Building and Maintaining Overlay Networks for Bandwidth-Demanding Applications

Min Sik Kim
Advisor: Simon S. Lam
Department of Computer Sciences
The University of Texas at Austin
{minskim,lam}@cs.utexas.edu

***Abstract*— The demands of Internet applications have grown significantly in terms of required resources and types of services. Overlay networks have emerged to accommodate such applications. The performance of an overlay network is, however, highly dependent on its topology; keeping logical links in an overlay network from interfering with each other is crucial in achieving high throughput. In this thesis, I propose a novel technique that uses wavelet analysis to find bottlenecks caused by the interference between logical links. Based on this technique, I design an algorithm that eliminates all such bottlenecks from a multicast tree, and prove that there remains no such bottleneck in the tree upon termination of the algorithm. The scalability of the shared congestion detection technique is improved using an indexed multidimensional space, which reduces its computational complexity from $O(N^2)$ to $O(N \log N)$. This research will let bandwidth-demanding applications build more efficient overlay networks to achieve higher throughput.**

I. Introduction

As the Internet grows in scale, the demands of applications also grow in terms of required resources and types of services. For example, multimedia streaming consumes much more bandwidth than traditional applications, and can benefit from QoS or IP multicast if available. However, the availability of such services is very limited and unlikely to improve much in the future. Overlay networks have emerged as alternatives, where those services are implemented on top of IP.

While overlay networks are successful in circumventing the limitations of IP, building and maintaining an overlay network is still challenging. In an overlay network, participating hosts are virtually fully-connected through the underlying Internet. However, since the quality of overlay connections varies, the performance of the overlay network is dependent on which subset of connections are chosen to be utilized. Therefore, maintaining a "good" overlay network topology is crucial in achieving high performance.

Selecting overlay connections requires information on their characteristics provided by the underlying network. Those characteristics are estimated through network measurements. Hence, the goal of my thesis is two fold: to develop network measurement and analysis techniques to obtain required information, and to design an overlay network protocol exploiting the information to improve the overlay topology.

II. Research Summary

One of the most popular applications of overlay networks is overlay multicast, or end-system multicast. The initial motivation of overlay multicast was to overcome limited availability of IP multicast. In overlay multicast, however, data may be delivered multiple times over the same physical link because multicast forwarding is performed without support from routers. It may result in a bottleneck on the link, especially in applications demanding high bandwidth such as multimedia distribution. Therefore, it is critical to build an efficient multicast tree that can provide large bandwidth while avoiding such bottlenecks.

A straightforward way to maximize the bandwidth of a multicast tree is to measure bandwidth of overlay connections, and choose those with large bandwidth as tree edges. In my thesis, I present an overlay multicast protocol that builds a tree in which the average bandwidth from the root node, computed over all receivers in the tree, is maximized [1]. Each node in the tree estimates bandwidth from its ancestor nodes using the TCP throughput of 32 KB data transmission, and makes a decision on topology update to find a local optimum. A fundamental difference from other approaches using heuristics is that the multicast tree is always heading toward the global optimum. Though each node behaves for its own good based on local information, the tree approaches an optimal state as it evolves. Convergence of the tree to an optimal tree was proved using an abstract network model.

A drawback of the above protocol is that high average bandwidth does not necessarily mean high throughput. In overlay multicast, even when each tree edge has ample bandwidth, it is very likely that there is a bottleneck link caused by multiple data delivery over the same physical link, which throttles the throughput of the entire downstream receivers. If the overlay multicast system is able to identify such bottlenecks by finding out which tree edges are sharing them, it can change the overlay topology to avoid the bottlenecks and improve the overall throughput.

The basic primitive required for the bottleneck identification is to decide whether two paths are sharing a congested (bottleneck) link or not. There have been many techniques to detect shared congestion [2]–[4], but their major weakness is that they require that the two tested paths share an endpoint,

either at the source or at the sink. Thus, they cannot be used for general overlay networks.

