1-q_D). \tag{58}$$

The total number of received transmissions $E(N_{r,t}^{H1})$ at receiver r is now:

$$E(N_{r,t}^{H1}) = E(M^{H1})(1-q_D) + (G-1)E(M^{H1})(1-q_D)p_l. \tag{59}$$

Finally, the bandwidth requirement W_R^{H1} for a receiver is:

$$W_R^{H1} = \sum_{i=1}^{N_{r,t}^{H1}} W_d(i) + \sum_{j=1}^{N_r^{H1}} W_a(j) \tag{60}$$

$$E(W_R^{H1}) = E(N_{r,t}^{H1})E(W_d) + E(N_r^{H1})E(W_a). \tag{61}$$

4.5.3 Group leader (inner node)

Since a group leader is a sender and receiver as well, the bandwidth requirement is the sum of the sender and receiver bandwidth requirements. However, $W_d(1)$ is not considered here, since the initial transmission is sent using the multicast routing tree rather than the ACK tree. Furthermore, a group leader may receive additional retransmissions only from $G-2$ group leaders, since its parents group leader and this group leader itself have to be subtracted.

$$W_H^{H1} = \underbrace{\sum_{m=2}^{M^{H1}} W_d(m) + \sum_{k=1}^{\widetilde{L}^{H1}} W_a(k)}_{\text{as sender}} + \underbrace{\sum_{i=1}^{N_g^{H1}} W_d(i) + \sum_{j=1}^{N_r^{H1}} +W_a(j)}_{\text{as receiver}}$$

$$E(N_g^{H1}) = E(M^{H1})(1-q_D) + (G-2)E(M^{H1})(1-q_D)p_l \tag{62}$$

$$\begin{aligned} E(W_H^{H1}) &= (E(M^{H1})-1)E(W_d) + E(\widetilde{L}^{H1})E(W_a) \\ &\quad + E(N_g^{H1})E(W_d) + E(N_r^{H1})E(W_a) \qquad (63) \\ &= E(W_S^{H1}) + E(W_R^{H1}) \\ &\quad - E(W_d(1)) - E(M^{H1})(1-q_D)p_l E(W_d). \qquad (64) \end{aligned}$$

The maximum throughput rates Λ_S^{H1}, Λ_R^{H1}, Λ_H^{H1} for the sender, receiver and group leader are:

$$\Lambda_S^{H1} = \frac{1}{E(W_S^{H1})}, \quad \Lambda_R^{H1} = \frac{1}{E(W_R^{H1})}, \quad \Lambda_H^{H1} = \frac{1}{E(W_H^{H1})}. \tag{65}$$

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

$$\Lambda^{H1} = min\{\Lambda_S^{H1}, \Lambda_H^{H1}, \Lambda_R^{H1}\}. \tag{66}$$

The total bandwidth consumption of protocol (H1) is then the sum of the sender's, leaf node receivers' and group leaders' bandwidth consumptions:

$$W^{H1} = W_S^{H1} + (R-G+1)W_R^{H1} + (G-1)W_H^{H1}. \tag{67}$$

4.6 Tree-Based Protocol (H2)

(H2) uses selective periodical ACKs (SAKs) and NAKs with NAK avoidance. The sender and group leaders collect all NAKs belonging to one round and send a retransmission if the waiting time has expired and at least one NAK has been received. We have to distinguish between the number of rounds and the number of transmissions. The number of rounds is equal or greater than the number of retransmissions, since if a sender or receiver receives no NAK within one round, no retransmission is invoked.

A SAK is sent by the receiver to announce its state, i.e. its received and missed packets, after a sequence of data packets have been received. We assume that a SAK is sent after a certain period of time. Therefore, when analyzing the processing requirements for a *single* packet, only the proportionate requirements for sending and receiving a SAK (W_Φ) is considered. S is assumed to be the number of SAKs received by the sender in the presence of possible SAK loss: $E(S) = (1-p_A)B$.

4.6.1 Sender (root node)

$$W_S^{H2} = \sum_{i=1}^{M^{H2}} W_d(i) + \sum_{j=1}^{\widetilde{L}^{H2}} W_n(j) + SW_\Phi \tag{68}$$

$$E(W_S^{H2}) = E(M^{H2})E(W_d) + E(\widetilde{L}^{H2})E(W_n) + E(S)E(W_\Phi) \tag{69}$$

$E(M^{H2})$ is determined analogous to protocol (N2) (B instead of R).

