Computer Communications 32 (2009) 668–678

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

End-to-end epidemic multicast loss recovery: Analysis of scalability and robustness

Öznur Özkasap *

Department of Computer Engineering, Koç University, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 30 March 2008
Received in revised form 21 November 2008
Accepted 22 November 2008
Available online 7 December 2008

Keywords:
Multicast loss recovery
Epidemic communication
Scalable multicast
End-to-end protocols
Robustness

ABSTRACT

For ensuring reliability at the transport level end-to-end multicasting, an efficient loss recovery mechanism is indispensable. We consider scalability, topology independence and robustness as the significant features that such a mechanism should offer, and demonstrate that an epidemic loss recovery approach is superior in all these aspects. We also show that the epidemic approach transparently handles network link failures by using pair-wise propagation of information, and compare it with feedback controlled loss recovery on identical network settings. The contribution of this work is the simulative analysis of recovery overhead distribution on multicast group members in the case of various link failures on the network, the impact of group size, randomized system-wide noise and message rate on scalability, and examination of various scenarios modeling the overlay networks. We investigate the important features of epidemic multicast loss recovery extensively together and reach concrete results on realistic network scenarios.

1. Introduction

At the transport level of the network architecture, end-to-end protocols support communication between the end application processes. There are two major forces that shape an end-to-end protocol. From the level above, the application processes that use protocol services have certain needs. Some of the common properties that a transport protocol can offer are message loss recovery that guarantees reliable message delivery, ordered message delivery, preventing message duplication, support for arbitrarily large messages transmitted by the application, support for flow control, and provision of multiple application processes at each host. At the other end, from the level below, the underlying network has certain constraints in the level of service it can provide. Typical constraints of the network are dropping messages, reordering messages, delivering duplicate copies of a given message, limiting messages to a finite size, delivering messages after an arbitrarily long delay. Such an underlying network structure is referred to as a best-effort level of service as exemplified by the Internet. The challenge of an end-to-end protocol is, therefore, to provide algorithms that turn these best-effort properties of the underlying network into the high level of service required by applications.

Increasing popularity of group-based applications in large-scale settings given the varying quality of service requirements stipulates efficient multicast communication mechanisms. Multicast paradigm matches well with several Internet group services such as multimedia, videoconferencing, distributed computation, data dissemination, database and real-time workgroups. When a multicast source disseminates information to many participants in such services, network resources would be utilized optimally because of the nature of multicasting. In particular, the main benefits are reducing overhead in source, bandwidth consumption in network, and latency seen by participants.

As depicted in Fig. 1, within end-to-end protocols, transport level multicast requirements can be broadly classified as *loss-sensitive reliable* services and *delay-sensitive interactive* services. While interactive applications such as multimedia conferencing can tolerate reliability in support of real-time delivery, data dissemination applications such as multicast file transport tolerate longer transfer delays. Several large-scale distributed applications exploiting multicast communication require reliable delivery of data to all participants. In addition, scalability, throughput stability, efficient loss recovery and buffer management are essential communication properties in large-scale settings. In particular, the degree of reliability guarantees required by multicast based applications differs from one setting to another. Thus, reliability guarantees provided by multicast communication protocols split them into three broad classes. One class of protocols offers *strong reliability* guarantees such as atomicity, delivery ordering, virtual synchrony, real-time support, security properties and network-partitioning support. These protocols allow limited scalability. The main drawback is that in order to obtain strong reliability guarantees, costly protocols are used which makes them unsuitable for large groups. The other class offers support for *best-effort reliability* in large-scale settings. Although protocols offering support for best-effort reliability in large-scale overcome message loss and failures, they do not guarantee end-to-end reliability. Common

* Tel.: +90 212 338 1584.
E-mail address: oozkasap@ku.edu.tr

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.11.033

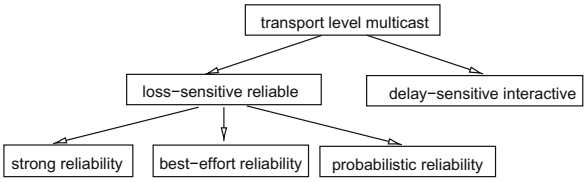

Fig. 1. Classification of transport level multicast requirements.

failure scenarios such as router overload and system-wide link noise can cause these protocols to behave pathologically [32] and hence lead to negative protocol effects on network performance. Within this context, example transport level reliable multicast protocols are the Internet Muse protocol [20] for network news distribution, scalable reliable multicast (SRM) [11], and pragmatic general multicast (PGM) [13]. Besides, there exist overlay multicast protocols for reliable data dissemination such as SplitStream [7], Scribe [8] and OverCast [18]. We examine their loss recovery properties in the next section.

In the spectrum of scalable reliable multicast protocols, *probabilistic reliability* protocols like Bimodal Multicast [3] provide weaker guarantees compared to other classes of multicast protocols with strong reliability guarantees. A probabilistically reliable multicast protocol is suitable for applications that are insensitive to small inconsistencies among participants. On the other hand, these protocols offer quality of service properties such as throughput stability, scalability and minimal delivery latency of multicast messages that are essential for some distributed applications. Throughput stability guarantee provides the steady delivery of multicast data stream to correct participants [3,26].

In this study, we focus on the epidemic loss recovery characteristics of the probabilistic protocols mentioned above. Our previous related work [29] has examined the inverted protocol stack approach of Bimodal Multicast, in which probabilistic mechanisms are used at low layers, and reliability properties introduced closer to the application. We have demonstrated that the inverted protocol stack approach works well on several network settings, and compared it with best-effort reliable multicast mechanisms. Analysis has mainly focused on interarrival and latency distributions, and network topologies of up to hundred nodes were investigated in the analysis. However, large-scale behavior as well as topology independence were not examined.

Our recent study [30] has concentrated on a different aspect, namely the traffic characterization of transport level reliable multicasting, and has shown that self-similarity is protocol dependent. Traffic at the link level was analyzed and it has been demonstrated that Markovian character of epidemic loss recovery distinguishes an inherently superior protocol. It discretely feeds well-behaved traffic and copes with the existing self-similarity. On the other hand, feedback controlled loss recovery mechanisms have been shown to trigger self-similarity.

In contrast to prior work, we focus on the scalability, topology independence and robustness aspects of end-to-end epidemic loss recovery and analyze these properties via extensive simulations. To the best of our knowledge, this is the first study investigating these important aspects extensively together and reaching concrete results on realistic network settings. We also compare the epidemic recovery with the nonhierarchical feedback control mechanism on various network scenarios, and provide a discussion of forward error correction approaches for scalable reliable multicast. Our contribution is the simulative analysis of recovery overhead distribution on multicast group members when there exists various link failures on the network; the impact of group size, randomized system-wide noise and message rate on scalability, and various topologies modeling the wide-area overlay networks. We show that load of recovery overhead is balanced among multicast group members, and it does not increase linearly with link failures. These features are also found to be independent from the underlying network topology, scalable and robust in providing full reliability. Furthermore, epidemic loss recovery is shown to outperform the feedback control mechanism. Our initial results examining robustness property and large-scale behavior were reported in [28,27], respectively.

Applications such as teleconferencing over the Internet, electronic stock exchange/trading, and health-care systems, can benefit from the properties provided by probabilistic multicast protocols as discussed in [3,26]. Such applications need to be scalable, and they must tolerate some data inconsistencies that may occur among the participants as long as these events are not frequent. For instance, in electronic stock exchange and trading environments, the trading information including orders and trades is multicast immediately to all members ensuring equal treatment and market transparency. A multicast protocol is used to disseminate trading information to all members at the same time and with minimal delay. In health-care systems, as another application area, patient telemetry data are refreshed frequently on monitors located in places such as the patient's room, nursing station, and physician's office. Since infrequent loss of data of this sort is tolerable, they can be transmitted using probabilistic protocols. On the other hand, some data types, like medication change order, still need strong end-to-end guarantees.

