Fast Message Dissemination for Emergency Communications

Ricardo Lent, Omer H. Abdelrahman, Gokce Gorbil and Erol Gelenbe
Department of Electrical and Electronic Engineering
Imperial College, London SW7 2BT, UK
Email: {r.lent, o.abd06, g.gorbil, e.gelenbe}@imperial.ac.uk

Abstract—**This paper presents an emergency communication system that is able to quickly deliver emergency messages over an unreliable and best-effort network such as the Internet. The proposed architecture employs application-layer multicast to rapidly deliver emergency traffic without the support of a dedicated network infrastructure. We introduce a distributed overlay tree construction and maintenance mechanism that produces consistent, loop-free and self-adaptive trees that dynamically change over time to effectively deal with varying network conditions and offer low message delays to end nodes. We evaluate the performance of the proposed approach through an experimental study conducted on a real-life networking testbed.**

I. Introduction

Emergency scenarios, such as natural disasters, terrorist threats and campus crimes, are accompanied by communication difficulties due to congestion and failures which are likely to occur in telecommunication networks during such circumstances. As a result, there are serious reliability issues with relying on a single technological solution for emergency communication. For instance, recent studies [1], [2] have pointed out several weaknesses in Short Message Service (SMS) based mass notification systems. Indeed, in order to prevent significant message losses due to communication failures as experienced in [3], [4], different communication channels and devices need to be utilized.

A critical component of emergency communications is multicasting. Information such as details of the emergency or evacuation guidelines may need to be conveyed to affected people, or critical information may need to be exchanged between various coordinating rescue teams such as the police, fire brigade and emergency medical services. Thus, there is a need for reliable and cost-effective communication tools that can adapt to varying network conditions during emergencies in order to satisfy the stringent delay requirements in such circumstances.

Although broadcast media are still dominant, the Internet continues to take on an increasing role in our lives. Indeed, a recent study [5] indicates an annual increase of 34% in Internet usage in the UK. It is therefore worthwhile to consider the Internet as a complimentary means for distributing emergency information. An example of such emergency notification systems that use multiple technologies, including the Internet, is AMBER alert [6] which is used in the US to aid in the recovery of abducted children.

IP multicast provides a low-cost bandwidth-efficient group communication service on the Internet but it has not been widely adopted by ISPs due to technological and business challenges [7]. Also, a best-effort network such as the Internet cannot adequately support the stringent time requirements of emergency communications, particularly when network conditions and traffic demands change rapidly.

Overlays enable multicast functionalities such as packet replication, routing and group membership management to be implemented at the application layer on the end hosts rather than at the network layer in the routers as in IP multicast. Hence, nodes participating in the multicast session can self-organize into an overlay network and take on the responsibility of forwarding data between themselves using only unicast connections. This approach conforms to the "smart-host dumb-network" paradigm that has driven the Internet, allowing for immediate deployment at the cost of less efficient utilization of network resources in comparison to IP multicast due to duplicate transmissions over the same physical links.

In this paper, we introduce a simple, yet effective, overlay multicast protocol that is targeted for emergency communication services characterized by low-latency requirements in rapidly varying network conditions. We introduce a distributed overlay construction and maintenance mechanism that aims to deliver emergency traffic from a source to intended receivers in the least possible time. These receivers can be emergency management teams or special devices that broadcast information relevant to the emergency to the general public. Our proposed protocol has many unique features which distinguish it from existing overlay multicast solutions.

The problem of designing efficient end-system or application-layer multicast (ALM) protocols consists mainly in constructing high quality and self-adapting overlays with minimal complexity and overhead. Building such overlays, however, requires knowledge of the underlying physical network and its performance metrics which are usually estimated using measurement techniques such as round-trip time (RTT). A direct consequence of gathering measurements for all pairs of nodes in the overlay is a large protocol overhead that grows with the multicast group size. Thus, an overlay construction algorithm needs to achieve good trade-off between topology estimation and measurement cost.

