Chapter 7

Conclusions and Future Work

In this thesis we addressed the problem of providing error control for two important classes of emerging Internet applications. These classes are (a) interactive continuous media applications (CM) and (b) large-scale multicast applications.

7.1. Interactive Continuous media Applications

We proposed an error control scheme for interactive CM applications that uses retransmission, an approach which was deemed inappropriate until a few years ago. By using a number of techniques including playout buffering, gap-detection and fast, conditional retransmission, we showed through implementation and experimentation, that despite popular belief, such an approach is not only feasible but also highly effective in reducing observed loss. Our experiments recorded improvements of several orders of magnitude, without any violation of the latency constraints of interactive CM applications.

7.1.1. Contributions

Our work on error control for interactive CM applications, was one of the first to demonstrate the feasibility of retransmission. Since then more work has been done by others that had lead to similar conclusions [5, 6]. Our contributions are as follows:

- We demonstrated the feasibility of retransmission by designing, implementing and testing a full-fledged continuous media transport protocol in the kernel of NetBSD Unix.

- We showed via analysis and experimentation over random, bursty and MPEG traffic, that our protocol reduces observed loss by orders of magnitude without violating the timing constraints of continuous media applications. Moreover, we have shown that our scheme sends retransmissions only when they are useful, which incurs minimal cost.

- We subjectively verified the effectiveness of our scheme using a raw video stream, and the results were observed to be markedly superior to the results without retransmission.

In summary, we believe we have presented strong evidence that retransmission can be effective in reducing loss in a wide range of CM applications and produced a real protocol to prove our hypothesis.

7.1.2. Future Work

We have already identified one area of future work in Chapter 2. We repeat it here and list more possible areas:

- **Extending to multiple retransmissions**: An important enhancement of our protocol is to extend it to attempt multiple retransmissions. Such an enhancement will be of great value in environments where the propagation delay between the sender and the receiver is small; examples include LANs and Campus networks. Having presented the design of this extension, we can implement and test it similar to the basic scheme.

- **Large-scale, continuous media multicast**: We can merge the error control schemes we presented in this thesis to develop an error control scheme for large scale CM multicast applications. This is another important class of applications, which includes distance learning, conference meetings, and multi-player games. We believe that LMS can provide a solid platform on which to develop such a scheme, because it provides the lowest recovery latency of all retransmission-based[1] schemes.

- **Enhancing CM applications to use protocol assistance**: Currently, most CM applications do not assume any help from the transport layer to aid in concealment; however,

1. excluding those using a FEC/retransmisison combination.

our protocol provides information that potentially can be very valuable to an application. We can modify an existing application to use protocol assistance and evaluate the benefits.

7.2. Large-Scale Multicast Applications

In order to solve the problem of scalable multicast error control, we enhanced the IP Multicast service model with a set of Light-weight Multicast Services (LMS). LMS enhances routers with a novel, yet simple forwarding functionality, which enables routers to structure receivers in a hierarchy with minimal or no involvement from receivers. A hierarchy, is an efficient way of achieving scalability. Locating LMS at the routers allows automatic placement of receivers on a recovery tree that exactly mirrors the underlying multicast tree, greatly improving the efficiency of error control. Schemes that do not use router assistance in building the recovery tree, resort to imprecise heuristics which degrade efficiency. LMS allows children to easily communicate with their parents to request retransmissions without implosion; it also allows parents to easily identify the subset of the children that suffered loss and accurately aim the retransmission.

We have demonstrated through simulation that with LMS, the performance of error recovery improves significantly over non-assisted schemes: implosion is eliminated, exposure is kept at negligible levels, and recovery latency is nearly optimal. In addition to improving performance, LMS greatly simplifies receivers by freeing them from the burden of topology discovery. We have also demonstrated through implementation and experimentation that LMS is easy to implement, integrates well with the existing multicast architecture, and its performance is as good or better than normal multicast.

7.2.1. Contributions

Our approach to the problem of scalability in error control for large multicast applications came from a new angle. We realized that most of the complexity in existing solutions comes from the difficulty in dealing with a dynamic, but largely unknown group topology. Without knowledge of topology, it becomes hard to coordinate group members to avoid problems like implosion, and certain penalties must be endured in order to achieve scalability. It is perhaps not an exaggeration to say that a disproportionately large part of the effort expended in most reliable multicast protocols proposed today, deals with gathering information about topology, either through measurement or heuristics. Our research showed how to solve this difficult problem as follows:

- We made the observation that forwarding and error control are two clearly separable components, and great benefits can be realized by decoupling and placing each one where it is more beneficial. We believe that the forwarding component belongs to the routers (after all this is what routers do best), and the actual error control component (detection, notification and recovery) belongs to the receivers. This separation is very clean, i.e., it does not violate any layering principles like the end-to-end argument [38] - the routers do not see any transport layer information and the endpoints know nothing about topology.

- We showed how to enhance the IP Multicast model to accommodate a new set of forwarding services (LMS) that are highly beneficial for efficient and scalable error control. The enhancements are in the form of simple, easy to understand high-level operations, like “send a request to the parent”, or “send a retransmission to part of the multicast group”.