The article is organized as follows. Section 2 reviews message loss recovery techniques in scalable reliable multicasting. In Section 3, we describe pull-based anti-entropy epidemic model and its usage for loss recovery in probabilistic reliable multicast protocols. Section 4 presents protocol state diagram, simulation topologies and settings. Section 5 details out the extensive analysis results. Finally, Section 6 concludes and discusses future work.

2. A review of loss recovery approaches in scalable multicast

Loss recovery is a key mechanism of a multicast service offering reliability. Our focus in this study is on scalable multicast services that attempt to ensure probabilistic reliable message delivery. In the case of large-scale multicast applications, a *sender-initiated* approach, in which it is the responsibility of the message source to detect losses, may cause feedback implosion, since each delivered message initiates an ACK from every group member. This, in turn, leads to overflow on the sender's buffer and congests the network. An alternative is the *receiver-initiated* approach in which receivers request retransmissions by generating negative acknowledgments (NACKs) upon detecting message losses. Performance comparison studies confirm that receiver-initiated multicast transport protocols outperforms their sender-initiated counterparts in terms of scalability. However, a problem with returning only NACKs is that the sender would need to keep messages in its buffer for a long time. Most reliable multicast transport protocols are either pure receiver-initiated or a hybrid of sender and receiver-initiated approaches.

We classify the key approaches that are representatives of several solutions for providing loss recovery in scalable multicasting into the following groups:

- feedback control
- forward error correction codes
- overlay multicast
- epidemic recovery

In the *feedback control* approaches, the key issue is to reduce the number of feedback messages that are returned to the sender. A model adopted by several wide-area applications and protocols

such as SRM [11] is referred to as feedback suppression. In SRM, when a receiver detects a missing message, it multicasts its feedback to the rest of the group. Multicasting feedback allows another group member to suppress its own feedback. A receiver lacking a message schedules a feedback with some random delay. An improvement to enhance scalability is referred to as local recovery, which is related to restraining the recovery of a message loss to the region where the loss has occurred.

Hierarchical feedback control mechanisms such as RMTP [31] and RRMP [43] protocols are adopted for achieving scalability for very large groups of receivers. In RMTP, multicast receivers are grouped into local regions and each region has a leader member responsible for feedback processing and retransmissions for its region. Such tree-based hierarchical approaches reduce the protocol overhead by feedback suppression at regions and retransmission delays. The randomized reliable multicast protocol, RRMP, offers efficient loss recovery for large multicast groups. By distributing the responsibility of loss recovery among all group members, it aims to improve the efficiency and robustness of tree-based protocols. In this approach, the drawback of failures of group leaders in tree-based hierarchical approaches is prevented as well.

Forward error correction (FEC) [25,37] is another key mechanism for providing reliability in scalable multicast, in which the idea is predicting losses and transmitting redundant data. For a (n,d) FEC code, d blocks of data messages are encoded into n blocks. That is, the sender includes redundant data of $(n-d)$ blocks for the transmission of d data blocks. A receiver can reconstruct d data blocks given any d blocks correctly received out of n blocks transmitted by the sender. In this way, the receiver side can use the redundant data for loss recovery without asking for retransmissions from the sender. A benefit of FEC-based solutions is that the retransmissions of data is not required in case of packet losses. Thus, FEC is preferable in scenarios where retransmission mechanisms are costly or not possible. With large multicast groups, message losses at different receivers become independent, and efficiency of retransmission-based solutions is degraded. Another benefit with FEC is that different loss patterns at different receivers can be recovered using the same set of transmitted data. A drawback of FEC could be the higher bandwidth consumption due to redundant data transmission.

PGM [13] is a reliable multicast protocol using a hybrid mechanism, namely FEC with a hierarchical approach and NAK suppression, to achieve scalability. It supports single-source multicast applications. It runs over a best-effort datagram protocol like IP multicast, but also needs router support for constructing hierarchy. Another approach in the category of FEC-based solutions, namely digital fountain [5], supports asynchronous reliable multicast for a group of heterogeneous participants. A digital fountain offers participants to retrieve content on their time of demand that is asynchronously. FEC-based erasure codes are used to implement the reliable multicast protocol based on digital fountain approach. In erasure codes, data encoded with the redundant packets are transmitted by the source. Any subset of these encoded packets with length equal to the length of the original data would be enough to reconstruct the data. Thus, in case of data loss at a receiver, redundant packets are used for loss recovery. Since these packets can be utilized by several receivers, this approach can significantly decrease the amount of retransmissions.

In the category of scalable *overlay multicast* protocols offering reliable data dissemination, well-known approaches are SplitStream [7], Scribe [8] and OverCast [18]. SplitStream [7] addresses the problem of distributing the forwarding load of traditional tree-based overlay multicast among the participating peers evenly. It is a high-bandwidth content distribution system based on end-system multicast, robust to node failures, and can manage peers with different bandwidth capacities. The content to be distributed is striped across multiple multicast trees. This fact increases the resilience to node failures. With suitable data encoding methods such as erasure coding and multiple description coding, applications can achieve data loss recovery in the case of node failures. SplitStream is built using Pastry [36], which is a scalable, self-organizing structured P2P overlay network similar to Chord [39]. Scribe [8] is a scalable, self-organizing and fully decentralized overlay multicast approach that offers best-effort reliability. It is built on top of Pastry and uses it for managing group creation, join, building multicast tree and repairing it in the case of a node failure. It uses TCP for reliable delivery of data in the multicast tree and for flow control. The load on participants is balanced, and in comparison to network layer multicast, delay and link stress achieved is acceptable. OverCast [18] is a tree-based scalable reliable overlay multicast protocol. By building bandwidth efficient and scalable distribution trees, the protocol is adaptive to changes in network conditions and provides fast peer joins. OverCast is a single-source multicast protocol and it utilizes replication of data on the overlay network to minimize bandwidth requirements. A survey and detailed classification of application layer overlay multicast protocols can be found in [17].

On the other hand, *epidemic* or randomized approaches to loss recovery have promising outcomes in terms of robustness and overhead. In this direction, Bimodal Multicast provides an epidemic loss recovery mechanism. In the next section, we describe pull-based anti-entropy as an epidemic mechanism for multicast loss recovery and its utilization in the context of Bimodal Multicast.

3. Pull-based anti-entropy for scalable multicast reliability

Epidemic protocols are simple, scale well and robust against common failures, and provide eventual consistency as well. They are based on the theory of epidemics which studies the spreading of infectious diseases through a population. They combine benefits of efficiency in hierarchical data dissemination with robustness in flooding protocols. Epidemic communication allows temporary inconsistencies in shared data among participants, in exchange for low-overhead implementation. Information changes are spread throughout the participants without incurring the latency and bursty communication that are typical for systems achieving a strong form of consistency. In fact, this is especially important for large systems, where failure is common, communication latency is high and applications may contain hundreds or thousands of participants. Epidemic or gossip style of communication algorithms have been used for several purposes [10]. Examples include large-scale direct mail systems [4], group membership tracking [14], support for replicated services [19], deciding when a message can be garbage collected [16], failure detection [33], loss recovery in reliable multicast [3,43], and distributed information management [34].

A popular distribution model based on the theory of epidemics is the anti-entropy [1]. According to the terminology of epidemiology, a site holding information or an update it is willing to share is called *infectious*. A site is called *susceptible* if it has not yet received an update. In the anti-entropy process, non-faulty sites are always either susceptible or infective. In this model, a site P picks another site Q at random, and exchanges updates with Q. For exchanging updates, there are three approaches:

- *Push approach: P* only propagates its updates to *Q*. In this case, updates can be propagated only by infectious sites. If many sites are infectious, the probability of each one selecting a susceptible site is relatively small. Since the update propagation is triggered

by an infectious site picking a susceptible one, a particular site may remain susceptible for a long period simply because it is not selected by an infectious site. Hence, this property may limit the speed of spreading updates.