To determine $E(\widetilde{L}^{H2})$ we consider that NAKs are received from the child nodes of this local group as well as may be received from other local groups with probability p_l (see Eq. 33, 34 and 59):

$$E(\widetilde{L}^{H2}) = \vartheta_1(1-q_N) + (G-1)\vartheta_1(1-q_N)p_l \quad (70)$$
$$\vartheta_1 = \sum_{k=1}^{E(M^{H2})} N_k \frac{1}{1-p_k}. \quad (71)$$

N_k, the number of NAKs sent in round k and p_k, the failure probability for empty rounds are obtained analogous to Eq. 35, 39, 40 and 44 of (N2). G, the number of group leaders is obtained analogous to Eq. 57 of (H1).

4.6.2 Receiver (leaf node)

Retransmissions are received mainly from the parent node, but may also be received from other group leaders. Analogous, NAKs are mainly received from other receivers in the same local group but may also be received from receivers in other local groups. The bandwidth requirement for a receiver is analogous to Eq. 47 of protocol (N2):

$$\begin{aligned}
&\underbrace{\begin{aligned}E(W_R^{H2}) &= E(M^{H2})(1-q_D)E(W_d) + E(W_\Phi) \\
&+ P(O_r^{H2} > 1)[E(O_r^{H2}|O_r^{H2} > 1) - 1]\frac{\vartheta_2}{\vartheta_3}E(W_n) \\
&+ [P(O^{H2} > 1)[E(O^{H2}|O^{H2} > 1) - 1]\vartheta_2 \\
&-P(O_r^{H2} > 1)[E(O_r^{H2}|O_r^{H2} > 1) - 1]\frac{\vartheta_2}{\vartheta_3}](1-q_N)E(W_n)\end{aligned}}_{\text{from this local group}} \\
&\underbrace{\begin{aligned}&+ (G-1)p_l E(M^{H2})(1-q_D)E(W_d) \\
&+ (G-1)p_l P(O^{H2} > 1)[E(O^{H2}|O^{H2} > 1) - 1]\vartheta_2 E(W_n)\end{aligned}}_{\text{from other local groups}}. \quad (72)
\end{aligned}$$

ϑ_2 and ϑ_3 can be obtained analogous to Eq. 48 and 49 of protocol (N2) with B instead of R.

4.6.3 Group leader (inner node)

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

$$\begin{aligned}
E(W_H^{H2}) &= E(W_S^{H2}) + E(W_R^{H2}) - E(W_d(1)) \\
&- p_l(E(M^{H2})(1-q_D)E(W_d) \\
&\quad + P(O^{H2} > 1)[E(O^{H2}|O^{H2} > 1) - 1]\vartheta_2 E(W_n)). \quad (73)
\end{aligned}$$

The second and third line in the above equation are the processing requirements for one other local group. They have to be subtracted because in contrast to the sender or receivers, a group leader has a local parent group *and* local child group which are already considered for the normal operations.

Finally, the maximum throughput rates are:

$$\Lambda_S^{H2} = \frac{1}{E(W_S^{H2})}, \quad \Lambda_R^{H2} = \frac{1}{E(W_R^{H2})}, \quad \Lambda_H^{H2} = \frac{1}{E(W_H^{H2})} \quad (74)$$
$$\Lambda^{H2} = min\{\Lambda_S^{H2}, \Lambda_H^{H2}, \Lambda_R^{H2}\}. \quad (75)$$

The total bandwidth consumption of protocol (H2) is:

$$W^{H2} = W_S^{H2} + (R-G+1)W_R^{H2} + (G-1)W_H^{H2}. \quad (76)$$

4.7 Tree-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, i.e. until all nodes have received the data packet. 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. In this case, a retransmission is acknowledged with an AAK rather than an ACK. So, only nodes whose AAK is missing must be queried in phase two.

