Toward end-to-end software defined QoE

Aditya Akella, UW-Madison

Large portions of our computing and network infrastructure are becoming software-defined -- our infrastructure is no longer limited by rigid, closed hardware, but instead is programmable via flexible software interfaces, and easily virtualizable and slice-able. Examples include networks including switches and NICs, storage systems, Internet exchange points, data centers, and clouds. In addition, a variety of recent innovations have introduced similar degrees of flexibility into networking stacks by enabling programmable control over offloading of network protocols to hardware accelerators. Finally, innovations in large scale overlay network technology have made flexible control over content distribution and routing possible. Together these three trends can offer unprecedented flexibility in joint control and orchestration of various portions of our network, storage, and compute infrastructure, giving rise to a variety of tantalizing possibilities.

In particular, we can now flexibly control configure *all aspects* of end-to-end flows, including both low level and high level issues such as: what kind of and how much compute resources they use (e.g., in the cloud or on mobile system), the specific backend systems flows invoke (e.g., personalization engines, anomaly detectors), the network paths they traverse and the specific attributes of those paths (e.g., the latency and bandwidth, or waypoints traversed), what kind of transport to use (e.g., leveraging in-network support vs. not), whether or not hardware support is leveraged to accelerate packet transmissions, and how the content being accessed is actually delivered (e.g., transcoded vs. not, encrypted vs. not, from a local cache vs. a remote data center). These attributes can be changed on the fly as needed according to, e.g., a user or application's need, much better than today.

It is my strong belief that such flexibility and control can form the basis for ensuring optimal end-to-end quality of experience (QoE) for user transactions, or "flows". Existing approaches for QoE focus on specific applications (e.g., video or the Web), and specific control knobs that operate at a particular layer (e.g., selecting the optimal CDN server, or performing traffic classification/engineering). This gives rise to at least two problems: first, it leads to silo-ed solutions, where techniques developed for one application cannot be easily applied to another equally important application and have to be engineered from the ground up. Second, focusing on individual components or layers can lead to mechanisms that are either highly sub-optimal, or have poor interactions with other mechanisms operating at a different layer.

My vision is to develop a new framework that enables end-to-end software defined control over flows and all their relevant attributes. I envision a system that works on the basis of a model of end-user QoE as a function of the different high- and low-level aspects outlined above. Such a model can be learned over time and refined as new data points emerge. During a flow's lifetime, or before a flow starts, the model can be used to determine the necessary actions to take, e.g., change one or more aspects of a flow's transmission or delivery, in order to ensure a positive improvement in end-user QoE. The actions can then be realized via suitable software defined interfaces.

Naturally, realizing this vision entails overcoming several technical research challenges: developing APIs that allow unified control over multiple infrastructure components and protocols; building robust and scalable control planes; developing predictive QoE models; leveraging the models to obtain real-time insights; orchestrating control over many flows as flows come and go; developing algorithms for updating attributes of multiple flows at once; ensuring that multiple administrative entities can co-existing in offering services for end-to-end software defined flows; and ensuring the system can operate effectively in an inter-domain setting.

To test various aspects of this research, we need a federated testbed with compute resources spanning data centers, end-hosts and mobile devices, software-defined networking and storage support at different locations, and software-defined exchanges. In addition, we would need the ability to conduct deep instrumentation of the experimental setup to drive various optimizations. This vision builds directly upon my prior work on developing predictive models of Internet video QoE [1], models for QoE optimizing network content delivery [2], and hardware acceleration [3]. It also leverages my prior work on software defined networking, cloud computing, and virtualization, as well as my experience with deploying and helping manage CloudLab, a large NSF-funded open cloud infrastructure.