

SDN API for Access Network Virtualization

John Matthews
and Craig Russell
CSIRO
Marsfield, Australia

Email: {john.matthews,craig.russell}@csiro.au

Vijay Sivaraman
UNSW
Sydney, Australia
Email: vijay@unsw.edu.au

I. SCOPE AND SIGNIFICANCE

Fixed-line Internet Service Providers (ISPs) are increasingly confronting a business problem. Residential data consumption continues to grow at 40% per annum while revenues are growing at less than 4% per annum, mainly due to flat-rate pricing. Meanwhile, content providers, who monetize their video offerings via ad-based or subscription based models are seeing a direct impact on their revenue from reduced user quality of experience (QoE). This demo shows how ISPs could exploit the service quality dimension (in addition to bandwidth and download quota), using software defined networking (SDN) technology, to differentiate their offerings and tap into new revenue opportunities. The demo network, shown in figure 1, is a greatly simplified emulation of content provider, ISP and home networks. The content provider is a single server providing video, FTP and web content. This is connected to the OpenFlow switch in the ISP. In reality the interconnecting network may include many internet administrative domains but its effect on content provider traffic is considered out of scope. A single access network switch emulates the ISP access network (including DSLAM, cable head-end or OLT devices) used to provide internet to the home. In reality some, or all, of the access network equipment may be operated independently from the ISP. The home internet connection is emulated using 10Mbps Ethernet, which is considered by this demo to be the single bottleneck for service delivery into the home. In reality other bottlenecks may exist within other parts of the ISP and access network; however these are considered out of scope. The home includes a large screen internet device for viewing streaming video along with two other devices used for web browsing, file transfers and viewing lower resolution streaming video. Our demo uses wired Ethernet connections within the home, rather than the more popular WiFi, simply to avoid interference from other demonstrations.

II. METHODOLOGY

Virtualization of the access network allows content providers to deliver quality content, such as streaming video or file transfers (typically software upgrades), without the complication of having to know about other traffic using the network. Unlike previous QoS schemes, such as DiffServ, the use of an API provides much greater agility for the provisioning of services. Furthermore our scheme requires no change to access network and home equipment. Content

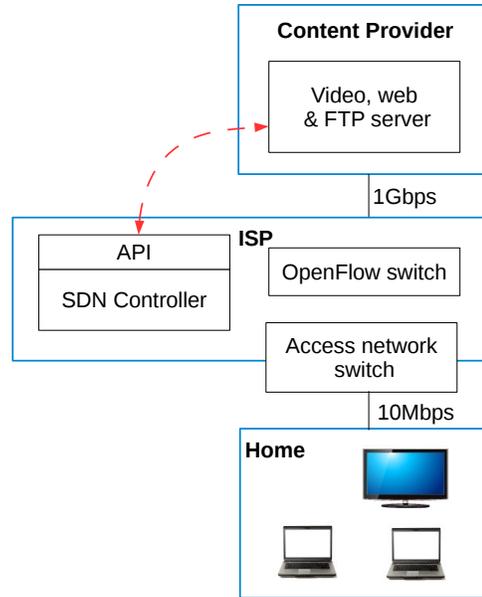


Fig. 1. The demo network

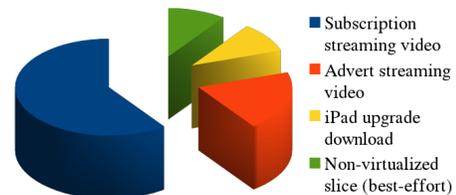


Fig. 2. Example home internet link slices

providers gain controls over the home internet link slices used to deliver their services. Figure 2 shows example slices for delivery of subscription streaming video, advert streaming video and iPad upgrade content. Homes retain control over the size of their internet link's non-virtualized slice, which operates as present on a best-effort basis. Thus, homes that dislike the thought of advertisements gaining priority over web browsing on their home internet link can simply increase the size of their non-virtualized slice.