- *Pull approach: P* only gets new updates from *Q*. When several sites are infectious, this approach works better. In this case, spreading updates is triggered by susceptible sites. It would be highly possible that such a site will contact an infectious one to pull in the updates and become infectious as well.
- *Push–pull approach:* This is a hybrid of pull and push approaches where *P* and *Q* send updates to each other.

One of the fundamental results of epidemic theory shows that simple epidemics eventually infect the entire population. If there is a single infectious site at the beginning, updates will eventually be spread across all sites using either form of anti-entropy. Full infection is achieved in expected time proportional to the logarithm of the population size.

Our case study for evaluating the behavior of pull-based anti-entropy model for multicast loss recovery, Bimodal Multicast [3], offers throughput stability, scalability and a bimodal delivery guarantee as the key properties. The protocol is inspired by prior work on epidemic protocols [9], Muse protocol for network news distribution [20], and the lazy transactional replication method of [19]. Bimodal Multicast consists of two sub-protocols, namely an *optimistic dissemination* protocol, and a *two-phase anti-entropy* protocol. The former is a best-effort, push mode, hierarchical multicast used to efficiently deliver a multicast message to its destinations. This phase is unreliable and does not attempt to recover a possible message loss. If IP multicast is available in the underlying system, it can be used for this purpose. Otherwise, a randomized dissemination protocol can play this role. The second stage of the protocol is responsible for message loss recovery. It is based on a pull-based anti-entropy protocol that detects and corrects inconsistencies in a system by continuous gossiping. The theory behind the anti-entropy protocol is given in [3]. The two-phase anti-entropy protocol progresses through unsynchronized rounds. In each round:

- Every group member randomly selects another group member and sends a digest of its message history. This is called a *gossip message*.
- The receiving group member compares the digest with its own message history. Then, if it is lacking a message, it requests the message from the gossiping process. This message is called *solicitation*, or retransmission *request*.
- Upon receiving the solicitation, the gossiping process retransmits the requested message to the process sending this request.

4. Simulation model and topologies

In this section, we first describe our simulation model via the protocol state diagram, and then explain our simulation topologies and scenarios.

4.1. Protocol state diagram

We use the simulation model we have developed for Bimodal Multicast protocol and its pull-based anti-entropy mechanism on ns-2 network simulator [2]. The simulation model consists of three modules: unreliable data dissemination using IP multicast protocol, pull-based anti-entropy protocol, and FIFO message ordering for applications. Details of our implementation are given in [29]. We follow an event-based approach as depicted by the state diagram of the protocol in Fig. 2. We define the following four events that trigger the protocol state actions:

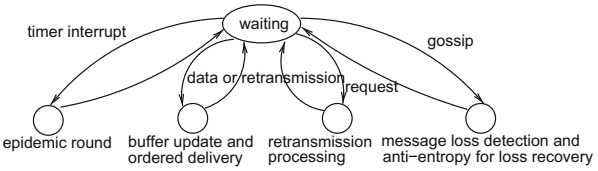

Fig. 2. Protocol state diagram.

- *Timer interrupt for an epidemic round:* In each epidemic round, request and retransmission counters are reset. Then, gossip message is sent to randomly selected members of the group, and the timer for the next epidemic round is scheduled.
- *Receipt of data or retransmission message:* When a member receives a data or a retransmission message, it updates the buffer and deliver messages that are now in order to the application.
- *Receipt of request message:* When a member gets a request message for a missing data, it performs retransmission processing operations.
- *Receipt of gossip message:* When a member receives a gossip message, message loss detection and request transmission for recovery are performed.

The duration of each round in the anti-entropy protocol is set to be larger than the typical round-trip time for an RPC over the communication links. The simulations conducted in this study use a round duration of 100 ms. Processes keep buffers for storing data messages that have been received from members of the group. Messages from each sender are delivered in FIFO order to the application. After a process receives a message, it continues to gossip about the message for a fixed number of rounds. Then, the message is garbage collected.

4.2. Topologies and simulation settings

Simulation topologies that we investigate consist of the following representative cases:

- Hierarchical dense topologies
- Clustered topologies
- Large-scale hierarchical sparse topologies

Hierarchical dense topologies are considered for various group sizes where all nodes are members of the multicast group and such topologies approximate the structure of the overlay networks. For this purpose, we have used several randomly generated transit-stub graphs produced by gt-itm topology generator [15]. These topologies consist of interconnected routing domains where each domain can be classified as either a stub or a transit domain [6]. A sample transit-stub topology consisting of total 20 nodes where there exists one transit domain with 4 nodes (0, 1, 2, and 3) is shown in Fig. 3(a). A certain link noise probability is set on every link that forms a randomized system-wide noise. An example application for this scenario would be a multicast based distance education session in a wide-area setting consisting of multiple local campus networks. In this case, a server would multicast the content to a group of nodes in the network. Another application could be an air traffic control system where the controller consoles are replicated to achieve fault tolerance, and need to communicate for providing data consistency.

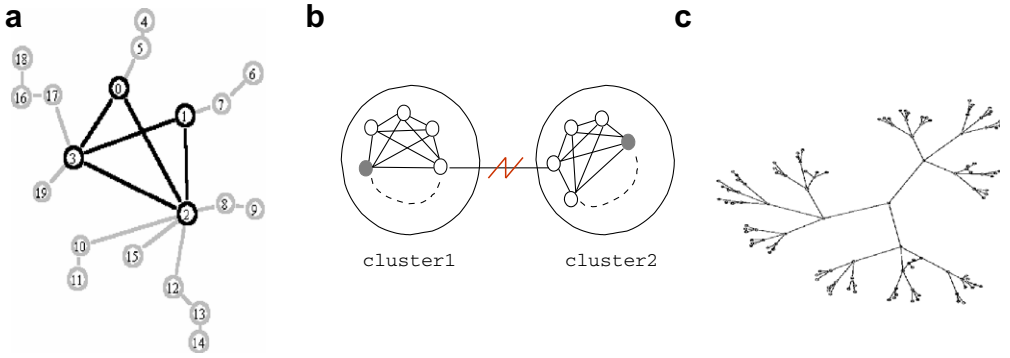

Fig. 3. Sample topologies: (a) transit-stub, (b) clusters, and (c) tree.

Clustered topologies include scenarios where LANs connected by long distance links and networks where routers with limited bandwidth connect group members and such configurations are common in today's networks as well. With these in mind, we constructed topologies with fully connected clusters, and a single link connecting those clusters. All nodes are considered as members of the multicast group where there exists one sender. Sender is located on the first cluster, and it generates 100 multicast messages per second. There is 0.01 intracluster noise rate formed in both clusters, and a varying high drop rate is injected on the link connecting the clusters which make it to behave as a bottleneck link. A sample cluster topology is depicted in Fig. 3(b).

Hierarchical sparse scenarios include large-scale transit-stub and tree topologies with 1500 and 1000 nodes, respectively. Group members are located at randomly selected nodes on the networks. A certain link noise probability is set on every link forming randomized system-wide noise. A sample tree topology is depicted in Fig. 3(c). In addition to incorporating network scenarios with various topologies to demonstrate topology independence, we varied operating parameters such as group size, multicast message rate of the sender, and system-wide noise rate. We obtain our results from several runs of simulations, each run consisting of a sequence of 35,000 multicast data messages transmitted by the sender.

5. Results

In this section, we analyze the robustness in the case of link failures, topology independence and scalability of epidemic multicast loss recovery using our simulation model. Furthermore, we compare its performance against the feedback control mechanism via simulations. Then, we conclude with a discussion of FEC approaches.