Table 2: Additional notations for (H3) and (H4)

W_{aa}, W_{aaq}	Bandwidth for an AAK or AAK query packet.
$W_{aa,\phi}, W_{aaq,\phi}$	Proportionate bandwidth for a periodical AAK or AAK query packet, respectively.
$W_{d,u}, W_{d,m}$	Bandwidth to send a data packet per unicast or multicast, respectively.
p_q	Probability for AAK query loss.
p_{AA}	Probability of a unicast AAK loss.
n_k	Current number of receivers that need a retransmission.
ϕ	Threshold for unicast retransmission. If n_k is smaller than ϕ, unicast is used for retransmission and multicast otherwise.
p_t	Probability that n_k is smaller than the threshold ϕ for multicast retransmissions and therefore unicast is used.
τ	Probability that a retransmission is necessary due to data or ACK loss.
$\hat{p}$	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, respectively.
L_a^w, L_{aa}^w	Number of sent ACKs or AAKs.
$\widetilde{L}_a^w, \widetilde{L}_{aa}^w$	Number of received ACKs or AAKs.
L_{aaq}^w	Number of sent AAK queries.
$\widetilde{L}_{aaq}^w$	Number of received AAK queries.
B_{aa}	Number of receivers in a local group 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.

4.7.1 Sender (root node)

The bandwidth requirement of a sender is:

$$\begin{aligned}
W_S^{H3} &= \sum_{j=1}^{M_m^{H3}} W_{d,m}(j) + \sum_{k=1}^{M_u^{H3}} N_u W_{d,u}(k) \\
&+ \sum_{i=1}^{\widetilde{L}_a^{H3}} W_a(i) + \sum_{w=1}^{L_{aaq}^{H3}} W_{aaq}(w) + \sum_{z=1}^{\widetilde{L}_{aa}^{H3}} W_{aa}(z). \quad (77)
\end{aligned}$$

M_m^{H3} and M_u^{H3} are the number of necessary multicast or unicast transmissions, respectively. $W_{d,m}$ and $W_{d,u}$ determine the bandwidth requirements for a multicast or unicast packet transmission. W_{aa} is the necessary bandwidth for an AAK and $\widetilde{L}_{aa}^{H3}$ 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 (W_{aaq}) 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. L_{aaq}^{H3} is the number of necessary unicast AAK queries in the presence of message loss.

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

$$M_u^{H3} = p_t(M^{H3} - 1) \quad (78)$$
$$M_m^{H3} = (1-p_t)(M^{H3} - 1) + 1. \quad (79)$$

Please note that the first transmission is always sent with multicast. The probability for a retransmission due to data or ACK loss is given by:

$$\tau = \underbrace{\overbrace{p_t p_D}^{\text{unicast}} + \overbrace{(1-p_t)q_D}^{\text{multicast}}}_{\text{data loss}} + \underbrace{[1 - (\overbrace{p_t p_D}^{\text{unicast}} + \overbrace{(1-p_t)q_D}^{\text{multicast}})] p_A}_{\text{no data loss but ACK loss}}. \quad (80)$$

$E(M^{H3})$ is determined by τ instead of $\tilde{p}$ and B instead of R analogous to Eq. 8 of protocol (A1). 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:

$$p_t = \frac{1}{M^{H3}} \sum_{k=1}^{M^{H3}} \begin{cases} 1 , n_k < \phi \\ 0 , n_k \geq \phi \end{cases} \quad (81)$$

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:

$$N_u = \frac{1}{M_u^{H3}} \sum_{k=1}^{M^{H3}} \begin{cases} n_k , n_k < \phi \\ 0 \ \ , n_k \geq \phi \end{cases} \quad (82)$$

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

$$E(N_r^{H3}) = \frac{N_u}{B} E(M_u^{H3})(1-p_D) + E(M_m^{H3})(1-q_D). \quad (83)$$

