# Sharing network state with application endpoints

Andreas Terzis
Chris Bentzel
Google

Mobile operators have to satisfy their customers' needs while managing their limited network and spectrum resources. The general term used to describe these management techniques is *bandwidth optimization*. In turn, bandwidth optimization techniques use knowledge about the mix of applications using the network. In some instances, Deep Packet Inspection (DPI) devices have been used to look at transport- and application-level headers to map data flows to applications, subscribers, and content providers. However, as the Internet moves to end-to-end encryption such mechanisms lose their effectiveness.

A lot has been said about how this encryption may impede network management, with some calling for limits on encryption that would impede innovation and privacy online. In contrast, we argue that it is possible to enable effective network management even where there is strong encryption. Application providers and network operators can collaborate to achieve this goal. We advocate for a model where operators selectively share network state to application endpoints and applications adapt their behavior to varying network conditions. In what follows we outline some of the proposed mechanisms and preliminary results from lab and field trials.

## Throughput Guidance

We recently proposed the concept of Throughput Guidance (TG), whereby a function in the Radio Access Network (RAN) of a mobile operator provides real time information about the throughput estimated to be available at the radio downlink interface between a UE and the base station the UE connects to. The transport stack can use this explicit information to inform several congestion control decisions. For example: (1) selecting the initial window size, (2) deciding the value of the congestion window during the congestion avoidance phase, and (3) reducing the size of the congestion window when the conditions on the "radio link" deteriorate. With this additional information, TCP can reduce its reliance on probing for additional resources and heuristics to reduce its sending rate after a congestion episode. Preliminary field tests in a production LTE network showed that TG reduces YouTube join latency, defined as the amount of time until the video starts playing, by 8% on average, re-buffering time by 20% on average, and rebuffer count by 2% on average. In addition to improving quality of experience for users, TG improves the utilization of providers' networks and while reducing bufferbloat. While promising, a number issues covering the discovery, safety, security, and general applicability of the TG mechanism in the broader Internet need to be resolved.

## Link Capacity Estimation

While TG provides information from the RAN, available link capacity is also available on the UE. The issue has been that while this information is available at the radio, it has not been exposed so far to applications. We have recently worked with radio vendors to expose this Link Capacity Information (LCE) to Android and provide a new interface as part of Android M. As a proof of concept we integrated LCE information to the QUIC protocol whereby the sender uses the LCE estimate to set its sending rate. Our preliminary evaluation showed that the resulting protocol improves throughput over TCP by 1.08–2.89x for small and medium flows. TCP achieves comparable throughput for large flows but at the expense of 2.38-2.65x higher RTT.

### Congested cells and Off-Peak APIs

TG and LCE optimize how applications use the operator's network resources. But there are times when these resources come close to exhaustion as application demand peaks during the day. Google has recently developed an API that allows a network to expose coarse information about its congestion state. Our initial implementation of this *Congested Cells API*, allows mobile operators to declare the congestion level of individual cells in their network with minute-level granularity. Congestion level in this context corresponds to the overall cell load. For example, it could correspond to the fraction of physical resource blocks (PRBs) in use. Communication happens over an authenticated HTTPS channel established between control functions inside operators and application providers. This explicit information will allow applications to bootstrap their adaptations (e.g., initial video data rate selection) and allow mobile operators to shed load through application cooperation.

Operators can use the same API to notify applications when (parts of) their network become underutilized (i.e., enter an off-peak period). The term off-peak in this context means that the utilization of the network falls below an operator-defined threshold. The implication of a network entering an off-peak period is that applications will schedule delivery of pending requests. For example, over the air updates (OTAs) can be scheduled when the network goes off-peak. A network provider can limit the total volume of off-peak traffic and a scheduler inside application provider's network uses the provided limits to coordinate pending transfers from all clients. The off-peak API delivers a dual benefit for the operator: (1) reduce peak demand, while (2) utilizing the traffic trough.

### Mobile Data Plan API

Users in emerging markets are frugal with their mobile data plans. Applications should then be transparent in their use of the user's limited data budget. For example, an application might notify the user of operations with high data cost. Likewise, applications should customize their behavior based on the status of the user's data plan (e.g., postpone backups if the user is close to, or has gone beyond their monthly data cap). We are working on an open data plan API that allows applications to learn about the user's data plan in a privacy-preserving way and are working with equipment vendors and network providers to enable this functionality.