Timely, Reliable, and Cost-effective Transport Service using Dissemination Graphs

Amy Babay
Department of Computer Science at Johns Hopkins University
babay@cs.jhu.edu

***Abstract*—. We present preliminary work that demonstrates the feasibility of deploying an Internet transport service that can support applications with stringent timeliness and reliability requirements (e.g. 130ms round-trip latency across the US with 99.999% reliability). We describe an approach to building such a transport service based on overlay networks and dissemination graphs. In this approach, each packet is sent over a subgraph of the overlay topology (a dissemination graph) that is chosen based on reliability, latency, and cost requirements.**

I. Introduction

The Internet natively supports end-to-end reliable communication (e.g. using TCP) or best-effort timely communication (e.g. using UDP). However, many applications require service that is both timely and reliable, and the demand for such communication services is increasing as new applications with strict timeliness and reliability requirements emerge.

Applications such as remote manipulation and remote robotic surgery bring severe constraints on timeliness. Human perception requires feedback to be received within about 130ms to be perceived as natural. This 130ms includes both the time for the command to reach the destination and for the feedback to be returned to the source of the command, translating to a latency requirement of about 65ms each way. Supporting such applications on a continent-wide scale is demanding: the network propagation delay across North America is about 35-40ms. In this work, we aim to develop technology toward supporting these applications.

In recent years, overlay network architectures have been developed to support applications with both timeliness and reliability requirements. These architectures use programmable overlay nodes in the middle of the network to enable hop-by-hop recovery protocols, rather than relying on the Internet's end-to-end recovery. Applications using these architectures include multimedia applications such as VoIP [1] and live television [2]. A live TV service, supporting interviews from remote studios, requires a one-way latency bound of about 200ms with a reliability such that no more than 10 out of 1 million packets do not arrive on time. A global overlay network with 10-20 well-situated overlay nodes can support such a service by using the 160-165ms available after accounting for a 35-40ms propagation delay to allow some buffering and recovery on overlay links. We were involved with a commercial service provider (LTN Global Communications) that uses this approach to support the TV and media industries [3].

In contrast to applications that can tolerate a 200ms one-way latency, for the demanding applications we are interested in, there is almost no flexibility to allow for recovery or buffering. Moreover, while techniques such as redundant sending along a single path and network coding can improve reliability, the combination of bursty loss on the Internet and the strict timeliness constraints of the target applications dramatically reduces their effectiveness. Thus, a different approach is needed.

For applications with such strict timeliness and reliability requirements, flooding on the overlay topology provides an optimal solution in terms of the quality of service it can provide. In this approach, each packet is sent on all possible paths, so it has the highest possible probability of reaching its destination within the time constraint. However, this approach is very expensive. Each time a packet is sent on a link, it incurs a cost. Since flooding requires each packet to be sent on every link, it incurs an extremely high cost.

A less expensive approach that does not reach the optimality of flooding is to send on k disjoint paths. For example, sending on two disjoint paths will cost slightly more than twice the cost of the single best path and will allow a packet to reach its destination as long as it is successfully transmitted along one of the two paths. Most existing systems that improve reliability by sending data redundantly over more than a single path use k disjoint paths (e.g. [4], [5]).

Sending packets on k disjoint paths allows for a coarse-grained trade-off between cost and reliability, as increasing k provides higher reliability at a higher cost. However, this approach uniformly invests resources along the paths from a source to a destination. This can be improved by investing fewer resources in more reliable parts of the network and more resources in less reliable parts of the network. In our experience, certain links in a network will have higher probabilities for loss than others, and certain regions will experience higher loss than others at particular times. By considering the loss characteristics of the network, we aim to provide optimal reliability subject to a given cost constraint, or conversely, to provide the minimum cost for a required level of reliability.

II. Dissemination-graph-based Transport Service

Our approach to transporting packets from a source to a destination in a timely, reliable, and cost-effective manner is to construct a dissemination graph based on the network topology, current loss characteristics, application latency constraints, and cost constraints. The dissemination graph is a connected subgraph of the overlay network topology that connects the source and destination. Each message from the source to the destination will be sent over all the edges included in the dissemination graph for that source-destination pair. Of course, if a message is lost and does not reach some node in the dissemination graph, that node cannot send it on its edges, even if they are included in the dissemination graph.