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

$$E(\widetilde{L}_a^{H3}) = B E(N_r^{H3})(1-p_A)p_c. \quad (84)$$

p_c is the probability that no AAK can be sent due to missing AAKs of child nodes. The number of AAK query rounds L_1, is determined by the probability $\hat{p}$ that a query fails:

$$\hat{p} = p_q + (1-p_q)p_{AA}. \quad (85)$$

$E(L_1)$ can be determined analogous to M^{A1} of protocol (A1) (see Eq. 8) with B_{aa} instead of R 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:

$$E(L_1) = \sum_{i=1}^{B_{aa}} \binom{B_{aa}}{i} (-1)^{i+1} \frac{1}{1-\hat{p}^i} \quad (86)$$

$$B_{aa} = B\left(p_c + (1-p_c)p_{AA}\right)^{E(N_r^{H3})}. \quad (87)$$

$p_c + (1-p_c)p_{AA}$ 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:

$$E(L_{aaq}^{H3}) = \textstyle\sum_{k=1}^{E(L_1)} B_{aa}\hat{p}^{(k-1)}. \quad (88)$$

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. 84).

$$E(\widetilde{L}_{aa}^{H3}) = B E(N_r^{H3})(1-p_{AA})(1-p_c) + B_{aa}. \quad (89)$$

Now, $E(W_S^{H3})$ is entirely determined by:

$$\begin{aligned} E(W_S^{H3}) = {} & E(M_u^{H3}) N_u E(W_{d,u}) + E(M_m^{H3}) E(W_{d,m}) \\ & + E(\widetilde{L}_a^{H3}) E(W_a) + E(L_{aaq}^{H3}) E(W_{aaq}) \\ & + E(\widetilde{L}_{aa}^{H3}) E(W_{aa}). \end{aligned} \quad (90)$$

4.7.2 Receiver (leaf node)

The bandwidth requirement at the receiver is given by:

$$\begin{aligned} W_R^{H3} = {} & \sum_{i=1}^{N_{r,t}^{H3}} W_d(i) + \sum_{j=1}^{L_a^{H3}} W_a(j) \\ & + \sum_{k=1}^{L_{aa}^{H3}} W_{aa}(k) + \sum_{l=1}^{\widetilde{L}_{aaq}^{H3}} (W_{aa}(l) + W_{aaq}(l)). \end{aligned} \quad (91)$$

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

$$E(N_{r,t}^{H3}) = E(N_r^{H3}) + (G-1)E(N_r^{H3})p_l. \quad (92)$$

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

$$L_a^{H3} = p_c E(N_r^{H3}) \quad (93)$$

$$L_{aa}^{H3} = (1-p_c)E(N_r^{H3}). \quad (94)$$

Here we assume that only transmissions from this local group are acknowledged. $\widetilde{L}_{aaq}^{H3}$, the number of AAK queries received by an receiver are:

$$\widetilde{L}_{aaq}^{H3} = \frac{1}{B_{aa}} E(L_{aaq}^{H3})(1-p_q), \quad (95)$$

where $1/B_{aa}$ is the probability to be a receiver that gets an AAK query. Finally, the expectation for a receiver's bandwidth requirements is:

$$\begin{aligned} E(W_R^{H3}) = {} & E(N_{r,t}^{H3})E(W_d) + E(L_a^{H3})E(W_a) + E(L_{aa}^{H3})E(W_{aa}) \\ & + E(\widetilde{L}_{aaq}^{H3})(E(W_{aa}) + E(W_{aaq})). \end{aligned} \quad (96)$$

4.7.3 Group leader (inner node)

The bandwidth requirement at a group leader consists of the sender and receiver bandwidth requirements (see Eq. 64):

$$\begin{aligned} E(W_H^{H3}) = {} & E(W_S^{H3}) + E(W_R^{H3}) \\ & - E(W_{d,m}(1)) - E(N_r^{H3})p_l E(W_d). \end{aligned} \quad (97)$$

Finally, the maximum throughput rates are:

$$\Lambda_S^{H3} = \frac{1}{E(W_S^{H3})}, \quad \Lambda_R^{H3} = \frac{1}{E(W_R^{H3})}, \quad \Lambda_H^{H3} = \frac{1}{E(W_H^{H3})} \quad (98)$$

$$\Lambda^{H3} = min\{\Lambda_S^{H3}, \Lambda_H^{H3}, \Lambda_R^{H3}\}. \quad (99)$$

The total bandwidth consumption of protocol (H3) is:

$$W^{H3} = W_S^{H3} + (R-G+1)W_R^{H3} + (G-1)W_H^{H3}. \quad (100)$$

4.8 Tree-Based Protocol (H4)

The generic definition of protocol class (H4) is given in Section 3.3. As in (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 is 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 per 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 and receiving AAKs ($W_{aa,\phi}$) as well as the costs for querying AAKs ($W_{aaq,\phi}$) can be set to a proportionate cost of the other costs.

