















$j$  non-faulty nodes do not receive the message is:

$$\binom{n-t-1}{j} p^j (1-p)^{(n-t-1)-j}$$

Therefore, the probability that at least  $t$  non-faulty nodes will receive the message is:

$$\sum_{i=0}^{(n-t-1)-t} \binom{n-t-1}{i} p^i (1-p)^{(n-t-1)-i}$$

Fig. 6 shows this probability graphically for  $p$  ranging from 0.1 to 0.5, with  $t$  set to 2 and increasing  $n$  starting at the minimum value 6. The figure signifies that the probability of at least  $t$  non-faulty nodes receiving a broadcast approaches 1 quickly as number of nodes is increased, especially when the omission probability  $p$  is not high. Also the graph shows consistency with the simulation results in Fig. 3 at  $n = 10$ .

Fig. 6: The probability that at least  $t$  non-faulty nodes receive the first broadcast of a message.

## 5 Hierarchical Multicasting

Up to this point the multicasting paradigm was limited to only include nodes within a single network. Now we consider multiple networks, connected by gateways.

### 5.1 Simple connected networks

Fig. 7 shows a network consisting of two subnetworks  $A$  and  $B$  that are connected by gateways. There are a total of  $n_g$  gateways directly connected to the two neighboring subnetworks. As in the previous discussion, each subnetwork uses a single shared medium. Two multicast scenarios are now possible. Either the multicast spans over a single subnet or it spans over both subnets, i.e., the multicast includes nodes in networks  $A$  and  $B$ . Since single networks have been already discussed in the previous sections, the focus will be on the latter scenario.

Let's consider the connectivity of the two subnets. If  $n_g = 1$ , then all communication between  $A$  and  $B$  passes through the same gateway. In configurations with  $n_g > 1$ , the gateways operate in a redundant mode where each gateway forwards messages to all subnetworks that are directly connected to and have nodes participating in the multicast. Note that this is different from standard network configurations where only one gateway usually forwards packets, i.e., the configuration builds a spanning tree and multiple gateways play only a role if an active gateway in the spanning tree fails.

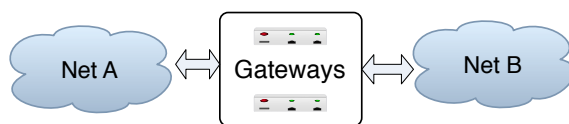


Fig. 7: Configuration with two networks

Consider the case where the subnetworks  $A$  and  $B$  communicate via a single gateway, i.e.,  $n_g = 1$ . Furthermore assume that a message broadcast originates on  $A$  and the multicast membership includes nodes in both subnets. Since the gateway is the only point of connection between the two subnetworks, the gateway will rebroadcast the message onto subnet  $B$ . However, the nodes in subnet  $B$  will not be able to receive the required  $(t + 1)$  identical copies of the same message from different nodes to accept the message. Additionally, the gateway can not forward the message onto  $B$  as a new one, since the message did not originate from the gateway. Furthermore, if the gateways are treated as standard network nodes, the requirement to receive  $(t + 1)$  identical copies would require the number of gateways to be at least  $(2t + 1)$ . However, this large number of gateways most likely is not practical for  $t > 2$ .

Therefore, to incorporate multiple subnetworks connected by gateways, the fault model needs to be adapted. Up to this point we considered  $t$  to be the number of faulty nodes. However, it seems reasonable to treat standard network nodes and gateways differently with respect to failure probability. It can be argued that a gateway is more reliable, or less prone to failure, due to the fact that it is not used as a general purpose computer and it is a special piece of hardware with dedicated software.

Let  $n_a$  and  $n_b$  denote the number of nodes in network segment  $A$  and  $B$  that participate in the multicast respectively, i.e.,  $n = n_a + n_b$ . The  $n$  nodes are in addition to the  $n_g$  gateways. Furthermore, let  $t_n$  denote the number of faulty nodes in all subnetworks

and let  $t_g$  be the number of faulty gateways between the two subnets, with the failure probability of a gateway to be much less than that of a network node, i.e.,  $t_g \ll t_n$ .