A. Problem Specification

The problem of selecting the best dissemination graph in terms of reliability, timeliness, and cost can be approached

from two different perspectives: maximizing reliability subject to a cost constraint or minimizing cost subject to a reliability constraint. In both cases, the dissemination graph is subject to the fixed topology of the network and a fixed timeliness constraint based on the requirements of the application.

To formally specify these two optimization problems, we introduce the following notation: We are given a graph G with vertices V_G and edges E_G that corresponds to the overlay network topology. Each edge $e \in E_G$ has a length l_e (corresponding to the latency of the link), a cost c_e, and loss rate p_e. [1] We are additionally given a source $s \in V_G$ and a destination $t \in V_G$.

We are concerned with the property we call the (s, t, L)-*timely-reliability* of G. In networking terms, the (s, t, L)-timely-reliability refers to the probability that a message sent by s reaches t within an application-specific latency requirement L. To consider the problem from a graph-theoretic perspective, we can define (s, t, L)-timely-reliability as follows:

Definition 1. (s, t, L)-TIMELY-RELIABILITY. *Let G_p be a graph obtained by randomly choosing to remove each edge $e \in E_G$ independently with probability p_e. The (s, t, L)-timely-reliability of G is then the probability that there exists a path from s to t in G_p of length at most L.*

To relate this definition to the intuitive networking definition, consider G_p to be the graph G as experienced by a particular packet. The edges in G_p correspond to links across which that packet can be successfully transmitted. The edges of G that are removed to obtain G_p are the links on which the packet would be lost. Note that G_p can be different for different packets. The (s, t, L)-timely-reliability of G gives us the probability that the G_p experienced by a packet allows it to reach its destination within its latency constraint.

We can now specify our two optimization problems:

Problem 1. MAXIMIZE TIMELY RELIABILITY. *Given a budget B, select a subgraph $H \subseteq G$ that maximizes the (s, t, L)-timely-reliability of H subject to the constraint that $\sum_{e \in E_H} c_e \leq B$.*

Problem 2. MINIMIZE COST. *Given a reliability requirement R, select a subgraph $H \subseteq G$ that minimizes $\sum_{e \in E_H} c_e$, subject to the constraint that the (s, t, L)-timely-reliability of H is at least R.*

In practice, the problem of maximizing reliability subject to a budget corresponds to the *customer* perspective of maximizing the quality of service received for a fixed cost. The problem of minimizing cost subject to a reliability requirement corresponds to the *service provider* perspective of minimizing the cost of providing an agreed upon quality of service.

B. Analysis

Without the latency constraint L, calculating (s, t, L)-timely-reliability, as specified in Definition 1, is exactly the classical two-terminal reliability problem [6]. This problem has been shown to be #P-hard [7]. Adding the latency constraint clearly does not make the problem easier, as in the case that the latency constraint is set high enough to include all paths (e.g. to the sum of the lengths of all edges), the problem is exactly two-terminal reliability. Since two-terminal reliability is at least NP-hard, it is almost surely intractable to compute (s, t, L)-timely-reliability exactly for an arbitrary graph.

Using the hardness of (s, t, L)-timely-reliability, we can show that Problems 1 and 2 are also at least NP-hard. If we could solve Problem 2, we could compute the reliability of a graph by solving Problem 2 with increasing reliability requirements until we find the point at which the requirement exceeds the reliability of the input graph and there is no solution to Problem 2.

Similarly, if we had an exact solution to Problem 1, we could compute the reliability of a graph by adding an edge between the source and destination whose cost is equal to the sum of the costs of all edges in the original graph. We set our budget equal to the cost of that edge, so the algorithm can either select that edge or the full original graph. We can then find the reliability of the original graph by increasing the reliability of that edge until the algorithm returns that edge as the maximally reliable solution.

Because of the hardness of these problems, we aim to find approximate solutions and explore techniques to make finding exact solutions feasible for some practical network topologies.