4.8.1 Sender (root node)

$$W_S^{H4} = \sum_{i=1}^{M^{H4}} W_d(i) + \sum_{j=1}^{\tilde{L}^{H4}} W_n(j) + \sum_{w=1}^{L_{aaq}^{H4}} W_{aaq,\phi}(w) + \sum_{z=1}^{\tilde{L}_{aa}^{H4}} W_{aa,\phi}(z) \quad (101)$$

$$E(W_S^{H4}) = E(M^{H4})E(W_d) + E(\tilde{L}^{H4})E(W_n) + E(L_{aaq}^{H4})E(W_{aaq,\phi}) + E(\tilde{L}_{aa}^{H4})E(W_{aa,\phi}) \quad (102)$$

$E(M^{H4})$ and $E(\tilde{L}^{H4})$ are determined analogous to protocol (H2). The number of AAK queries is determined by the probability $\hat{p}$ that a query fails:

$$\hat{p} = p_q + (1 - p_q)p_{AA}. \quad (103)$$

The number of query rounds $E(L_1)$ can be determined analogous to M^{A1} of protocol (A1) (see Eq. 8) with B_{aa} instead of R and $\hat{p}$ instead of $\tilde{p}$. 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 successfully receptions, the number of nodes to query in phase 2 is the number of lost AAKs.

$$E(L_1) = \sum_{i=1}^{B_{aa}} \binom{B_{aa}}{i}(-1)^{i+1}\frac{1}{1-\hat{p}^i} \quad (104)$$

$$B_{aa} = Bp_{AA}. \quad (105)$$

The total number of unicast query messages in all rounds are:

$$E(L_{aaq}^{H4}) = \sum_{k=1}^{E(L_1)} B_{aa}\hat{p}^{(k-1)}. \quad (106)$$

Using unicast, only those nodes are queried whose AAK is missing. So finally, the number of received AAKs at the sender is equal to the number of child nodes in the ACK tree:

$$E(\tilde{L}_{aa}^{H4}) = B. \quad (107)$$

4.8.2 Receiver (leaf node)

$$\begin{aligned} E(W_R^{H4}) = {} & E(M^{H4})(1-q_D)E(W_d) \\ & \underbrace{\begin{aligned} & + P(O_r^{H4} > 1)[E(O_r^{H4}|O_r^{H4} > 1) - 1]\frac{\vartheta_2}{\vartheta_3}E(W_n) \\ & + E(W_{aa,\phi}) + E(\tilde{L}_{aaq}^{H4})(E(W_{aaq,\phi}) + E(W_{aa,\phi})) \\ & + [P(O^{H4} > 1)[E(O^{H4}|O^{H4} > 1) - 1]\vartheta_2 \\ & -P(O_r^{H4} > 1)[E(O_r^{H4}|O_r^{H4} > 1) - 1]\frac{\vartheta_2}{\vartheta_3}](1-q_N)E(W_n) \end{aligned}}_{\text{from this local group}} \\ & \underbrace{\begin{aligned} & + (G-1)p_l E(M^{H4})(1-q_D)E(W_d) \\ & + (G-1)p_l P(O^{H4} > 1)[E(O^{H4}|O^{H4} > 1) - 1]\vartheta_2 E(W_n) \end{aligned}}_{\text{from other local groups}} \end{aligned} \quad (108)$$

ϑ_2 and ϑ_3 can be obtained analogous to (N2) with B instead of R. $E(\tilde{L}_{aaq}^{H4})$, the number of received AAK queries and replied AAKs is (see Eq. 95):

$$\tilde{L}_{aaq}^{H4} = \frac{1}{B_{aa}}E(L_{aaq}^{H4})(1-p_q), \quad (109)$$

and the number of rounds O^{H4} is determined analogous to (N2).