It can be shown that to tolerate the failure of  $t_n$  nodes and  $t_g$  gateways, one needs

$$n \geq 2t_n + 2$$

nodes and

$$n_g \geq 2t_g + 1$$

gateways, where no assumption is made about the distribution of the faulty nodes in the different subnetworks. Hence, the bound for  $N$  is the same as that in the single network case, i.e.,  $N = n_a + n_b + b$ .

The criterion for accepting a message via the broadcast paradigm across the gateways is different. Since authentication is assumed, i.e., no gateway can impersonate to be a different node, each receiving node can recognize whether a message is broadcast by a gateway. Therefore, a simple majority message forwards by the non-faulty gateways is sufficient. Consequently, a receiving node can vote on a correct message using a  $(2t_g + 1)$  majority of received messages. This voting effectively constitutes gateway fault masking and can be viewed as a restoring organ.

The overhead resulting from sending a single message to another network is of factor  $n_g$  plus the overhead associated with voting. Note that this only applies to messages sent across subnetworks and not to messages sent to the same subnetwork that contains the originating node. Furthermore, since the failure rate of a gateways is assumed significantly lower than that of a network node,  $t_g$  and thus  $n_g$  will be small. For example,  $t_g$  is likely to be equal to 1 or 2 resulting in 3 or 5 gateways, respectively.

A multicast network can be represented by a special kind of network graph in which the vertices are nodes or gateways and the edges are *logical* network connections. Given two graphs  $G_i$  and  $G_j$  with respective vertex sets  $V_i$  and  $V_j$  and edge sets  $E_i$  and  $E_j$ , the *union*  $G = G_i \cup G_j$  has  $V = V_i \cup V_j$  and  $E = E_i \cup E_j$ . The *join*  $G = G_i + G_j$  consists of  $G_i \cup G_j$  together with all edges joining  $V_i$  and  $V_j$ , i.e.,  $\forall v_p \in V_i$  and  $\forall v_q \in V_j$ ,  $e_{p,q} \in E$ , the edge set of the join graph. Fig. 8 shows an example of a join graph. The edges defining the join operation between  $G_1$  and  $G_2$  are shown as dashed lines. Note that the edge connectivity is logical, i.e., in the context of broadcasting, if a vertex emits a message, all connected vertices receive the message (in the error-free case). This should not be confused with point-to-point messages, where vertices emit messages to neighboring vertices on one edge at a time.

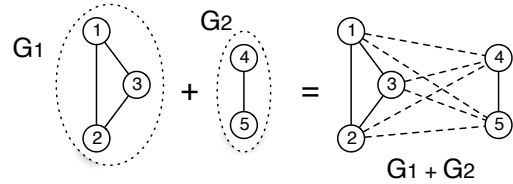


Fig. 8: Join operation (+) of two graphs

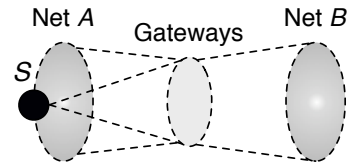


Fig. 9: General join graph of two subnetworks

Let's revisit the requirement that a new broadcast must be received by at least  $t_n$  other non-faulty nodes in the context of multiple subnetworks. The fact that these nodes may now span over two neighboring networks has no consequence other than the overhead associated with sending  $(2t_g + 1)$  messages for each message that needs to cross gateways. As long as no more than  $t_g$  gateways are affected by failure or malicious act, each node which receives forwarding messages from gateways will be able to determine whether it should accept the message, i.e., if  $(t_n + 1)$  identical messages are received, or discard the message, i.e., if  $(n - t_n - 1)$  NACKs are received.