There has been considerable work in the literature on providing emergency communication services through the Internet. The problem of interoperability between emergency responders with different communication technologies is considered in [8]. The works in [9], [10] present methods for identifying IP caller location during an emergency, which is necessary to direct the call to the correct responder and to dispatch aid to the right location. Finally, [11] proposes a differentiated services (DiffServ) approach for providing quality of service (QoS) for emergency traffic.

Our paper is most closely related to [12], [13]. In [12], overlay networks are used to improve survivability and reliability of communication services during emergency scenarios characterized by large-scale network failures. The scheme builds on the overlay routing protocol of [14], and simulation results indicate that the overlay approach can increase the probability of network recovery after failures in comparison to Border Gateway Protocol (BGP). In [13], a prototype application for emergency response communications is implemented. The application integrates a peer-to-peer (P2P) architecture [15] with GPS data in order to provide location-based messaging and video streaming services to emergency responders.

The rest of this paper is organized as follows. In section II we review some of the previous work on application-layer multicast. Section III presents the proposed overlay tree construction and maintenance protocol. Section IV provides a detailed description of the implementation of the protocol, and presents experimental results conducted on a network testbed. Finally, we provide our concluding remarks in section V.

II. Related Work

Previous work on end-system multicast protocols can be classified, according to the overlay structure used, into tree-based [16]–[19] and mesh-based [20]–[22]:

1) Tree-based protocols: arrange members into a tree structure so that data dissemination can be done by a simple flooding technique whereby each node replicates and forwards every packet to all of its neighbors except the source node. According to the construction mechanism (and hence the purpose) of the tree, tree-based protocols are classified as *source-specific* and *group-shared* tree protocols. Source-specific tree protocols aim to minimize the path costs from a specific source to all other destinations in the group, thus optimizing the tree for a given member. Therefore, such protocols are efficient for one-to-many applications, but may incur high overhead due to maintenance costs for multi-source communications. Group-shared tree protocols, on the other hand, try to minimize the overall tree cost in order to efficiently support many-to-many applications using the same overlay tree, but at the expense of higher communication costs for any specific source.

2) Mesh-based protocols: build, in addition to the data delivery tree, a mesh topology that can be used to disseminate control messages, recover from tree partitions and improve performance. Consequently, protocols of this group are more robust to failures due to the availability of multiple or redundant links between group members. However, they suffer from high control overhead which limits their scalability. Existing mesh-based protocols differ mainly in the tree construction procedure and the order in which the data topology (tree) and the control topology (mesh) are built.

In general, most existing end-system multicast solutions build group-shared overlays that, although incur very small overhead, cannot fulfill the critical low-latency requirements of emergency communications. Moreover, source-specific tree protocols derive the data delivery paths from mesh topologies that require very large control overhead in order to maintain high quality mesh structures. Our paper differs from existing work in this area by several unique features which are summarized below:

- We propose a distributed tree construction and maintenance technique which allows for complete reconfiguration of the overlay tree topology in response to changing workload or network conditions. Conversely, existing ALM protocols try to optimize the overlay either by using centralized solutions or having group members make independent transformation decisions that may lead to sub-optimal performance, loops or partitions.
- The proposed tree construction and maintenance mechanism embeds both metric (i.e. link cost) measurement and link selection within the same process, which removes the need for additional monitoring efforts as done in comparable approaches.
- We propose an adaptive tree refinement algorithm which is triggered probabilistically according to the performance of the overlay. This is more efficient than periodic refinement techniques that require optimization over refinement periods in order to avoid unnecessary maintenance overhead.
- The overlay topology produced by the protocol is loop-free by construction, therefore no loop detection or recovery procedures are needed.
- The initial tree construction algorithm uses a distributed flooding technique which may incur a large overhead. However, the tree maintenance process uses search space reduction to improve scalability by selecting the best peers at each tree refinement step.
- The protocol was developed with the possibility of using *deputies* [23], [24] in mind, which are dedicated nodes that can be invoked to temporarily enter a multicast group to improve performance.