5.1. Recovery overhead distribution among group members, topology independence and robustness

We investigate loss recovery overhead distribution of the epidemic loss recovery on three scenarios. For the hierarchical dense topologies, message rate of the multicast source is 50 messages per second. Fig. 4 shows request messages received by each receiver in the multicast group for group and network sizes of 40, 80, and 120 nodes, and system-wide noise rate of 0.01 on all links. The *noise rate* indicates the probability of a message getting lost on each link of the network. It is observed that the load of overhead traffic is balanced over group members, a desirable property that avoids overloading one member or a portion of members with high overhead traffic. Increasing number of participants does not have a significant effect on the distribution only causing a negligible increase in the number of requests received by each group receiver. Likewise, Fig. 5 demonstrates analysis results where system-wide noise rate is increased to 0.1. It is obvious that increasing noise rate by a multiple of 10 on every link would increase the amount of message losses experienced on the network. In fact, when compared to Fig. 4, overhead does not increase linearly with the noise rate, and average overhead per member remains almost constant as group size scales up. The average overhead values per member in units of messages per second for different noise rate and group sizes are summarized in Table. 1. Comparative simulation results with nonhierarchical feedback control approach for this scenario are given in Section 5.3.

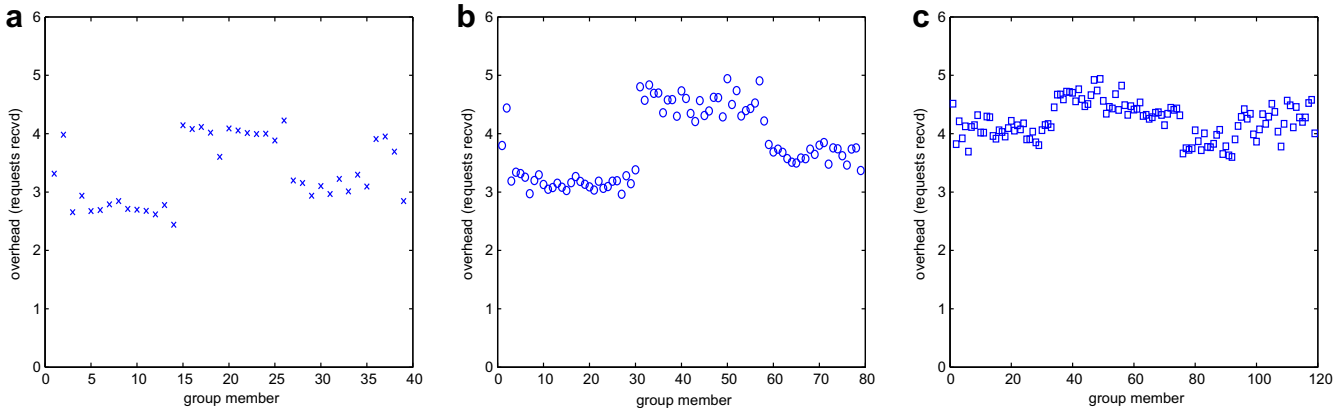

Fig. 4. Request messages received by each group member, Bimodal Multicast, hierarchical dense, noise rate: 0.01, number of nodes: (a) 40, (b) 80, and (c) 120.

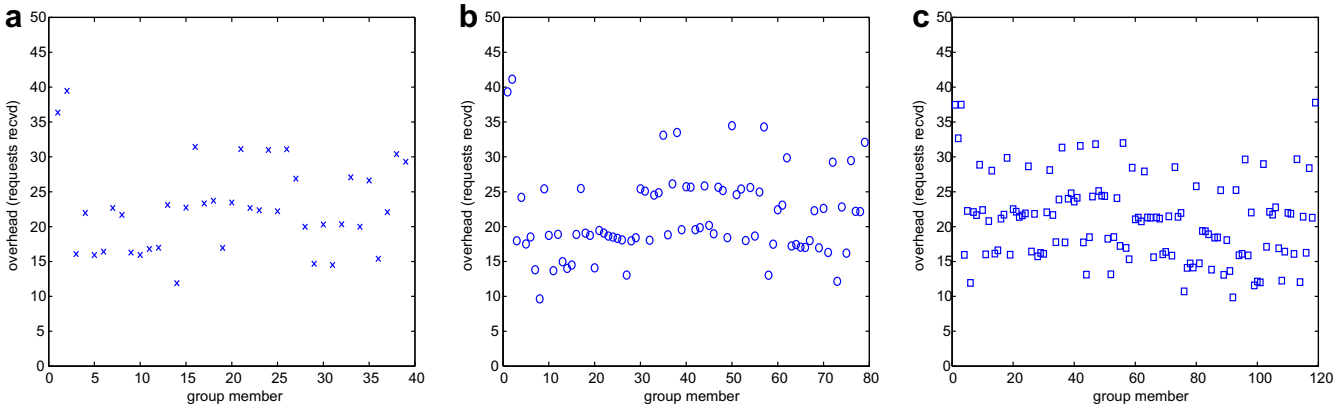

Fig. 5. Request messages received by each group member, Bimodal Multicast, hierarchical dense, noise rate: 0.1, number of nodes: (a) 40, (b) 80, and (c) 120.

Table 1
Hierarchical dense case: average overhead values.

Noise rate	40 nodes	80 nodes	120 nodes
0.01	3.34	3.80	4.22
0.1	22.58	21.48	20.82

Table 2
Clusters connected by a bottleneck link: average overhead values.

Noise rate	60 nodes	120 nodes
0.25	9.87	8.94
0.50	14.18	11.67

For clustered topologies, message rate of the multicast source is set to 100 messages per second. Fig. 6 shows request messages received by each receiver in the multicast group where group size is 60 and 120, respectively, for two different bottleneck noise levels (0.25 and 0.50). The first half of the group members reside in cluster 1 and the rest in cluster 2. Due to the high bottleneck noise rate, a large percentage of messages gets dropped during the transmission from cluster 1 to cluster 2. The resultant overhead distribution on group members shows the effect of bottleneck noise, where the members in the first cluster take a major part in the loss recovery process. However, the overhead on group members within clusters is balanced and even decreases on average as group size doubles from 60 to 120. This is because of the bottleneck effect on the link connecting two clusters. Request messages transmitted from one cluster to the other through the bottleneck are subject to a high noise rate and hence message drops. The average overhead values in units of messages per second are measured as given in Table 2.

For the large-scale sparse scenario, during each simulation, message rate of the multicast source is set to either 10 or 100 messages per second. Fig. 7(a) shows request messages received by each receiver in the multicast group where group size is 10 and system-wide noise rate on all links is 0.01. Values in these graphs show the averages obtained through several simulation runs. Similar distribution is found for retransmission messages generated by each receiver in the group. Like previous topologies, hierarchical dense and clusters, overhead traffic in large-scale sparse case is balanced over group members. Moreover, increasing multicast transmission rate of the sender by a multiple of 10 (from 10 messages per second to 100) does not change the overhead distribution, only causing a slight increase in the number of requests received by each group receiver.