TABLE I
EFFECT OF VIRTUALIZATION FACTOR ON VIDEO, BROWSING AND FTP

App	$\alpha = 0.0$		$\alpha = 0.8$		$\alpha = 1.0$	
	mean	std	mean	std	mean	std
7.5Mbps video MOS	2.87	0.44	3.10	0.31	3.25	0.01
2.1Mbps video MOS	3.25	0.00	3.25	0.01	3.25	0.01
Web page load (s)	2.84	0.86	3.10	1.61	4.85	3.55
FTP stretch	1.60	0.20	1.97	0.77	2.45	1.07

III. DEMONSTRATION EXPERIMENTS

In the first experiment we show the hit and miss video quality available at most homes today. We attempt to view a high quality subscription streaming video on the large screen home device at the same time as viewing lower quality advert streaming video and running a FTP file transfer on the other two home devices. Video quality is poor due to the large screen video requiring more than its TCP "fair share" of the home internet link. Common workarounds, available today, are to re-schedule the FTP download to run during the night or to compromise on video quality by switching to a lower resolution video on the large screen device. In the second experiment we show that it is possible for the user to achieve their desired video quality or file transfer duration using our ISP API. In this case the video server, which knows the user's video quality expectations, sends a bandwidth reservation request to the ISP API. For example, to request a 7.5Mbps minimum bandwidth slice of the internet link to home device 10.10.5.18, the JSON format request would be: { hello: jukebox, type: minbw, nwsrc: 10.10.7.31/32, nwdst: 10.10.5.18/32, proto: 6, sprt: 8080, dprt:22400. bw: 7500 } . If the ISP accepts this bandwidth reservation request then the flow is assigned to a dedicated minimum bandwidth queue and the ISP may charge the content provider for delivery. Similarly the FTP server, which knows the file size and the user's file transfer time expectations, sends a bulk transfer request to the ISP API. For example, to request a 1.8GB TCP file transfer to home device 10.10.5.19 that must complete within 1 hour, the JSON format request would be: { hello: jukebox, type: bulk, nwsrc: 10.10.7.31/32, nwdst: 10.10.5.19/32, proto: 6, sprt: 24380, dprt: 20, len: 1800000, deadline: 3600 } . If the ISP accepts this bulk transfer request then the flow is given a dedicated queue, treated as "elastic", and the user may be charged at a lower rate.

A bigger 3 home prototype system was built at CSIRO to demonstrate the scheme's feasibility and to quantify the effects of virtualization on video, browsing and FTP bulk file transfers. Each home ran 3 internet devices; a large screen that intermittantly showed a 7.5Mbps (peak rate) video, a small screen that intermittantly showed a 2.1Mbps (peak rate) video, and a home PC that either repeatedly downloaded a 1.1MB web page in Internet Explorer or performed FTP bulk file transfers. All devices were controlled by PowerShell scripts to emulate randomized user activity. A Markov process was used to control video playback. On average, 40% of time was spent idle and 60% playing video. Idle duration between web browsing and FTP bulk file transfers was randomized to

follow a log-normal distribution with mean 10 minutes and standard deviation 2 minutes. FTP files were chosen from a Pareto distribution with shape parameter 4.5 and scaled such that their sizes ranged between 100 and 500MB with high probability. Bandwidth reservation requests to the ISP API could only succeed when there was sufficient capacity remaining, at that time, on the internet link. Video quality was quantified by measuring mean opinion score (MOS) [1]. The results collected from many experiment runs are summarised in table I. The home user has a single control over internet link virtualization factor α with our scheme. This sets a limit on the total proportion of the link that can be virtualized (see figure 2 for an example). The results show that increased α leads to more consistent high quality video on the large screen but makes no difference to the small screen. The TCP "fair share" of bandwidth is sufficient for the 2.1Mbps video; as long as there are no more than four TCP flows over the 10Mbps internet link. Increased large screen quality comes at the expense of increased web page load times and FTP file transfers being stretched over longer time frames. However, the results also show that the user retains control over the trade off between consistent large screen video quality and fast bulk file transfer or browsing.

IV. EQUIPMENT, SPACE AND SETUP TIME

The demo requires the following equipment:

- 1 x Desktop PC with no screen or keyboard which acts as the content provider server.
- 1 x Pica8 OpenFlow switch.
- 1 x 10 port 1000/100/10Mbps Ethernet switch.
- 3 x Macbook Pros act as the home internet devices.
- 1 x 64 inch screen + mouse keyboard used as the display for one of the Macbooks

The server, Ethernet switch and one Macbook are located under the table. Other devices, including the Pronto switch sit on top. At least 1.2m of bench width is required. I estimate about 1 hour of setup time - including time to transport equipment from car to desk. This demo requires power - 6 sockets. It is self-contained with no internet access necessary. However, WiFi internet access would be useful to help with the running of the demo.

REFERENCES

- [1] R. Mok, E. Chan and R. Chang, Measuring the quality of experience of HTTP video streaming, In Proc. IFIP/IEEE Int'l Symp. on Integrated Network Management, May 2011