III. The Protocol

Our proposed protocol is a source-tree based application-layer multicast protocol which constructs the overlay tree in a distributed manner. A source-specific approach was adopted as we believe most emergency communications will involve few data sources, from where information will be sent out to many recipients; this approach guarantees better performance in single-source, multiple-receiver type of communications.

In the protocol, a delivery tree is source-specific and it is composed of three types of nodes: a sender s (tree root), a set of receivers M and a set of deputies D, which may be the empty set. Therefore, tree $T = \{s\} \cup M \cup D$. Each node $n \in T$ maintains two information sets: (i) the set of children C_n, which consists of n's current children in the distribution tree, and (ii) the set of possible successors N_n, which contains nodes that can potentially be selected as the children of n in the next tree construction cycle. Note that $C_l = \emptyset$ for all leaf nodes $l \in T$ and members of C_n may potentially change over time $\forall n \in T$ as the tree adapts itself to changing conditions. The set N is used to reduce the search space (i.e. the distributed broadcast space during tree (re-)construction) when the number of receivers ($|M|$) is large. Therefore, $N \subseteq M$. For the following discussion, you may assume $N = M$. Deputies are network nodes that can be used to support improved tree construction in order to enable performance improvements. These nodes are not among the group communication end-points ($D \cap M = \emptyset$ and $\{s\} \notin D$) but they are included in the delivery tree as their inclusion offers benefits during group communication (e.g. improved end-to-end latency). Deputy units are usually determined by an external process based on either policy or performance, but their use will not be discussed in this paper (for a similar protocol which uses deputies, we would like to refer the reader to [24]). For the following discussion, you may assume $D = \emptyset$.

The initial tree has a trivial star topology: $C_s = M$. This initial topology will change as the delivery tree adapts itself to varying conditions (e.g. inter-traffic patterns, network loads, and changes in M) by using the tree construction algorithm (TCA), which aims to improve the delivery latency of messages under the current substrate network conditions.

A. *Tree Construction and Message Delivery*

The Tree Construction Algorithm (TCA) is a distributed algorithm that can discover low-latency distribution trees for s and consists of the following main steps:

1) Sender s creates a candidate set Z_s, which consists of possible children of s in the to-be-constructed multicast tree, and sets $C_s = \emptyset$. The possible children are selected from N_s based on their fitness as observed from previous tree constructions. Note that $N_s = M \cup D - \{s\}$ when there are no previous observations (i.e. during the first tree construction). We measure node fitness as a binary metric in this paper, which indicates whether a given node had assumed a child role in the previous tree construction. This binary value $f_s(n) = 1$ if node n was a child of s in the previous tree, and $f_s(n) = -1$ otherwise. The set Z_s is constructed by including all nodes with a positive fitness value (call this selection $Z_s^+ = n \in N_s : f_s(n) > 0$). In addition, a random selection of nodes with negative fitness values are included in Z (call this selection Z_s^-), where $|Z_s^+| = |Z_s^-|$. The addition of Z^- enables the algorithm to deal with changing network dynamics by exploring new tree configurations. Members of D are also added to Z_s, so $Z_s = Z_s^+ \cup Z_s^- \cup D$. Note that when there are no previous observations, $Z_s = N_s$.
2) Sender s unicasts L copies of the Tree Build Message (TBM) to each node in Z_s. Each TBM carries a unique tree identifier assigned by s along with the sender's identifier and predecessor. Each node remembers the last tree identifier, so stale (delayed) TBMs can be easily discarded. Using multiple TBMs for each candidate node enables the algorithm to collect sufficient observations on the underlying network conditions in the presence of varying network delays. Please note that these TBM transmissions occur in cycles, with each cycle consisting of $|Z|$ transmissions for a total of L cycles.
3) A node receiving L TBMs from the same predecessor will send a Child Attach Message (CAM) to the predecessor and repeat the process described in step 2 with its own candidate set after appending its node identifier to the Tabu List B in the TBM. The Tabu List B in the TBM helps units to construct more efficient candidate sets and loop-free trees by excluding previously used units in the tree construction process (i.e. $Z = N \cup D - \{s\} - B$). In addition, the node sending the CAM will send a copy of the CAM to the root to inform it of the new attachment.
4) A node receiving a CAM will add the sending node to its child set C. Once s receives $|M|$ attachment notifications, it will start using this new tree.
5) Data transmission proceeds by each node retransmitting data arrivals to nodes in its C set.