I propose a novel technique, delay correlation with wavelet denoising or DCW, to detect shared congestion between two Internet paths [5]. The technique is based upon the following observations. Suppose each pair of probe packets from two source nodes arrive at the congested point simultaneously. Then clearly the one-way delays of the two paths measured by those probe packets should have high correlation between them. In practice, however, such synchronous behavior is impossible to achieve since source nodes use different clocks and the delays from source nodes to the congested point are random. Furthermore, the measured correlation is also affected by moderate congestion, if any, in non-shared portions of the paths. I discovered that these interfering delay variations can be filtered out with wavelet denoising. In addition, by selecting an appropriate wavelet basis function, I found that the technique works effectively for probe sequences with a "synchronization offset" as large as one second. This large synchronization offset is adequate to devise protocols to make the technique work for any two Internet paths that do not share a common end point. In contrast, previous approaches require a synchronization offset to be less than 50 ms, which limits their applications in overlay networks. Besides, simulations showed that DCW achieved faster convergence and broader application than previous techniques, while using fewer probe packets.

Once identified, bottlenecks can be eliminated by replacing edges passing through them with other, under-utilized overlay connections. However, it should be done very carefully. When a tree edge is cut in overlay multicast, another edge must be added to maintain connectivity. But the newly added edge may cause another bottleneck. Even worse, eliminating the new bottleneck may reincarnate the old one, resulting in oscillation. I designed an algorithm that removes bottlenecks caused by multiple data delivery, without incurring such oscillation. I proved that the algorithm always terminates, and that on termination there remains no such bottleneck in the multicast tree [6]. In a case where the source rate is constant and the available bandwidth of each link is not less than the source rate, my algorithm guarantees that every receiving node receives at the full source rate. In simulations with a network of a dense receiver population, our algorithm found a tree that satisfied all the receiving nodes while other heuristic-based approaches often failed. Furthermore, since the algorithm is very careful in changing the tree topology not to increase depth of the tree unnecessarily, its relative delay penalty [7] was close to that of the tree built by a greedy algorithm that optimizes delay.

The shared congestion detection technique can be used to find "better" paths in many other applications, such as overlay QoS routing, file download from multiple servers, and exploiting path diversity. However, its requirement that the shared congestion detection should be performed for every congested pair of paths limits its application to large-scale systems; $O(N^2)$ tests are required to detect all shared congestion among N paths. There have been studies to reduce complexity by performing per-cluster tests instead of per-path tests [3], [8], but they still need $O(N^2)$ tests in the worst case, and as much even in the average case if paths are widely spread as in large-scale overlay networks.

To reduce the computational complexity, I designed an efficient clustering algorithm that groups paths sharing the same bottleneck into the same cluster. The algorithm stores measurement data for each path into a multidimensional space, where data sets from paths sharing congestion are located closely to each other. Because data sets are indexed using a tree-like structure, adding paths and searching neighbors in the space take sub-polynomial time. Hence, the algorithm can reduce the computational complexity of shared congestion detection for N paths from $O(N^2)$ to $O(N \log N)$, making the shared congestion detection technique more scalable.

III. Conclusion

My thesis focuses on inferring network characteristics and improving overlay network topology using them. One of the most critical network characteristics for bandwidth-demanding applications is shared congestion, which tells whether two paths are sharing the same bottleneck or not. To detect shared congestion, I proposed a robust technique based on wavelet denoising, and designed an algorithm that improves the throughput of an overlay multicast tree by identifying bottlenecks using the proposed technique and eliminating them. I also proposed a clustering algorithm to make the shared congestion detection in large-scale systems feasible. The technique and algorithms will serve as a foundation on which future applications can achieve higher throughput by building more efficient overlay networks.

References

[1] M. S. Kim, S. S. Lam, and D.-Y. Lee, "Optimal distribution tree for Internet streaming media," in *Proceedings of the 23rd International Conference on Distributed Computing Systems*, May 2003.

[2] K. Harfoush, A. Bestavros, and J. Byers, "Robust identifi cation of shared losses using end-to-end unicast probe," in *Proceedings of the 8th IEEE International Conference on Network Protocols*, Nov. 2000.

[3] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *Proceedings of the 10th IEEE International Conference on Computer Communications and Networks*, Oct. 2001.

[4] D. Rubenstein, J. Kurose, and D. Towsley, "Detecting shared congestion of flows via end-to-end measurement," *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 381–395, June 2002.

[5] M. S. Kim, T. Kim, Y. Shin, S. S. Lam, and E. J. Powers, "A wavelet-based approach to detect shared congestion," in *Proceedings of ACM SIGCOMM 2004*, Aug. 2004.

[6] M. S. Kim, Y. Li, and S. S. Lam, "Eliminating bottlenecks in overlay multicast," in *Proceedings of IFIP Networking 2005*, May 2005.

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

[8] O. Younis and S. Fahmy, "Flowmate: Scalable on-line flow clustering," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, Apr. 2005.