- We designed and implemented the LMS algorithms, and showed that the programming effort involved is minimal.

- We designed and created a simulation of LTP, a transport protocol that uses LMS. We determined that by using LMS LTP becomes much less complex than non-assisted protocols. Its complexity is in fact on par with an equivalent unicast protocol.

- We implemented and verified the operation of LMS; our evaluations showed that the processing overhead of LMS at the routers is at least as good as regular multicast processing, thus dispelling any doubts as to whether LMS can be implemented in the routers. Our implementation showed that LMS requires minimal changes to existing code (about 250 lines of new C-code)

- We used simulation to evaluate the performance of LTP/LMS and compare it with SRM and PGM; we demonstrated that the performance of the LTP/LMS combination is three to ten times better than both these protocols, in terms of latency. In terms of exposure, LMS is only slightly worse than PGM, which eliminates exposure. SRM without local recovery suffers dramatically from exposure compared to LMS, which offers orders of magnitude lower exposure than SRM.

- We discussed how LMS can be useful for other applications, like building an ACK-based protocol, scalable voting, simple anycast and maintaining congruency in other tree-based protocols.

In summary, one of the more significant contributions of our work is that it has provided us with a vantage point from where to try to solve other important, yet difficult problems like multicast congestion control.

7.2.2. Future Work

We have already discussed some areas of future work in LMS, in Chapters 4 and 6. We recap our discussion here, and add some further thoughts:

- **Replier State**: As we discussed in Chapter 6, we did not implement the LMS component responsible for propagating the replier state in the network (the LMS-HIER component). We sketched an approach on how to implement this component, utilizing small modifications to IGMP and the multicast routing daemon (mrouted). Completing the design and implementation of this component would be an important addition to LMS.

- **Incremental Deployment**: It is highly unlikely that LMS can be deployed in all the routers on the Internet at once. Thus, any scheme like LMS that proposes modification to the existing multicast model, must develop a plan for incremental deployment. We have discussed an incremental deployment approach in Chapter 4, using source-path messages, similar to PGM. In addition, we believe it is possible that LMS can gain leverage off the plan for deployment of IPv6[62]. We have not worked out the details of such a task yet; however, it is important to study its feasibility and propose a deployment plan.

- **LMS in the router fast-path**: In Chapter 6, we discussed the possibility of introducing LMS into the hardware fast-path of certain high-end routers. We claimed that since we have demonstrated that LMS is no more complex than normal multicast forwarding, it is very likely that it can be implemented in hardware. We can verify our claim by studying the architecture of such routers and determining the feasibility of such a venue.

- **Security**: As we discussed on Chapter 4, LMS introduces no security holes in the current IP model. However, further studies may be conducted to determine how to integrate LMS with currently proposed security schemes.

- **Other applications and services**: We have claimed that LMS offers services that are general enough to be used by other multicast applications. LMS, as currently presented, enables general services including a *scalable collect service*, which enables scalable feedback from any number of receivers to the sender, and *fine grain multicast*, where a message can be accurately aimed towards a subset of the receivers in a multicast group. Other general services can be implemented on top of the LMS services, and a detailed study can be carried out to define them. In addition, encouraged by the clean separation of functionality that LMS achieves with error control, we can explore the possibility of doing the same for other difficult applications, for example congestion control. In the process, we may define other useful building blocks to add to LMS.

- **Integration with existing services**: There have been some recent proposals for router services to improve the performance of existing protocols, including ECN[59], RED[60], and CBQ[61]. We can investigate the possibility of integrating LMS with such services.

7.3. Source Code Availability

The source code for our *ns* simulations and Implementation can be obtained by contacting the author at http://www.ccrc.wustl.edu/~christos.

7.4. Closing Remarks

Will multicast ever become ubiquitous in the Internet? That is a question whose answer seemed clear a few years ago, when researchers believed in the obvious advantages of multicast. Despite the initial optimism, multicast has not taken off yet. Deering, the chief architect of IP Multicast along with many other prominent researchers, have recently expressed skepticism and disappointment[84] on the current state of multicast, but maintained their optimism about its fate. There are many theories about why multicast is off to a slow start, ranging from the lack of a "killer application," fears from Internet Service Providers (ISPs) about users flooding their networks, to ISPs not knowing how to charge for multicast services. Recent work has tried to address some of these issues. Express [82] and Simple Multicast [83] are two very recent approaches that propose to simplify the existing multicast model to satisfy some of the ISPs' concerns.

We are at a point where the Internet has been heavily commercialized, and other issues are beginning to emerge that were mostly brushed away when the Internet was a predominantly academic entity. These include billing, security, routing policy, and more. The Internet has also grown so big, that making any fundamental changes is becoming increasingly difficult (some will say impossible). This is probably the major criticism of our work: even if LMS satisfies the skeptics, will it be possible to deploy it? Our answer is that Cisco is already attempting to do just that with PGM, which faces similar obstacles. Thus, it may still be possible to make changes inside the network, we just hope that you do not have to be a giant router company to do so.