In the worst case, the number of TBMs sent for tree construction is $O(|L|(|N| + |D|)^2)$. However, in practice the actual number of messages will be much smaller with the implementation of the systematic selection of the candidate set Z, which makes the protocol suitable for small-to-medium-sized networks. It would be expected that our algorithm scales well for networks of up to several hundred nodes.

B. Tree Maintenance

The TCA builds the fastest tree to deliver messages from s to M at the time of construction. However, network dynamics and variations in the traffic load may change the optimality of the tree for the current conditions. In order to enable tree adaptation due to performance degradation, our protocol includes the following tree maintenance behavior:

1) The root tags data messages with their creation time and leaf units return the tag to the root in a Tree Information Message (TIM), which enables the root to measure the round-trip time (RTT) to leaf nodes. In order to limit protocol overhead, we can limit the sending rate of TIMs from leaf nodes to the root: one approach would be to allow up to γ messages per second for each leaf node, where γ is a constant and predetermined value. The exact setting of γ depends on a multitude of factors, such as available bandwidth at the network, inter-traffic load, and sending rate of data from the source to the receivers.
2) The root uses the RTT measurements to update parameters rtt_{lt} and rtt_{ma}, which track the long-term worst message RTT average (using an exponential average) and the short-term average (using a moving-point average of the last h samples), respectively. h controls history length for rtt_{ma}, where large values of h provide a smoother average at the cost of reactiveness to latest network conditions, and small values of h provide more reactive tree maintenance, possibly at the cost of more frequent tree reconstructions and therefore protocol overhead.
3) A tree reconstruction is triggered with probability $P : f(\gamma_0, rtt_{lt}, rtt_{ma})$, where γ_0 is a prefixed value that limits the tree reconstruction rate. In this study, a tree reconstruction is triggered with $P = 1$ whenever $rtt_{ma} >= 2rtt_{lr}$ and $\gamma_0 = 1/15$.

IV. Implementation and Experimentation

A. Implementation

The protocol was implemented to support the functionality of a multicast overlay with the possibility of using different transport protocols in the substrate network (e.g., UDP and TCP with or without SSL support). In addition to implementing the overlay multicast algorithm itself, a protocol was developed to allow:

- The dynamic subscription to and release from the multicast tree. New nodes are allowed to attach to the root node and then are appropriately relocated down the tree on the next tree reconstruction cycle. It is interesting to note that in most cases, these new attachments would induce the tree to change because of the performance impact that they have on existing nodes in the tree, since the root now sends data messages to these new nodes as well, which will affect the message delivery latencies of other nodes.
- Information dissemination about nodes in the tree by the root node, which occur either when the protocol starts managing tree configurations or changes occur to the tree (e.g. when new units attach to the tree).
- Explicit Child Release Messages (CRM), which may be sent by nodes upon reception of unexpected data from a predecessor. These messages help to maintain a consistent tree under the presence of malfunctioning nodes.
- Tree Reconfiguration Requests (TRR), which are sent to the root by nodes which have been unexpectedly disconnected from the tree because of a predecessor failure or disconnection. A TRR will be sent by a node to the root if the node detects that one of its children is not longer available (e.g. due to node or link failure).

Furthermore, the current implementation can expose node state as a web service, which allows convenient remote inspection and monitoring of the protocol operation for testing and verification purposes.