The join graph associated with Fig. 7 is shown in Fig. 9. Node  $S$  broadcasts to the nodes in the two subnetworks, shown in dark gray, and the gateways are depicted in the light-gray shaded oval. Formally, node  $S$  broadcasts on subnetwork  $A$  and to the gateways. The gateways, by the nature of the join operation, forward the messages to all participating nodes in  $B$ , which vote on the messages received from the gateways. The additional overhead associated with the gateways is captured by the join operation of the two subnetworks, when a new message is broadcast for the first time. Specifically, a new message broadcast onto subnetwork  $A$  needs to be rebroadcasted by at most  $n_g$  gateways. But, the message overhead due to rebroadcasts crossing the gateways is less. This is because the number of gateways is small, and thus  $(t_g + 1)$  identical copies are needed to accept a message in comparison to  $(t_n + 1)$  needed by the nodes on the originating subnetwork. For example, three gateways can mask one failure, which is a relatively small price to pay.

## 5.2 General hierarchical multicasting

Whereas the previous subsection considered the special case of two subnetworks connected by gateways, this subsection is concerned with general network configurations. Consider a scenario with  $n_s$  network segments that are interconnected with gateways. Fig. 10 shows a configuration with  $n_s = 4$  and de-

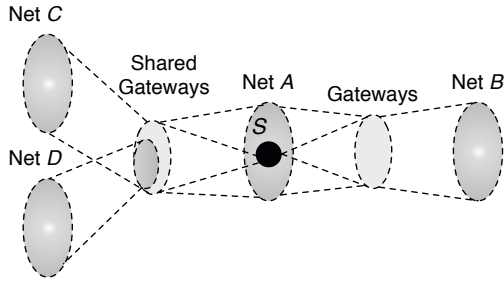


Fig. 10: General join graph with multiple networks

picts the possible scenarios in which subnetworks can be connected with gateways. The segments  $A$  and  $B$  are connected via a group of gateways, which are not connected to any other networks. On the other hand, networks  $A, C$ , and  $D$  share gateways. In fact the gateways connecting  $A$  and  $D$  are a subset of the gateways connecting  $A$  and  $C$ . The network model of the previous subsection can now be extended to multiple subnetworks. The following cases can be enumerated, where the associated network interconnections are indicated in parenthesis:

1. *Two subnetworks are interconnected by non-shared gateways.* This is the scenario for subnetwork  $(A, B)$ , where the number of non-shared gateways is  $n_{g(A,B)}$ . In order to mask  $t_{g(A,B)}$  gateway faults, at least  $(2t_{g(A,B)} + 1)$  gateways are needed.
2. *Multiple subnetworks share gateways.* This is the case involving subnetworks  $A, C$ , and  $D$ . To mask  $t_{g(A,C)}$  gateway faults, a total of  $n_{g(A,C)} \geq 2t_{g(A,C)} + 1$  gateways are needed. Similarly, a total of  $n_{g(A,D)} \geq 2t_{g(A,D)} + 1$  gateways are needed to mask  $t_{g(A,D)}$  gateway faults. However, with respect to  $n_{g(A,C)}$  no more than  $t_{g(A,D)}$  of the shared gateways (depicted by the darker shaded oval) may fail, and the remaining  $(t_{g(A,C)} - t_{g(A,D)})$  faults must occur on non-shared gateways (drawn in a lighter shade of gray in the oval).

The discussion above assumed subnetworks to be only one hop away from the initial sending node. Mul-

iple hops are considered in Fig. 11. There is no assumption about the distribution of the  $t_n$  faulty nodes, e.g., they could be concentrated in specific subnets or distributed randomly over all subnets. In the first case, multi-hop messages are forwarded from one group of gateways to the next and due to the majority argument of  $(2t_g + 1)$  messages, there will always be a majority of correct messages relayed to the next hop.

