Review of Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism
By Holly Esquivel

Summary

The authors of this paper present a new architecture and mechanism for real-time applications on networks. They call their protocol ISPN architecture, which allows for both guaranteed and predicted service of real-time applications. They explain that there are four main issues that their architecture was trying to solve: a commitment for service, a interface in which the source and network could agree on parameters, a packet scheduling mechanism, and a way to establish service and traffic agreements. Real-time application characteristics are explained along with the types of delay these packets face such as propagation, switch transmission and queue delays. After looking at delay, they saw that real-time applications have different limits on the amount of delay they can experience and thus, can sometimes be bursty. They develop/integrate different algorithms to help achieve these types of services to different application flows. For guaranteed service they propose using a token bucket , but more specifically a WFQ algorithm. This allows for a fluid flow model where one flow can only overwhelm itself and hurt its own delay. Thus the jitter a flow might experience should only be proportional to the burst. For predicted service they utilize a modified FIFO+ algorithm. This way the jitter experienced is spread over a number of flows, thus allowing for bursty traffic. Although there is no isolation this keeps the one flow from having a majority of its packets surpass the play-back point at the cost of a few packets of other flows being delayed. They then explain an adaptation to the FIFO+ algorithm which goes in packet headers to assist flows from being punished if they are on multi-hop routes. They decide two different service interfaces which service each different service model and how the service models can be combined to form one unified scheduling algorithm. They simulate their algorithm and show it is able to achieve fairly low delay.

Pros

- They give a detailed description of what real-time applications they are trying to accommodate and how their traffic is different from other network traffic.
- They simulate their algorithm in comparison to two other common algorithms on the tested network topology.
- They successfully describe and define the idea of two types of real-time network flows – predicted and guaranteed.
- The importance of Admission Control is described even if they don’t know how exactly to handle it.

Cons

- They present a feasible architecture, but they seemed evaluate it poorly because of the lack of experiments they had showing the performance the system could achieve (other than a few simulations).
- The network topology in Figure 1 seemed unrealistic of a network topology we would see today that uses real-time applications.
- The overhead associated with adding/calculating the information for the multi-hop sharing doesn’t really seem evaluated. They say the benefits outweigh the overhead, but it doesn’t say if this can be done successfully at line speeds.