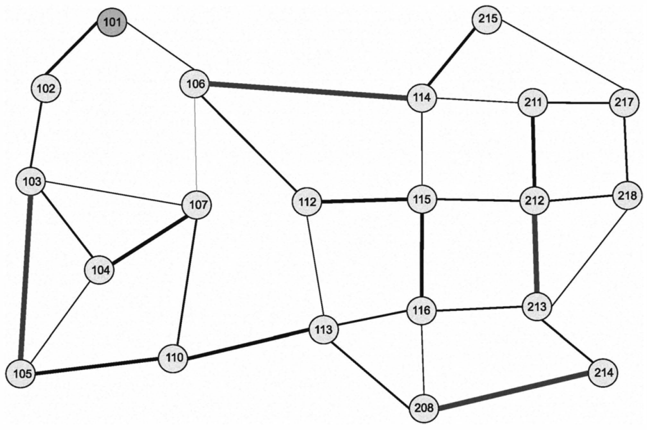

Figure 1. The 21-node testbed used in our experiments. Link costs are proportional to line thicknesses

B. Experimental Results

In this section, we present our experimental results regarding the proposed protocol's ability to rapidly adapt to changing network conditions. We ran emulation experiments on a real-life networking testbed which consists of 21 routers running Linux and connected via 10 Mb/s links. Figure 1 shows our experimental network topology, consisting of 21 nodes and 33 links. To further emulate the properties of wide area networks, random delays were added to the links according to a Pareto distribution, with average values in the range of 20 ms – 80 ms. These delays are shown proportional to line thicknesses in figure 1. *Open Shortest Path First* (OSPF) dynamic routing protocol is deployed as the network-layer routing protocol in the network.

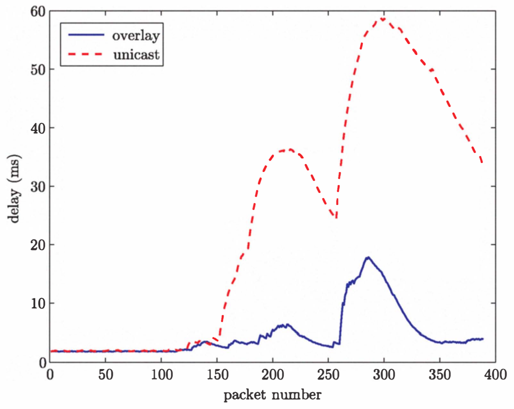

Figure 2. Measured message round trip delay to leaf nodes, $|M| = 8$

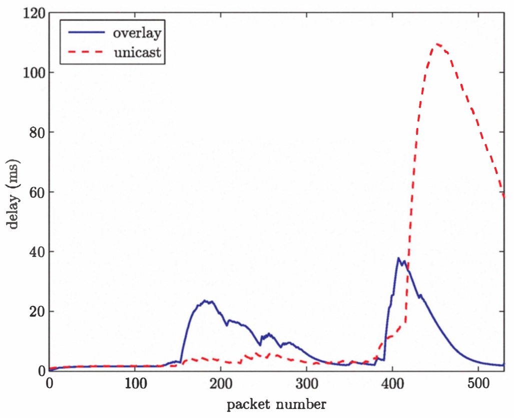

Figure 3. Measured message round trip delay to leaf nodes, $|M| = 6$

Each node in the network can act as either a router or an end-host that participates in the emergency multicast session. Node 101 (highlighted) acts as the source (root) in our experiments, which produces a traffic flow of 2600-byte messages at approximately 1 msg/s. This message length accounts for all protocol and network headers. In these experiments, TCP was used as the transport protocol, so measured message delays are affected by network conditions (i.e. network traffic) and TCP flow and congestion windows. In addition to the artificial delays introduced on the links, background traffic is generated in the experiments to further emulate a real-life emergency networking setting with inter-traffic.