4.8.3 Group leader (inner node)

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

$$\begin{aligned} E(W_H^{H4}) = {} & E(W_S^{H4}) + E(W_R^{H4}) - E(W_d(1)) \\ & - p_l(E(M^{H4})(1-q_D)E(W_d) \\ & + P(O^{H4} > 1)[E(O^{H4}|O^{H4} > 1) - 1]\vartheta_2 E(W_n)). \end{aligned} \quad (110)$$

Finally, the maximum throughput rates at the sender, receiver, group leader and overall throughput are:

$$\Lambda_S^{H4} = \frac{1}{E(W_S^{H4})}, \quad \Lambda_R^{H4} = \frac{1}{E(W_R^{H4})}, \quad \Lambda_H^{H4} = \frac{1}{E(W_H^{H4})} \quad (111)$$

$$\Lambda^{H4} = min\{\Lambda_S^{H4}, \Lambda_H^{H4}, \Lambda_R^{H4}\}. \quad (112)$$

The total bandwidth consumption of protocol (H4) is:

$$W^{H4} = W_S^{H4} + (R - G + 1)W_R^{H4} + (G-1)W_H^{H4}. \quad (113)$$

5. NUMERICAL RESULTS

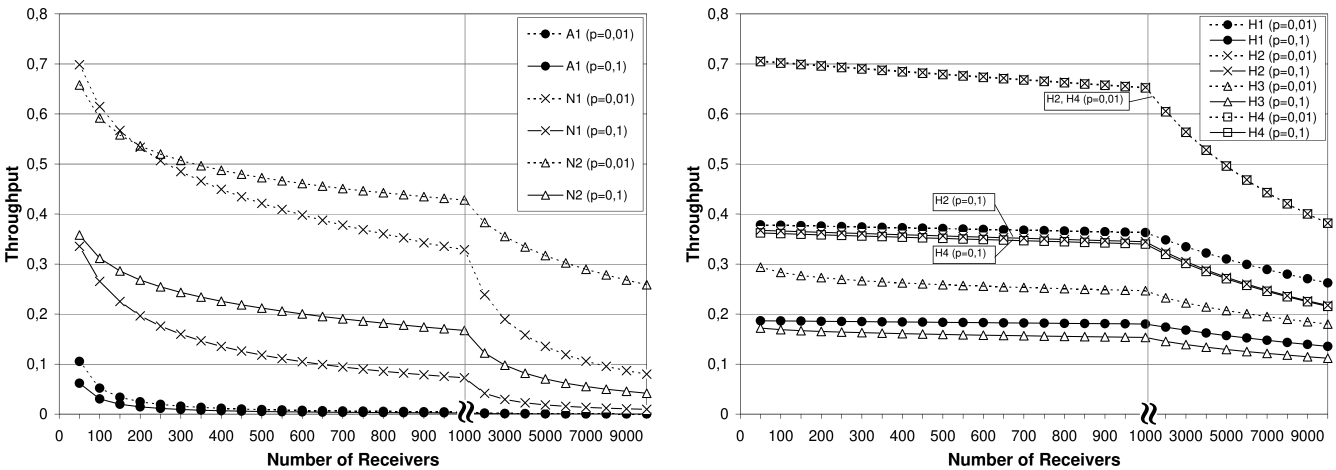
We examine the relative performance and bandwidth consumption of the analyzed protocols by means of some numerical examples. The mean bandwidth costs are set equal to 1 for data packets (W_d, $W_{d,u}$, $W_{d,m}$), 0.1 for control packets (W_a, W_n, W_{aa}, W_{aaq}) and 0.01 for periodical control packets (W_ϕ, $W_{aa,\phi}$ and $W_{aaq,\phi}$). The following graphs show the throughput of the various protocol classes relative to the normalized maximum throughput of 1.

Figure 1.a shows the bandwidth limited maximum throughput of the sender-initiated protocol (A1) and the receiver-initiated protocols (N1) and (N2). The loss probability for data packets as well as control packets is 0.01 for the dotted lines and 0.1 for the solid ones. The probability for simultaneous NAK sending in (N2) is set to 0.1.

The results in Figure 1.a show, that a protocol based on positive acknowledgments like (A1) is not applicable for large receiver groups, since the large number of ACKs overwhelms the sender. The performance of (N1) and (N2) is much better than (A1)'s performance. Particularly, if packet loss probability is low, only few NAK messages are returned to the sender which improves the performance. (N2) with NAK avoidance scheme provides the best performance of all non-hierarchical approaches.