Fig. 7(b) shows request messages generated by each receiver in the multicast group in the same settings. Our simulation results are consistent with the following probability calculation. We assume independent loss probabilities at the various links within a path

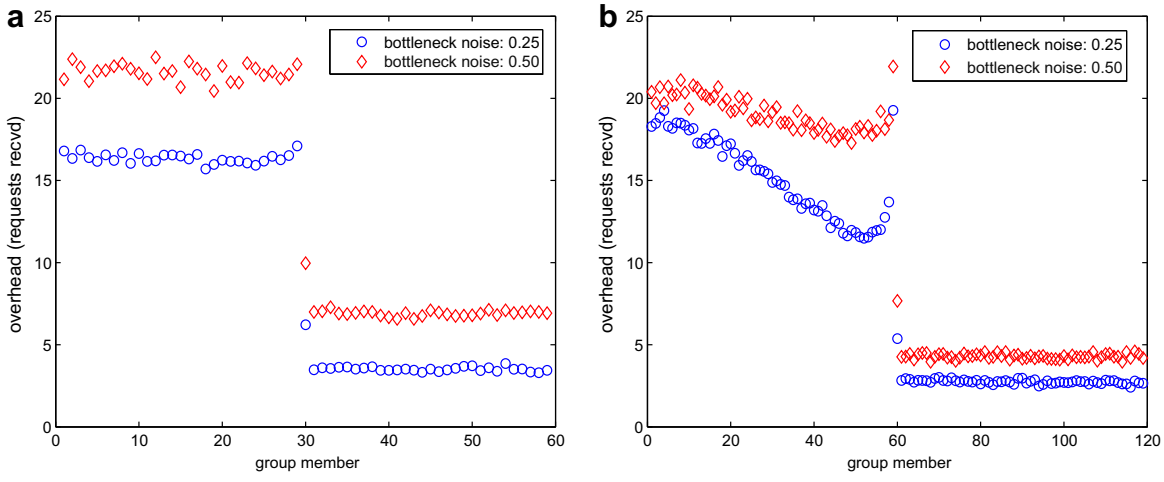

Fig. 6. Request messages received by each group member, Bimodal Multicast, clustered networks: (a) 60-member cluster and (b) 120-member cluster.

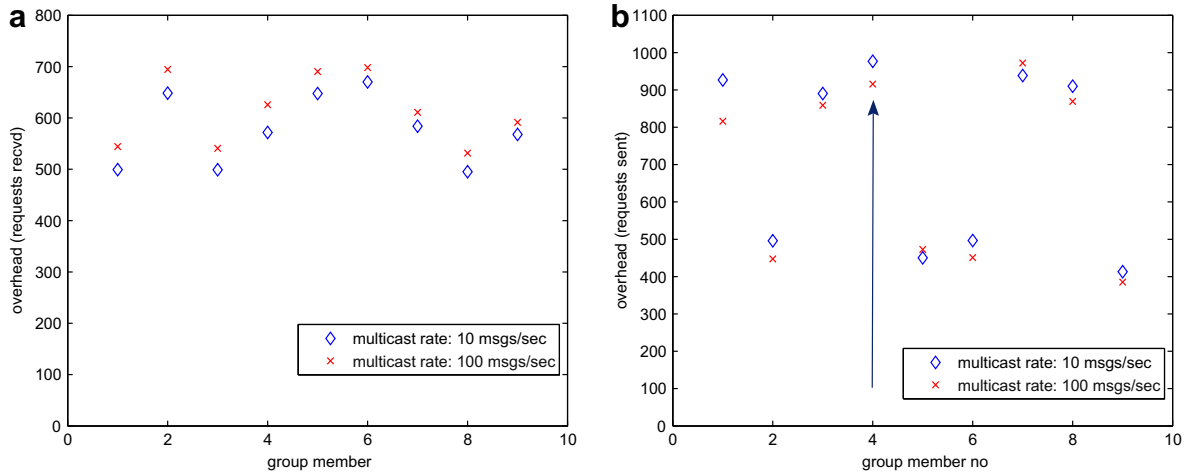

Fig. 7. (a) Requests received and (b) requests generated by each group member, Bimodal Multicast, 1500-node network, group size: 10, noise rate: 0.01.

on the network. If there are n links on the route from the sender to a specific receiver where noise rate on each link is p_i, then the probability of message loss experienced by the receiver can be calculated as follows. Let $P(loss)$ be the probability of loosing a message on its route from sender to the receiver. Then

$$P(loss) = 1 - P(no\ loss) = 1 - (1 - p_i)^n$$

As an example, consider the receiver 4 in Fig. 7(b) that is $n = 3$ links away from the sender. According to the above calculation for all multicast data messages transmitted by the sender, when $p_i = 0.01$ we would expect 951 messages to be lost on the way to receiver 4. This would essentially trigger the same amount of request messages to be generated by the receiver. That is actually what we have measured as our analysis result in consistency with the theoretical findings.

Fig. 8 demonstrates our results when we scale the group size from 10 to 50 on the network of 1500 nodes. Similar to our previous results, we observe that the load of overhead traffic is balanced over group members for both request (Fig. 8(a)), and retransmission traffic (Fig. 8(b)). Increasing message rate of the sender does not cause a significant overhead increase on each member, which is an indication demonstrating the scalability of epidemic loss recovery. This is partially due to the limits of requests and retransmissions imposed by the loss recovery mechanism. Each node can generate a certain amount of request and retransmission messages per epidemic round.

Overall, investigated recovery overhead features of the epidemic approach also show topology independence property. We observe similar overhead behavior on different network topologies. Furthermore, for all these network scenarios, full reliability during multicasting is achieved, that is all message losses are recovered and all receivers successfully deliver multicast data. In other words, robustness of epidemic loss recovery is shown in the case of various link failures and link bottlenecks.

5.2. Scalability: impact of group size, randomized system-wide noise and message rate

We now explore the impact of an increase in group size, message rate and randomized noise over network links on the scalability. For the hierarchical dense scenarios, Fig. 9 shows average loss recovery overhead in the form of requests received and sent on all receivers of the multicast group where system-wide noise rate and group size vary. A significant scalability result is that as group size increases, protocol overheads remain stable. Increasing noise rate p_i by a multiple of 10 (0.01–0.10) on all links causes a slight increase in the overhead experienced by each node. For instance, overhead percentage in the form of requests received increases

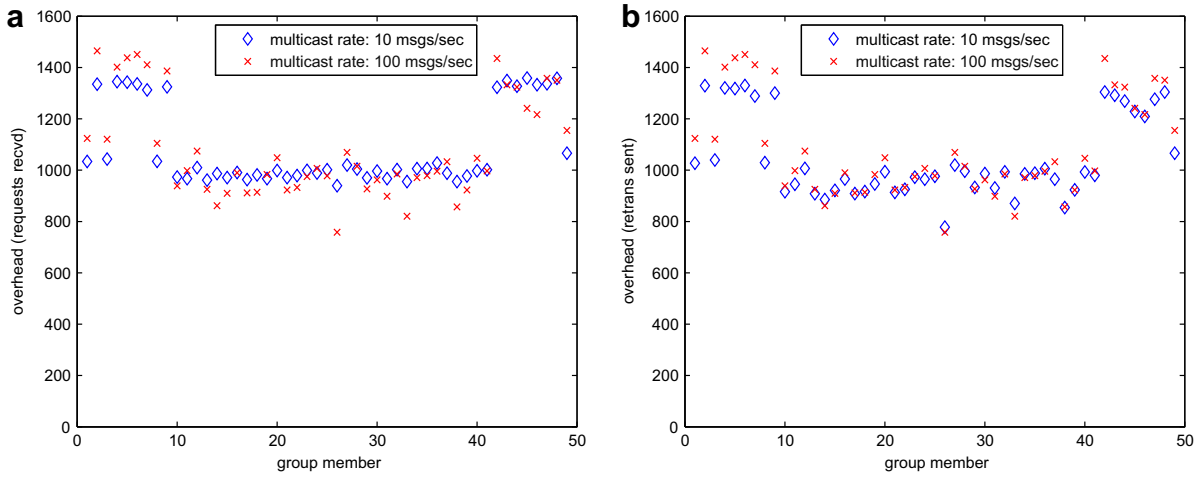

Fig. 8. (a) Requests received and (b) retransmissions generated by each group member, Bimodal Multicast, 1500-node network, group size: 50, noise rate: 0.01.