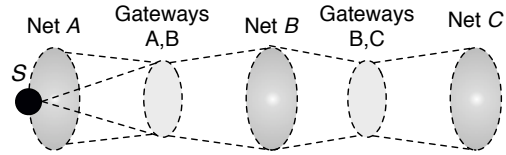


Fig. 11: General join graph with multiple hops

Considering  $n_g$  gateways between any two neighboring subnetworks and  $n_s$  segments, so that  $n = n_1 + n_2 + \dots + n_{n_s}$ , the bounds in the previous subsection regarding the two subnetworks still hold for the  $n_s$  segments, i.e.,  $N \geq 2t_n + b + 2$  and  $n_g \geq 2t_g + 1$ . Similarly, the overhead associated with a multicast spanning over multiple  $n_s$  subnets is induced by the message traffic of the redundant gateways, i.e.,  $(2t_g + 1)$  messages per subnet hop, and it is bound by a total of  $n_s n_g = n_s(2t_g + 1)$  gateways. If gateways are shared, as shown in Fig. 10 for networks  $A, C$ , and  $D$ , the number of gateways is reduced by the number of shared gateways.

## 6 Conclusion

Using a broadcast medium, it is difficult or unlikely that messages can be received asymmetrically, but it has been shown that an asymmetric behavior is still possible by allowing a faulty node to broadcast different messages for a given original message at different times. It has been shown that  $(t_n + 1)$  identical messages are needed to deliver a message if the message is not received in the first broadcast, and  $(n - t_n - 1)$  NACKs are needed to ignore a message. The research did not discuss transmission of messages by the use of digital signatures. If messages are digitally signed, the receipt of only one digitally signed message is sufficient to accept the message, instead of  $(t_n + 1)$  identical messages. But there would not be any changes in  $(n - t_n - 1)$  NACKs that are needed to address M1.

The protocol assumes that a faulty node can behave in OS, TS, and TA manners at the protocol level. In addition, it is assumed that the network medium can temporarily experience omissions in the form of OS or SOA, due to different scenarios such as loss of

a message or not being able to correct errors.

It has been shown that to tolerate the failure of  $t_n$  nodes and  $t_g$  gateways, one needs  $N \geq 2t_n + b + 2$  nodes, possibly spread over  $n_s$  subnets, and  $n_g \geq 2t_g + 1$  gateways, connecting any two adjacent subnets.

One of the requirements in achieving reliable multicasting is reaching agreement in delivering a message. Normally, a Byzantine agreement algorithm is needed to ensure that every non-faulty node agrees on delivering a message. The process of reaching this agreement for every message will become very prohibitive if new messages are transmitted continuously, or if a point-to-point topology were used. The proposed protocol reduces message complexity, as each node can independently determine the safety of delivering a message to the user. Furthermore, simulation has shown message complexity is dynamically adjusted according to the level of network fault condition, whilst the correct operation of the network is maintained.