In Figure 1.b, the results for the hierarchical protocol classes (H1), (H2), (H3) and (H4) are shown. The number of child nodes is set equal to 10 for all classes and the probability for receiving packets from other local groups, p_l, is set equal to 0.001. (H3) is shown with $\phi = 0$ which corresponds with protocol (H1) except for the additional aggregated ACKs of (H3). $\phi = 0$ means that all retransmission are sent with multicast.

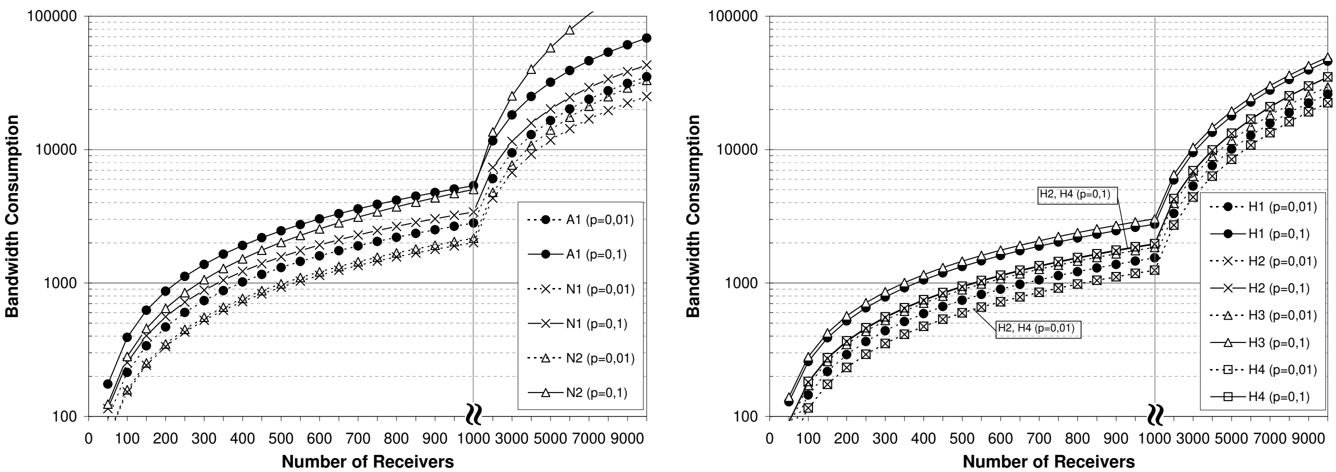
All protocol classes experience a throughput degradation with increasing group sizes although the local group size



(a) sender and receiver-initiated protocols

(b) tree-based protocols

Figure 1: Throughput of reliable multicast protocols



(a) sender and receiver-initiated protocols

(b) tree-based protocols

Figure 2: Bandwidth consumption of reliable multicast protocols

remains constant. This results from our assumption that a packet is received with probability $p_l = 0.001$ outside the scope of a local group. With increasing number of receivers, the number of groups increase also and therefore more packets from other local groups are received. Note that if each local group is assigned a separate multicast address for retransmissions, no packets from other local groups are received and therefore p_l has to be set equal to 0. In this case, the throughput of all hierarchical approaches remains constant.

The protocols with negative acknowledgments and NAK avoidance provide again the best performance. As it can be further seen in the figure, the additional overhead for periodical aggregated acknowledgments is very low, therefore (H4) provides almost the same performance as (H2). In case of (H3), the aggregated acknowledgments are sent after every correct message transmission. Therefore, the performance reduction compared to (H1) is more significant than between (H4) and (H2). If (H3)'s aggregated ACKs are also sent periodically as in (H4), the performance would be almost the same as (H1)'s performance. This means that the additional costs for providing reliability even in the presence of node failures are small and therefore acceptable for protocol implementations.

Due to readability, the result for (H3) with $\phi = 2$ is not shown in the figure. With $\phi = 2$, only retransmissions for equal or more than 2 nodes are made using multicast and with unicast otherwise. In this case, (H3) provides better performance especially for a large number of receivers. For a packet loss probability of 0.1 the performance is equal to (H1).