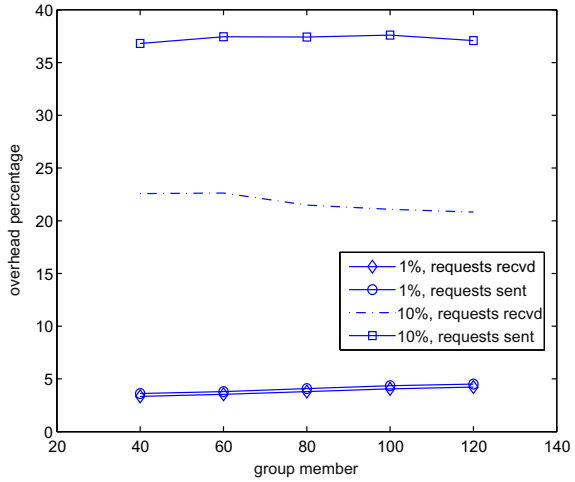

Fig. 9. Overhead percentage vs group size, hierarchical dense.

from 3% to 21% on average for all group sizes. These observations indicate the scalability of the epidemic loss recovery as group size and noise rate expand on hierarchical dense topologies. On large-scale transit-stub sparse topologies with 1500 nodes, Table 3 shows average loss recovery overhead (in the form of number of request messages received and retransmissions generated) observed on all receivers of the multicast group where three parameters, namely system-wide noise rate, group size and multicast message rate per second, vary. For an increase in noise rate, probability calculation of the previous section suggests a linear increase in recovery traffic where the other parameters are constant. In practice, our simulation results demonstrate better behavior with an expected increase in overhead traffic that is below linear an effect indicating the scalability of the loss recovery. As an example, for a group size of 50 and message rate of 100, overhead changes from 3% to 21% as noise rate increases from 0.01 to 0.1.

Furthermore, scaling the group size from 10 to 50 on the identical network settings causes a very slight change in the overhead load that is another dimension of scalability. For instance, for a noise rate of 0.1 and message rate of 100, overhead changes from 17% to 21% as group size increases from 10 to 50. Moreover, increasing message rate of the sender (from 10 messages to 100 messages per second), while keeping the other factors constant, has an insignificant effect on the overhead which makes epidemic loss recovery a good candidate for high-speed data distribution.

5.3. Comparison with nonhierarchical feedback control

Prior studies [21,23,3] have shown that, in feedback control mechanism for loss recovery as employed by SRM, random packet loss can trigger high rates of overhead messages. In addition, this overhead grows with the size of the system. Related to this scalability problem, we explore the behaviors of epidemic loss recovery of Bimodal Multicast and nonhierarchical feedback control of SRM on a large-scale network with high-speed data transfer. We choose feedback control for comparison, because it represents a commonly adapted model for multicast loss recovery and also its implementation is publicly available on ns-2.

In these simulations, we construct large-scale tree topologies consisting of 1000 nodes. Up to hundred of the 1000 nodes are randomly chosen to be group members. We set the message loss rate to 0.1% on each link with the sender located at the root node injecting 100 multicast messages per second. The results for the background overhead of each protocol in the form of request message traffic are shown in Fig. 10. They demonstrate that, as the network and process group size scale up, the number of control messages received by group members during loss recovery increases linearly for SRM protocol, an effect previously reported in [21,23]. These costs remain almost constant for Bimodal Multicast versions (in graphs these are labeled as Pbcast and Pbcast-ipmc for short). Pbcast-ipmc is the version of Bimodal Multicast that uses IP multicast for message repairs during loss recovery. Compared to the basic Pbcast, Pbcast-ipmc has a slightly lower overhead in the form of request messages. If multiple receivers missed a message, Pbcast-ipmc increases probability of rapid convergence during loss recovery.

On hierarchical dense scenarios, comparative results for epidemic and nonhierarchical feedback control approaches are displayed in Fig. 11. Three different group sizes (40, 80 and 120 members) are investigated for varying system-wide noise rates (1% and 10% per link). In Fig. 11(a), it is observed that, for a fixed noise rate on links, when group size scales up, overheads remain very stable for Bimodal Multicast. On the other hand, for SRM, number of request messages received grows with an increase in group and network size as shown in Fig. 11(b). Furthermore, number of request messages received by group members is quite low for Bimodal Multicast when compared to SRM. Pair-wise propagation of information among group members, when anti-entropy is used, triggers the better performance. On the other hand, SRM uses multicast dissemination during loss recovery in order to provide high reliability.

Likewise, we analyze the effect of growth in system-wide noise on the loss recovery overhead as the system size scales up. Fig. 12

Table 3
Impact of randomized noise, group size and message rate.

Parameters			Loss recovery overhead		
Noise rate	Group size	Message rate	Requests	Retransmissions	Percentage
0.01	10	10	575.88	575.88	2
		100	614.17	614.17	2
	50	10	1083.33	1044.92	3
		100	1079.98	1079.98	3
0.1	10	10	5204.97	5204.97	16
		100	5577.13	5577.13	17
	50	10	6873.06	6870.45	21
		100	7076.72	7076.72	21

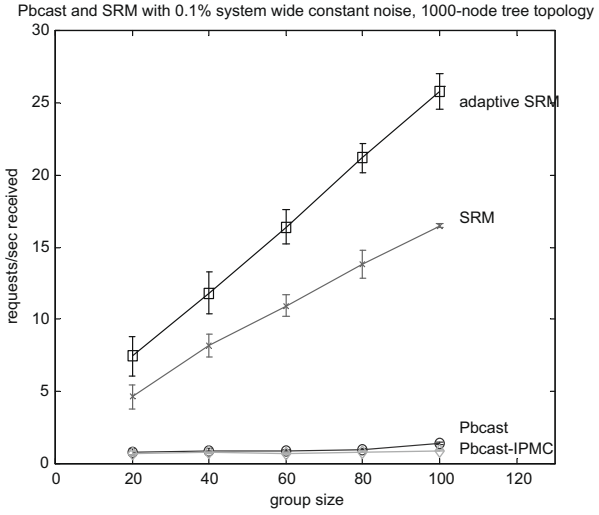

Fig. 10. Overhead in the form of requests per second for Bimodal Multicast and SRM, 1000-node tree topologies with 0.1% system-wide noise.

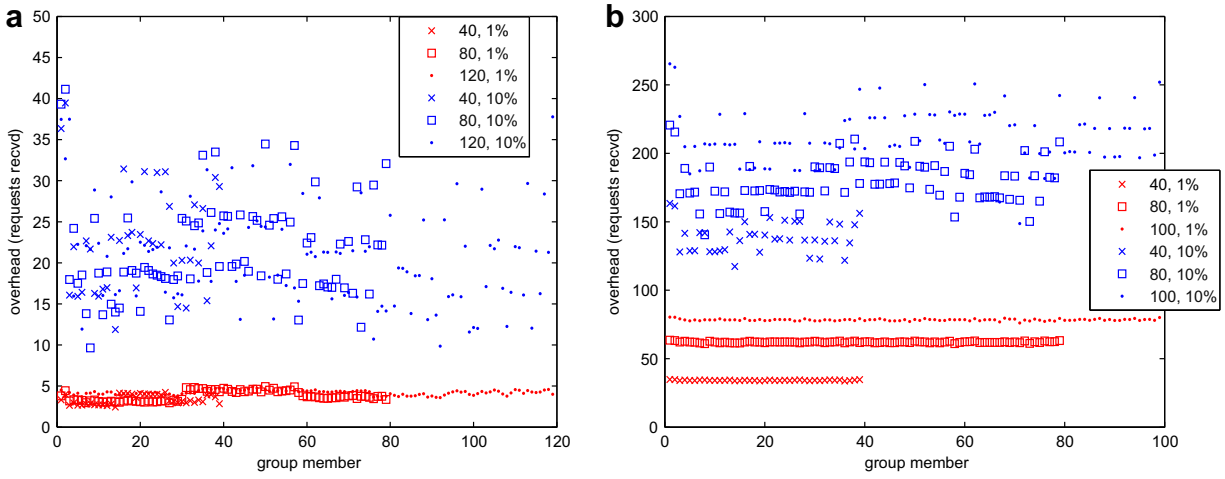

Fig. 11. Overhead for protocols, hierarchical dense: (a) Bimodal Multicast and (b) SRM.

demonstrates these results. In this analysis, we consider the average number of request messages both received and sent by group members. Scalable behavior of anti-entropy loss recovery has been observed for group sizes up to 120, whereas nonhierarchical feedback control reveals increase in the overhead as group size scales up, and a rapid increase has been observed for group sizes larger than 100.

