Sprout Summary

Sprout is an end-to-end transport protocol designed for interactive applications, like video conference and live streaming, over cellular networks, which achieves high throughput and low delay. The novelty of Sprout protocol arises from two aspects. Firstly, it uses the receiver's observed packet arrival times as the congestion control signal, rather than the packet loss, roundtrip time, or one-way delay. Secondly, over the observed packet arrival times, it models the cellular channel capacity using a Poisson process with an underlying rate lambda, whose distribution changes over time. Sprout uses Brownian motion to model the evolution of the probability distribution on lambda. Based on these stochastic models, Sprout proposes a control protocol to quickly adapt the sending rate of a flow. By utilizing the above designs, Sprout aims at avoiding the queue buildup problem suffered by existing congestion control algorithms over the highly variable cellular network. Compared with existing video conference applications, Sprout reduces the end-to-end self-inflicted delay by a factor of 7.2-8.7, and achieves a factor of 1.9-4.4 improvement in the throughput.

In COS563, our discussion mainly involves the background, the stochastic model design, and the evaluation. We first focused on the background. Cellular wireless networks are highly unpredictable compared with wired networks, where each device experiences a different time-varying bit rate because of variations in the wireless channel, especially when the device is mobile. Hence, it makes more sense to use statistics at the receiver not the traditional congestion control signals. In addition, another distinguishing feature that contributes to the performance gain of Sprout is that cellular carriers generally keep a separate queue for each device in a cell. The separate queues make the end-to-end delay being dominated by self-interaction, not cross traffic. However, this requires the assumption that the bottleneck is at the cellular links. If the bottleneck is in the upstream network, where shared queues are adopted, Sprout might not correctly forecast the channel capacity, and adopts an inappropriate sending rate. Recently, given faster and faster access link, cellular access links might not be the bottleneck anymore, e.g., when Verizon customer watch Netflix the bottleneck could be at the peering link.

As for the design part, we first discussed its general idea of learning a stochastic model for rate adaptation. Sprout models the cellular links using a Poisson process, where the rate lambda of the Poisson varies over time in a Brownian motion in normal case. However, we think that inferring the rate in outage case is less than perfectly presented in section 3.1. The design part would be clearer if the authors can explicitly discuss the model in normal and outage two states, and give corresponding mathematical descriptions. In addition, Sprout assumes that the rate lambda is at most 11Mbps, which doesn't hold in current cellular networks. With the evolution of cellular networks, both the range of lambda and the discretization scheme need to be reconsidered, when applying Sprout. For the evaluation section, we discussed the experiment setup, where Sprout uses a trace-driven emulation for the sake of comparing multiple schemes over the same real-world condition. We also notice that in Figure 7, the points in figures only show the mean value of throughput and delay. However, the variation also matters when evaluating a control protocol. A more comprehensive results would be presenting the variation range of throughput and delay of different schemes in the figures.