By comparing Figure 1.a and 1.b it can be seen that tree-based protocols are superior. Their throughput degradation with increasing multicast group size is much smaller compared to non-hierarchical approaches. Furthermore, they are more robust against high packet loss probabilities.

The following figure depicts the bandwidth consumption of all analyzed protocols. Figure 2.a shows that the bandwidth consumption of (N2) for small group sizes is below (A1)'s requirements. However, for large group sizes and higher loss probability, (N2)'s bandwidth consumption is 2,5 times the bandwidth consumption of (A1). (N1) provides the lowest bandwidth consumption of the three classes.

In Figure 2.b it can be seen that tree-based protocols require less overall bandwidth than non-hierarchical approaches. In fact, tree-based protocols save in most cases about 50% of the bandwidth costs and compared to (N2) up to 85% (please note the logarithmic y-axis). The results for (H3) with $\phi = 2$ are not shown in the figure, which are in any case lower compared to (H3) with $\phi = 0$. In case of a packet loss probability of 0.01, (H3) with $\phi = 2$ requires only the bandwidth of (H1). If the packet loss probability is higher than 0.01, the bandwidth consumption is even smaller than that of (H1). For example, with packet loss probability 0.1, (H3)'s bandwidth consumption with $\phi = 2$ is more than 30% below (H1)'s requirements. In contrast to non-hierarchical approaches, i.e. if local group sizes are small, NAK with NAK avoidance protocols require low bandwidth costs. Therefore, protocols (H2) and (H4) provide the lowest bandwidth consumption.

6. SUMMARY

We have analyzed the throughput in terms of bandwidth requirements and the overall bandwidth consumption of sender-initiated, receiver-initiated and tree-based multicast protocols assuming a realistic system model with data packet loss, control packet loss and asynchronous clocks. Of particular importance are the analyzed protocol classes with aggregated acknowledgments. In contrast to other hierarchical approaches, these classes provide reliability even in the presence of node failures.

The results of our numerical examples show that hierarchical approaches are superior. They provide higher throughput and lower overall bandwidth consumption compared to sender-initiated or receiver-initiated protocols. The protocol classes with aggregated acknowledgments lead to only a small throughput decrease and slightly increased overall bandwidth consumption compared to the same classes without aggregated acknowledgments. This means, that the additional costs for providing a reliable multicast service even in the presence of node failures are small and therefore acceptable for reliable multicast protocol implementations.

7. REFERENCES

[1] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6):784–803, Dec. 1997.

[2] 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*, pages 988–995, New York, Apr. 1998. IEEE.

[3] B. Levine and J. Garcia-Luna-Aceves. A comparison of reliable multicast protocols. *ACM Multimedia Systems*, 6(5):334–348, Sept. 1998.

[4] B. Levine, D. Lavo, and J. Garcia-Luna-Aceves. The case for reliable concurrent multicasting using shared ack trees. In *Proceedings of the Fourth ACM Multimedia Conference (MULTIMEDIA'96)*, pages 365–376, New York, Nov. 1996. ACM Press.

[5] 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*, New York, Oct. 2000. IEEE.

[6] 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*, pages 972–979, New York, Apr. 1998. IEEE.

[7] 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*, 15(3):407–421, Apr. 1997.

[8] 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*, pages 221–230, New York, May 1994. ACM Press.

[9] G. Poo and A. Goscinski. Performance comparison of sender-based and receiver-based reliable multicast protocols. *Computer Communications*, 21(7):597–605, June 1998.

[10] 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*, pages 139–145, New York, Oct. 1999. IEEE.

[11] T. Speakman, D. Farinacci, S. Lin, and A. Tweedly. PGM reliable transport protocol specification. Internet draft <draft-speakman-pgm-spec-04.txt>, work in progress, 2000.

[12] T. W. Strayer, B. J. Dempsey, and A. C. Weaver. *XTP – The Xpress transfer protocol*. Addison-Wesley Publishing Company, 1992.

[13] B. Whetten and G. Taskale. An overview of the reliable multicast transport protocol II. *IEEE Network*, 14(1):37–47, Feb. 2000.

[14] 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)*, pages 333–344, New York, Nov. 1996. ACM Press.