5.4. Discussion of FEC mechanisms and raptor codes for reliable multicast

Recall that, for a (n,d) FEC code, d blocks of data messages are encoded into n blocks which the sender transmits. A receiver can reconstruct d data blocks given any d blocks correctly received out of n blocks transmitted by the sender. Significant performance features of FEC mechanisms are their encoding/decoding efficiency and algorithm time complexity.

FEC codes can be classified as small block, large block and Fountain/rateless codes [24]. An example for *small block codes* is the well-known Reed–Solomon erasure codes in which the optimal values for d and n are relatively small. A drawback with the use of larger (n,d) values in this case is the very large encoding and decoding times [35]. However, small block codes are preferred since a receiver can initiate decoding as soon as it receives d blocks out of n. *Large block codes* work well in terms of the performance of encoding and decoding with large d values. In this category, most codes derive from the well-known Low Density Parity Check (LDPC) codes [12]. A limitation is that the value of n should be defined prior to encoding and cannot be changed. *Fountain or expandable codes* overcome this limitation and support very large d values and variable n values without degrading the recovery efficiency. Thus, for a given set of input data blocks, a Fountain code generates as many (potentially limitless) encoded blocks as needed. Since the ratio d/n so-called the code rate can be very small, these codes are also referred to as *rateless* codes, and they are suitable for multicasting/broadcasting protocols over heterogeneous networks.

An efficient and well-known set of mechanisms in the category of Fountain codes are the Raptor codes [38]. They are proposed as an extension to Luby-Transform (LT) codes with the property of linear time encoding and decoding, and offer improved reliability for data dissemination. Encoding time is independent of the amount of repair data generated, and decoding time is independent of packet loss patterns. As another beneficial feature, they provide flexibility in parameter selection. Raptor codes are used in commercial systems of Digital Fountain corporation [45] for reliable data transmission over heterogeneous networks, and also have been selected as the standard for reliable multimedia multicast/broadcast in 3G mobile services. Mobile devices benefit from

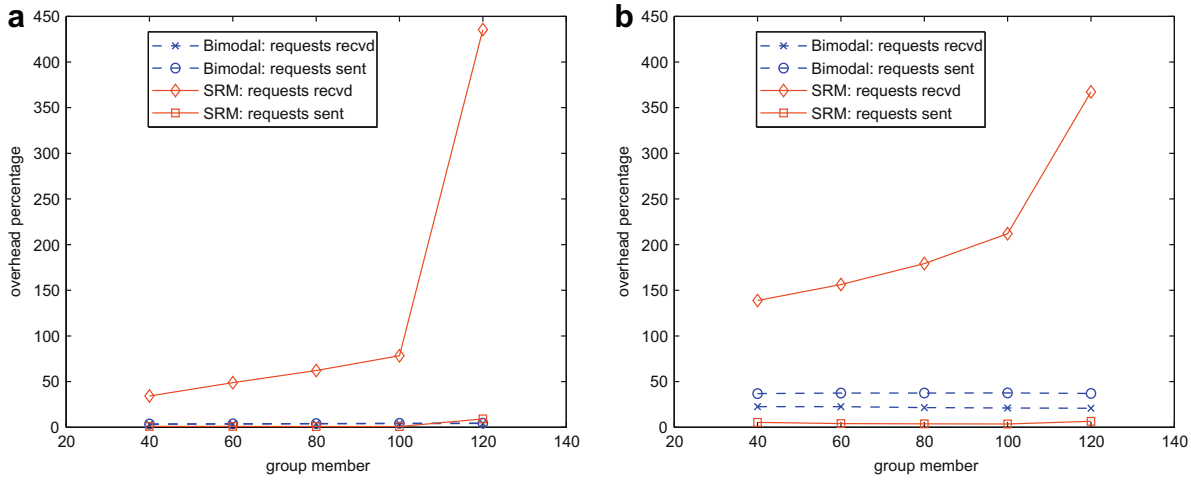

Fig. 12. Overhead vs group size for Bimodal Multicast and SRM, hierarchical dense: (a) 0.01 system-wide noise and (b) 0.10 system-wide noise.

reliable multicast mechanisms originally offered for the Internet. As discussed in [22], there are two approaches to support mobile users. One is that the traditional point-to-point cellular networks are extended with multicast/broadcast capabilities. For instance, Multimedia Broadcast/Multicast Service (MBMS) [44] specifies one-to-many content distribution for 3GPP. In the second approach, traditional broadcast systems such as Digital Video Broadcast (DVB) are extended with IP Datacast services. Raptor codes [38] have been selected for MBMS based on their high performance among the other FEC codes. DVB has also included application layer FEC using Raptor codes for IP Datacast services.

6. Conclusions and future work

An indispensable element for ensuring reliability at the transport level end-to-end multicasting is message loss recovery. This study yields conclusions on scalability and robustness of end-to-end epidemic loss recovery in multicast communication. Hierarchical and clustered topologies with various parameters are the representative network scenarios we have analyzed to demonstrate topology independence. On large-scale scenarios, both dense and sparse multicast groups are investigated. Our results show that epidemic loss recovery produces balanced overhead distribution among group receivers and it is scalable as group size, multicast message rate and system-wide noise rate increase. Robustness against network link failures is studied as well. We also compare the epidemic loss recovery with the feedback control model of SRM available on ns-2. Besides, a discussion of FEC mechanisms and in particular the Raptor codes for scalable reliable multicast is provided.

This study is significant since it extensively analyzes epidemic loss recovery model with various topology, link noise, group size and message rate properties representing overlay network scenarios. As part of the future work, we plan to run epidemic multicast loss recovery on network testbeds such as PlanetLab and Emulab, and compare it with overlay multicast approaches, like SplitStream [7], available on the testbeds. In this way, we aim to analyze the effects of power-law topologies as well. Also, the impact of firewalls and network address translation boxes, that can prevent peers from directly communicating in a loss recovery mechanism and possible solutions will be considered [34].

Network topologies with power-law degree distribution exhibit a hierarchy resembling the loosely hierarchical structure of the Internet [40]. Such degree-based topologies capture the large-scale structure of the networks well, and power-law distribution is meaningful for topologies consisting large number of nodes. For instance, power-law topology generator Inet [42] produces topologies with minimum 3037 nodes in order to represent large-scale structure of the Internet. This is due to the fact that, with small number of nodes, the degree distribution will not characterize the implicit hierarchy present in the topology. In this study, however, we observed that simulating such large topologies on ns-2, with the required level of network model details, becomes intractable due to huge computing resource needs. On the other hand, structural topologies such as tree and transit-stub are useful for smaller scale simulations as mentioned in [40]. Hence, we have used up to 1500 nodes with hierarchical topologies in our simulations.

As another research direction, scalability and robustness of loss recovery would be vital for multicast services over wireless networks as well [41]. Wireless multicast applications cover a wide range including mobile commerce, distance education, location-based services, and military command and control. These applications would especially benefit from multicast support with various forms of reliability guarantees. Adaptive multicast protocols that take the available bandwidth, loss rate characteristics, and user mobility into account are needed in such networks.