In the first experiment, the multicast group consists of 8 members. The experiment was run for 30s without background traffic and then a flow of rate 5 Mb/s was injected on link 106-114. After another 30s, link 106-112 was saturated temporarily with a flow of rate 10 Mb/s. Figure 2 depicts the average round-trip latency for leaf nodes over time for a single experiment run, for both overlay multicast and unicast. The resulting plot shows the collective reports of all leaf nodes ordered by their arrival time at the root unit. It

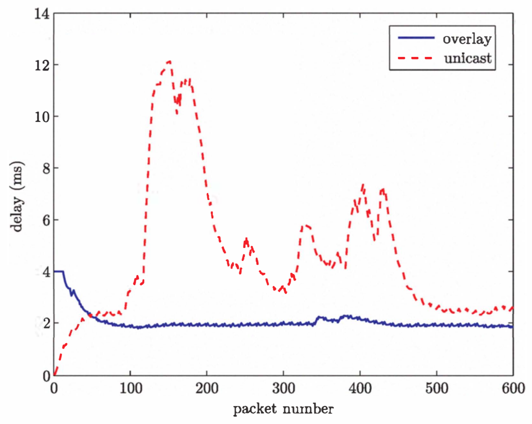

Figure 4. Measured message round trip delay to leaf nodes, $|M| = 12$

can be observed that compared to unicasting, our protocol is both more adaptive and performs better (with lower latencies for leaf nodes) in changing network conditions.

A second experiment was conducted for a group of 6 units, where an interfering flow of rate 1 Mb/s was injected on link 106-112 and later increased to 10 Mb/s. The results of this experiment are depicted in Figure 3, which shows that our protocol responds to both flows by adapting the delivery tree. It can be observed that our protocol has worse performance than unicasting for a short time after the injection of the light flow; this is due to the latency introduced by tree reconstruction (i.e. due to protocol overhead). This indicates that further investigation is required for the appropriate selection of protocol parameters.

Finally, we studied the performance of the protocol with a larger group of 12 nodes and in the presence of multiple interfering flows of rate 1 Mb/s. The resulting average message latencies are depicted in Figure 4. Although our protocol initially took a longer time to setup the distribution tree, this time was well-spent as the protocol performs substantially better than unicast in the rest of the experiment.

V. Conclusions

This paper has introduced an overlay multicast protocol for emergency message delivery over unreliable and best-effort networks. A key property of the proposed protocol is self-adaptation, which is achieved by reconfiguring the delivery tree in response to changes in the network. Such changes are monitored by our protocol through a continuous but non-obtrusive reporting of message latencies by leaf nodes in the tree. Multicast tree construction is totally distributed and does not require external network monitoring support as in other approaches. Instead, the protocol implements a distributed mechanism that selects peers based on their measured suitability to forward messages in the tree. The resulting protocol produces loop-free trees, is simple to implement, scalable to groups of hundreds of nodes and able to produce low-latency data delivery trees. This paper has presented experimental results studying the performance of the proposed algorithm in a real-life network testbed, showing that the protocol can adapt quickly to changes in network conditions and delivers messages to receivers with lowered latency. These results have also revealed that further investigation regarding the appropriate selection of protocol parameters is required for optimized protocol performance.

Acknowledgment

This work was partially funded by the EU FP7 DIESIS project.

References

[1] P. Traynor, "Characterizing the limitations of third-party eas over cellular text messaging services," Georgia Institute of Technology, Tech. Rep., Sep. 2008.

[2] G. Gow, T. McGee, D. Townsend, P. Anderson, and S. Varnhagen, "Communication technology, emergency alerts, and campus safety," *IEEE Technology and Society Magazine*, vol. 28, no. 2, pp. 34–41, 2009.

[3] BBC News Online, "7 july report highlights failings," http://news.bbc.co.uk/1/hi/england/london/5046346.stm, Jun. 2006.

[4] Wikipedia, "Virginia tech massacre," http://en.wikipedia.org/wiki/Virginia_Tech_massacre, 2009.