C. Solution Approaches

Flooding on the overlay topology provides an optimally reliable but very expensive solution. An initial approach that preserves the optimality of flooding at a lower cost is *time-constrained flooding*. In time-constrained flooding, packets are never sent to nodes from which they cannot reach their destination within the time allowed.

To determine the time-constrained flooding dissemination graph between a source and a destination, we first run Dijkstra's algorithm from the source to mark each node with its distance (in terms of network latency) from the source. We then run Dijkstra's algorithm from the destination to similarly mark each node with its distance from the destination. We then iterate over each edge in the graph. If the distance from one of that edge's endpoints to the source, plus the distance from the other endpoint to the destination, plus the latency of that edge is within the time constraint, the edge is included in the time-constrained flooding graph. Otherwise, the edge is not included. In this way, we select only the edges that are on some path from the source to the destination that is within the time constraint.

While time-constrained flooding is optimal in terms of reliability, it does not consider the cost of the dissemination graph beyond removing edges that do not improve (s, t, L)-timely-reliability. To find optimal solutions in terms of cost and reliability, we must restrict the size of the solution space to make the problem tractable for practical overlay topologies.

One way to restrict the solution space is to assume that all edge costs are equal. If a single Internet service provider is used, this is generally true. If multiple providers are used, link costs may vary, but the prices of different providers are

[1] Note that we assume that loss on different links is independent. While this assumption may not always hold in practice, it simplifies our model and still provides a good guide for dissemination graph design.

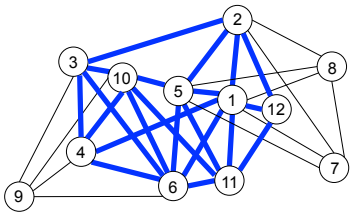

Fig. 1. Time-constrained flooding dissemination graph from node 1 to node 4 with a 66ms latency constraint (21 edges)

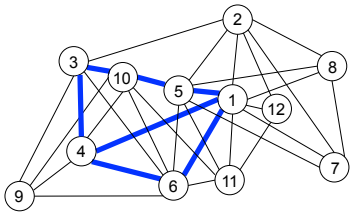

Fig. 2. 3-disjoint-paths dissemination graph from node 1 to node 4

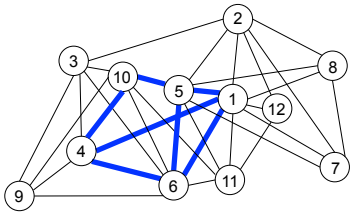

Fig. 3. Optimal 7-edge dissemination graph from node 1 to node 4 with 0.5% loss on all links

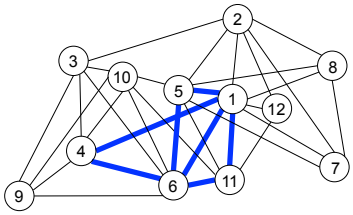

Fig. 4. Optimal 7-edge dissemination graph from node 1 to node 4 with 5% loss on node 1's links, 0.5% on other links

generally comparable. Thus, we simply use the number of edges in a dissemination graph as its total cost.

We further restrict the solution space by reducing the number of edges in the topology we consider. Time-constrained flooding eliminates edges that will not be part of any optimal solution, so we can use the time-constrained flooding dissemination graph as the input graph for our calculations, rather than the complete topology. This allows us to compute over a smaller graph without affecting the quality of our solutions.

Finally, we improve the trade-off between cost and reliability while also restricting the solution space by allowing packets to be recovered once, as long as the recovery can be completed within the time constraint. The 25-30ms available after accounting for propagation delay for our target applications is generally sufficient to allow a single recovery (across an overlay link with up to 12-15ms one-way latency). This technique provides higher reliability for the same budget, making the solution practical for the extremely high-reliability applications we consider. It also makes the computation more feasible, since we can find the optimal dissemination graph for a particular budget (number of edges) by considering every possible subgraph with that number of edges, but this is easier to compute when the budget is small. By reducing the budget needed to achieve our target reliabilities, this technique also reduces the solution space we must consider.