Acknowledgments

This research is supported by TUBITAK (The Scientific and Technical Research Council of Turkey) under CAREER Award Grant 104E064. We thank to anonymous reviewers for their valuable suggestions and comments.

References

[1] N.T.J. Bailey, The Mathematical Theory of Infectious Diseases and its Applications, second ed., Hafner Press, 1975.
[2] S. Bajaj, L. Breslau, D. Estrin, et al. Improving simulation for network research, USC Computer Science Dept., Technical Report, 1999, pp. 99–702.
[3] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, Y. Minsky, Bimodal multicast, ACM Transactions on Computer Systems 17 (2) (1999) 41–88.
[4] A.D. Birrell, R. Levin, R.M. Needham, M.D. Schroeder, Grapevine, an exercise in distributed computing, Communications of the ACM 25 (4) (1982) 260–274.
[5] J.W. Byers, M. Luby, M. Mitzenmacher, A digital fountain approach to asynchronous reliable multicast, IEEE Journal on Selected Areas in Communications 20 (8) (2002) 1528–1540.
[6] K. Calvert, M. Doar, E.W. Zegura, Modeling internet topology, IEEE Communications Magazine (1997).
[7] M. Castro, P. Druschel, A.M. Kermarrec, A. Nandi, A. Rowstron, A. Singh, SplitStream: high-bandwidth multicast in cooperative environments, ACM SIGOPS Operating Systems Review 37 (5) (2003) 298–313.
[8] M. Castro, P. Druschel, A.M. Kermarrec, A. Rowstron, SCRIBE: a large-scale and decentralized application-level multicast infrastructure, IEEE Journal on Selected Areas in Communications 20 (8) (2002) 1489–1498.
[9] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, Proceedings of the ACM PODC, 1987, pp. 1–12.
[10] P. Eugster, R. Guerraoui, A.-M. Kermarrec, L. Massoulie, Epidemic information dissemination in distributed systems, IEEE Computer (2004) 60–67.
[11] S. Floyd, V. Jacobson, C. Liu, S. McCanne, L. Zhang, A reliable multicast framework for light-weight sessions and application level framing, IEEE/ACM Transactions on Networking 5 (6) (1997) 784–803.
[12] R.G. Gallager, Low density parity check codes, IEEE Transactions on Information Theory 8 (1) (1962).
[13] J. Gemmell, T. Montgomery, T. Speakman, J. Crowcroft, The PGM reliable multicast protocol, IEEE Network 17 (1) (2003) 16–22.
[14] R.A. Golding, K. Taylor, Group membership in the epidemic style, Technical Report, UCSC-CRL-92-13, Univ. of California at Santa Cruz, 1992.
[15] GT-ITM Topology Generator. Available from: <http://www.isi.edu/nsnam/ns/ns-topogen.html>.
[16] K. Guo, Scalable message stability detection protocols, Ph.D. Dissertation, Cornell University, Dept. of Computer Science, 1998.
[17] M. Hosseini, D.T. Ahmed, S. Shirmohammadi, N.D. Georganas, A survey of application-layer multicast protocols, IEEE Communications Surveys & Tutorials 9 (3) (2007) 58–74.
[18] J. Jannotti, D.K. Gifford, K.L. Johnson, M.F. Kaashoek, J.W. O'Toole Jr., Overcast: reliable multicasting with on overlay network, in: OSDI'00: Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation, 2000.
[19] R. Ladin, B. Lishov, L. Shrira, S. Ghemawat, Providing availability using lazy replication, ACM Transactions on Computer Systems 10 (4) (1992) 360–391.
[20] K. Lidl, J. Osborne, J. Malcome, Drinking from the firehose: multicast USENET news, USENIX Winter, 1994, pp. 33–45.
[21] C. Liu, Error recovery in scalable reliable multicast, Ph.D. Dissertation, University of Southern California, 1997.
[22] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, W. Xu, Raptor codes for reliable download delivery in wireless broadcast systems, CCNC 2006, Las Vegas, January 2006.
[23] M. Lucas, Efficient data distribution in large-scale multicast networks, Ph.D. Dissertation, Dept. of Computer Science, University of Virginia, 1998.
[24] C. Neumann, V. Roca, R. Walsh, Large scale content distribution protocols, SIGCOMM Comput. Commun. Rev. 35 (5) (2005) 85–92.
[25] J. Nonnenmacher, E.W. Biersack, D. Towsley, Parity-based loss recovery for reliable multicast transmission, IEEE/ACM Trans. Networking 6 (4) (1998) 349–361.
[26] O. Ozkasap, Scalability, throughput stability and efficient buffering in reliable multicast protocols, Technical Report, TR2000-1827, Dept. of Computer Science, Cornell University, 2000.
[27] O. Ozkasap, Large-scale behavior of end-to-end epidemic message loss recovery, in: Proceedings of QofIS, Third COST 263 International Workshop on Quality of Future Internet Services, lncs 2511, Zurich, 2002, pp. 25–35.
[28] O. Ozkasap, Scalability and robustness of pull-based anti-entropy distribution model, in: Proceedings of ISCIS, 18th International Symposium on Computer and Information Sciences, lncs 2869, Antalya, 2003, pp. 934–941.
[29] O. Ozkasap, Performance study of a probabilistic multicast transport protocol, Performance Evaluation Journal 57 (2) (2004) 177–198.

[30] O. Ozkasap, M. Caglar, Traffic characterization of transport level reliable multicasting: comparison of epidemic and feedback controlled loss recovery, Computer Networks Journal 50 (2006) 1193–1218.
[31] S. Paul, K. Sabnani, J.C. Lin, S. Bhattacharyya, Reliable multicast transport protocol (RMTP), IEEE Journal on Selected Areas in Communications 15 (3) (1997) (special issue on Network Support for Multipoint Communication).
[32] V. Paxson, End-to-end internet packet dynamics, in: Proceedings of SIGCOMM, 1997, pp. 139–154.
[33] R. van Renesse, Y. Minsky, M. Hayden, A gossip-style failure detection service, in: Proceedings of Middleware'98, 1998, pp. 55–70.
[34] R. van Renesse, K. Birman, W. Vogels, Astrolabe: a robust and scalable technology for distributed system monitoring, management, and data mining, ACM Transactions on Computer Systems 21 (2) (2003) 164–206.
[35] L. Rizzo, Effective erasure codes for reliable computer communication protocols, ACM Computer Communication Review 27 (2) (1997).
[36] A.I.T. Rowstron, P. Druschel, Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems, in: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, 2001, pp. 329–350.
[37] D. Rubenstein, S. Kasera, D. Towsley, J. Kurose, Improving reliable multicast using active parity encoding services, Computer Networks 44 (1) (2004) 63–78.
[38] A. Shokrollahi, Raptor codes, Research Report DR2003-06-001, Digital Fountain, 2003.
[39] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup service for internet applications, in: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, San Diego, CA, 2001, pp. 149–160.
[40] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, W. Willinger, Network topology generators: degree-based vs. structural, SIGCOMM Comput. Commun. Rev. 32 (4) (2002) 147–159.
[41] U. Varshney, Multicast over wireless networks, Communications of the ACM 45 (12) (2002) 31–37.
[42] J. Winick, S. Jamin, Inet-3.0: Internet Topology Generator, University of Michigan Technical Report CSE-TR-456-02, 2002. Available from: <http://topology.eecs.umich.edu/inet/>.
[43] Z. Xiao, K.P. Birman, A randomized error recovery algorithm for reliable multicast, in: Proceedings of IEEE Infocom, 2001.
[44] 3GPP TS 26.346 V6.1.0, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service; Protocols and Codecs, June 2005.
[45] Digital Fountain. Available from: <http://www.digitalfountain.com/>.