#### References:

- [1] M.H. Azadmanesh, R.M. Kieckhafer, Exploiting Omissive Faults in Synchronous Approximate Agreement, *IEEE Transactions on Computers*, Vol.49, No.1010, 2000, pp. 1031-1042.
- [2] O. Babaoglu, R. Drummond, Streets of Byzantium: Network Architectures for Fast Reliable Broadcasts, *IEEE Transactions on Software Engineering*, Vol.SE-11, No.6, 1985, pp. 546-554.
- [3] F. Barsotti, A. Caruso, S. Chessa, The Localized Vehicular Multiast Middleware: A Framework for Ad Hoc Inter-Vehicles Multiast Communications, *WSEAS Transactions on Communications*, Vol.5, No.9, 2006, pp. 1763-1768.
- [4] T. Bates, Multiprotocol Extensions for BGP-4, *Network Working Group*, RFC 2858, <http://faqs.org/rfcs/rfc2858.html>, 2000.
- [5] K. Birman, T. Joasph, "Communication Support for Reliable Distributed Computing", *Lecture Notes in Computer Science*, Vol.448, 1987, pp. 124-137.
- [6] X. Defago, A. Schiper, P. Urban, Total Order Broadcast and Multiast Algorithms, *ACM Computing Surveys*, Vol.36, No.4, 2004, pp. 372-421.
- [7] K. Driscoll, B. Hall, H. Sivencrona, P. Zumsteg, Byzantine Fault Tolerance, From Theory to Reality, *Lecture Notes in Computer Science (LNCS)*, Computer Safety, Reliability, and Security, Vol.2788, 2003, pp. 235-248.
- [8] H. Eriksson, MBone: The Multicast Backbone, *CACM*, Vol.37, No.8, 1994, pp. 54-60.
- [9] W. Fenner, Internet Group Management Protocol, *Network Working Group*, RFC 2236, <http://faqs.org/rfcs/rfc2236.html>, 1997.
- [10] K.P. Kihlstrom, The SecureRing Group Communication, *ACM Transactions on Information and System Security*, Vol.4, No.4, 2001, pp. 371-406.
- [11] L. Lamport, et al, The Byzantine Generals Problem, *ACM TOPLAS*, Vol.4, No.3, 1982, pp. 382-401.
- [12] D. Laqab, *Survivable Multicast Communication in Bus-based Networks*, MS Thesis, Computer Science Department, University of Nebraska-Omaha, 2006.
- [13] B.N. Levine, J.J. Garcia-Luna-Aceves, A Comparison of Reliable Multicast Protocols, *Multimedia Systems*, Vol.6, No.5, 1998, pp. 334-348.
- [14] X. Li, M.H. Ammar, S. Paul, Video Multicast over the Internet, *IEEE Network*, Vol.13, No.2, 1999, pp. 46-60.
- [15] P.M. Melliar-Smith, L.E. Moser, Trans: A Reliable Broadcast Protocol, *IEE Proceedings*, Vol.140, No.6, 1993, pp. 481-493.
- [16] P.M. Melliar-Smith, L.E. Moser, V. Agrawala, Broadcast Protocols for Distributed Systems, *IEEE Transactions on Distributed and Parallel Systems*, Vol.1, No.1, 1990, pp. 17-25.
- [17] C.K. Miller, *Multicast Networking and Applications*, Addison-Wesley, 1999.
- [18] L.E. Moser, P.M. Melliar-Smith, Byzantine-Resistant Total Ordering Algorithms, *Information and Computation*, Vol.150, No.1, 1999, pp. 75-111.
- [19] M. Paulitsch, J. Morris, B. Hall, K. Driscoll, E. Latronico, P. Koopman, Coverage and the Use of Cyclic Redundancy Codes in Ultra-Dependable Systems, *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, 2005, pp. 346-355.
- [20] M. Pease, et al, Reaching Agreement in the Presence of Faults, *JACM*, Vol.27, No.2, 1980, pp. 228-234.
- [21] C. Shih, T. Shih, Cluster-based Multicast Routing Protocol for MANET, *WSEAS Transactions on Computers*, Vol.6, No.3, 2007, pp. 566-572.
- [22] H. Shin, K. Cho, A IP Multicast Technique for the IPTV Service, *WSEAS Transactions on Communications*, Vol.6, No.1, 2007, pp. 274-277.
- [23] P.M. Thambidurai, Y.K. Park, Interactive Consistency with Multiple Failure Modes, *Proceedings of the 7th Reliable Distributed Systems Symposium*, 1988, pp. 93-100.

- [24] B. Wang, C. Hou, A Survey on Multicast Routing and its QoS Extension: Problems, Algorithms, and Protocols, *IEEE Network*, Vol.14, No.1, 2000, pp. 22-36.
- [25] P.J. Weber, Dynamic Reduction Algorithms for

Fault Tolerant Convergent Voting with Hybrid Faults, PhD Dissertation, Electrical & Computer Engineering, Michigan Technological University, 2006.