These restrictions make it feasible to compute optimal dissemination graphs in terms of reliability and cost for many practical networks. The introduction of a single recovery makes it practical to support extremely high reliabilities that have not been addressed in previous work aiming to optimize cost subject to reliability and latency requirements in overlay routing (e.g. [8]). Allowing a single recovery can improve reliability by 1-2 orders of magnitude without meaningfully increasing cost. In Section III, we present results from a case study of a practical global overlay network topology.

D. Implementation Considerations

The overlay network approach has been used to support applications with both timeliness and reliability requirements using programmable nodes running hop-by-hop recovery and overlay routing protocols. The availability of reasonably priced data center space around the world has recently enabled commercial services like LTN to use this approach to offer timely, reliable transport services over the Internet. Such techniques could also be used to run overlays on public clouds like Amazon Web Services.

Our lab has developed Spines [9], an open-source messaging infrastructure that is used to deploy overlay networks on the Internet. Spines includes support for source-based overlay routing, which we intend to use to implement routing based on the dissemination graphs we construct. In the source-based routing in Spines, a source stamps each of its packets with a bitmask specifying the edges in the overlay topology on which that packet should be sent. Each bit in the bitmask corresponds to one edge in the topology, and bitmasks can be recomputed dynamically based on changing network conditions. Spines currently supports flooding and k-node-disjoint-paths routing using these bitmasks. We are working to extend it to support routing according to dissemination graphs based on timeliness, reliability, and cost constraints.

III. Case Study Results

We present a case study of a practical overlay network topology. This topology includes overlay nodes in twelve data centers around the world that we have access to through LTN Global Communications. We deployed Spines on this topology and measured the latency of each link: the calculations we present here are based on those measurements. LTN guarantees 99.999% reliability (5 nines) and 200ms one-way latency [3]. We investigated whether we could achieve the same 99.999% reliability guarantee with the stricter latency constraint of about 130ms round-trip across North America. We present results for sending from a city on the East Coast (node 1 in Figures 1-4) to a city on the West Coast (node 4) with a 66ms one-way latency constraint (132 ms round-trip).[2] All reliabilities are calculated exactly using exhaustive search, assuming equal edge costs, restricting the input graph using time-constrained flooding, and allowing for a single recovery, as discussed in Section II-C. Note that the reliabilities reported here are higher than they

[2]We chose 66ms (rather than 65ms) because no dissemination graph could provide 99.999% reliability with the budget we consider (7 edges) and a latency constraint of 65ms in the most demanding loss scenario we consider.

would be in an actual deployment, due to the assumption that loss on different links is independent. In reality, correlated loss on adjacent links may reduce overall reliability.

First, we determined the dissemination graph for time-constrained flooding from node 1 to node 4 on the practical topology. The resulting dissemination graph is shown in Figure 1. When all edges have a loss rate of 0.5%, which is reasonable for the Internet, this graph is highly reliable, achieving over 12 nines reliability (99.999999999987%). However, it uses 21 edges, which is very expensive.

Since routing along k disjoint paths can achieve good reliability at a reasonable cost, we evaluated the reliability of a dissemination graph of 3 disjoint paths. We chose the 3 paths selected by the k-node-disjoint-paths routing in Spines, which selects paths so as to minimize the sum of their latencies. This dissemination graph is shown in Figure 2. It includes 7 edges and has a reliability of over 8 nines (99.99999912%) with 0.5% loss on each link.

While 3 disjoint paths provide good reliability, they do not necessarily provide an optimal solution. We calculated the optimal 7-edge dissemination graph in terms of timely-reliability by exhaustively checking all possible combinations of 7 edges out of the 21 edges of the time-constrained flooding graph in Figure 1. The resulting optimal graph is shown in Figure 3. This graph also includes 7 edges and provides over 8 nines reliability, but its exact reliability (99.99999974%) is slightly higher. Using this graph we would expect about 3 out of 1 billion packets to be lost, compared to about 9 packets out of 1 billion for the graph in Figure 2.

One of the benefits of our dissemination graphs is that they can invest more resources in less reliable parts of the network. Therefore, we also evaluated reliability when loss is not uniformly 0.5%. We considered a situation in which a problem in the vicinity of node 1 causes loss in that area and the loss rate on all of node 1's links increases to 5%.