[5] Nielsen Online, "The 10 most heavily used web brands account for 45 percent of all uk internet time," http://uk.nielsen.com, May 2009.

[6] The AMBER Alert Program. http://www.amberalert.gov.

[7] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service and architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, Jan./Feb. 2000.

[8] R. Frank, T. Scherer, and T. Engel, "Emergency group calls over interoperable networks," in *Proc. 11th IEEE International Conference on Computational Science and Engineering Workshops (CSEWORKSHOPS '08)*, Jul. 2008, pp. 349–354.

[9] K. K. A. Zahid, L. Jun, K. Kazaura, and M. Matsumoto, "Ip network for emergency service," in *Proc. 3rd international conference on Mobile and ubiquitous multimedia (MUM '04)*. College Park, MD, USA: ACM, 2004, pp. 165–170.

[10] T. Kikuchi, M. Noro, K. Yamazaki, H. Sunahara, and S. Shimojo, "Lifeline communication system in the internet," in *Proc. International Symposium on Applications and the Internet Workshops (SAINTW '04)*, Tokyo, Japan, Jan. 2004, pp. 236–242.

[11] M. Noro, T. Kikuchi, K. Baba, H. Sunahara, and S. Shimojo, "Qos support for voip traffic to prepare emergency," in *Proc. International Symposium on Applications and the Internet Workshops (SAINTW '04)*, Tokyo, Japan, Jan. 2004, pp. 229–235.

[12] G. Hasegawa, S. Kamei, and M. Murata, "Emergency communication services based on overlay networking technologies," in *Proc. 4th International conference on Networking and Services (ICNS '08)*, Gosier, Guadeloupe, Mar. 2008, pp. 159–164.

[13] A. Bahora, T. Collins, S. Davis, S. Goknur, J. Kearns, T. Lieu, T. Nguyen, J. Zeng, B. Horowitz, and S. Patek, "Integrated peer-to-peer applications for advanced emergency response systems. part ii. technical feasibility," in *Proc. Systems and Information Engineering Design Symposium*, Charlottesville, VA, USA, Apr. 2003, pp. 261–268.

[14] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 131–145, 2001.

[15] J. Liebeherr, M. Nahas, and W. Si, "Application-layer multicasting with delaunay triangulation overlays," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1472–1488, Oct. 2002.

[16] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "Almi: an application level multicast infrastructure," in *Proc. 3rd conference on USENIX Symposium on Internet Technologies and Systems (USITS'01)*, Berkeley, CA, USA, 2001, pp. 49–60.

[17] L. Mathy, R. Canonico, and D. Hutchison, "An overlay tree building control protocol," in *Proc. 3rd International COST264 Workshop on Networked Group Communication (NGC '01)*. London, UK: Springer-Verlag, 2001, pp. 76–87.

[18] D. Helder and S. Jamin, "End-host multicast communication using switch-trees protocols," in *Proc. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, Berlin, Germany, May 2002.

[19] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: a framework for delivering multicast to end users," in *Proc. INFOCOM '02*, vol. 3, NY, USA, Jun. 2002, pp. 1366–1375.

[20] Y. hua Chu, S. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1456–1471, Oct 2002.

[21] P. Francis, "Yoid: Extending the internet multicast architecture," http://www.isi.edu/div7/yoid, April 2000.

[22] Z. Li and P. Mohapatra, "Hostcast: a new overlay multicasting protocol," in *Proc. IEEE International Conference on Communications (ICC '03)*, vol. 1, May 2003, pp. 702–706.

[23] Y.-h. Chu, A. Ganjam, T. S. E. Ng, S. G. Rao, K. Sripanidkulchai, J. Zhan, and H. Zhang, "Early experience with an internet broadcast system based on overlay multicast," in *Proc. USENIX Annual Technical Conference*, 2004.

[24] R. Lent, "Improving federation executions with migrating hla/rti central runtime components," in *Proc. 14th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD '09)*, Pisa, Italy, Jun. 2009, pp. 1–5.