In this case, the reliabilities of the dissemination graphs shown in Figures 2 and 3 drop below our target of 5 nines to 99.9986% and 99.9988%, respectively. However, computing the optimal dissemination graph for this case shows that it is still possible to achieve 5 nines reliability within our 7-edge budget if those edges are correctly selected. The optimal dissemination graph is shown in Figure 4 and achieves a reliability of 99.99974%. In fact, the graph in Figure 4 continues to provide 5 nines reliability even when the loss rate on node 1's links increases to 10%, while the reliabilities of the graphs in Figures 2 and 3 fall to about 99.98%. The reliabilities of all the graphs we evaluated in each of the three loss scenarios are summarized in Table I.

	Time-constrained flooding (Figure 1)	3 disjoint paths (Figure 2)	Optimal normal case (Figure 3)	Optimal problem case (Figure 4)
0.5% loss on all links	12 nines	8 nines	8 nines	6 nines
5% loss around node 1	11 nines	4 nines	4 nines	5 nines
10% loss around node 1	8 nines	3 nines	3 nines	5 nines

TABLE I. DISSEMINATION GRAPH RELIABILITIES UNDER VARYING LOSS CONDITIONS

IV. FUTURE DIRECTIONS

We intend to deploy a practical dissemination-graph-based transport service capable of supporting stringent latency and reliability requirements. The next step in this direction is to implement dissemination-graph-based routing in Spines and deploy this system with emulated loss rates and latencies to validate our model and reliability calculations. Ultimately, we plan to deploy the service over the Internet on a global scale using the overlay topology presented in the case study.

To provide a complete service, we are also working on developing algorithms to dynamically adapt dissemination graphs to changing network conditions. While we can compute optimal dissemination graphs for practical networks like the one in Section III, these calculations take tens of seconds, which is considerably slower than we would like to adapt to changing network conditions. We envision a two-stage approach in which we find an approximation of the best dissemination graph that we can use as soon as a change in conditions is detected, providing acceptable performance while the optimal graph is calculated. Since problems on the Internet can take minutes to hours to resolve, finding the optimal dissemination graph for current conditions can dramatically improve performance while maintaining a reasonable cost.

This preliminary work indicates the feasibility of deploying for the first time an Internet transport service that supports the stringent latency and reliability requirements of demanding applications such as remote manipulation on a continent-wide scale and at a reasonable cost.

ACKNOWLEDGMENT

I thank Yair Amir and Michael Dinitz for their guidance and collaboration in this work, as well as Emily Wagner and Amit Mehta for their contributions to the development of time-constrained flooding. This work was supported in part by DARPA grant N660001-1-2-4014. Its contents are solely the responsibility of the authors and do not represent the official view of DARPA or the Department of Defense.

REFERENCES

[1] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis, "An overlay architecture for high-quality VoIP streams," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1250–1262, Dec 2006.

[2] Y. Amir, J. Stanton, J. Lane, and J. Schultz, "System and method for recovery of packets in overlay networks," U.S. Patent 8 437 267, May, 2013.

[3] LTN Global Communications, "LTN Global Communications," http://www.ltnglobal.com, retrieved April 7, 2015.

[4] P. Papadimitratos and Z. J. Haas, "Secure message transmission in mobile ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 193 – 209, 2003.

[5] A. C. Snoeren, K. Conley, and D. K. Gifford, "Mesh-based content routing using XML," in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, 2001, pp. 160–173.

[6] C. J. Colbourn, *The Combinatorics of Network Reliability*. New York, NY, USA: Oxford University Press, Inc., 1987.

[7] L. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.

[8] K. Karenos, D. Pendarakis, V. Kalogeraki, H. Yang, and Z. Liu, "Overlay routing under geographically correlated failures in distributed event-based systems," in *On the Move to Meaningful Internet Systems*, 2010, pp. 764–784.

[9] Johns Hopkins Distributed Systems and Networks Lab, "The Spines messaging system," http://www.spines.org, retrieved April 7, 2015.