User Control of Quality of Experience in Home Networks using Software-defined Networking

Himal Kumar\textsuperscript{1}, Hassan Habibi Gharakheili\textsuperscript{2}, Vijay Sivaraman\textsuperscript{2}

\textsuperscript{1} Indian Institute of Technology, Patna, India
\textsuperscript{2} University of New South Wales, Sydney, Australia
Overview

- This paper is about **service quality**
- Investigating whether SDN can add some value
- Focus less on Technology
- and more on architecture and interface
Motivation: Home Network

- Complexity of Home networks

- Growing number of household devices
  - Computers, tablets, smart-phones, TVs, IoT, ...
  - Increased peak-load and congestion on access link

- Yes indeed users want better quality!
  - Providing end-users some means to control service quality
  - Employing SDN to ISP expose programmatic interface for the users
Our proposal: SDN-driven virtualization

- Service quality control exposed via “APIs”
- Create dynamic on-demand “slices” in the network
  - Bandwidth Slicing
  - Separate queue per device/application
- Selective (per flow) control over quality
- APIs open for (any) end-user
Use-cases

- QoE for *streaming video* (e.g. YouTube, NetFlix):
- Elastic *bulk transfer* (e.g. Software upgrades, P2P)
- Low-volume *web-browsing*
Interface: GUI

- Weighted mean to calculate bandwidth
- John’s bandwidth = $9 / (6+9+7+3)$
The APIs

- **Caller ID**: Id of the entity requesting the service
- **Call Type**: Service being requested
- **Flow tuple**: IP src, dst addresses and ports
- **Bandwidth**: Bandwidth requested by flows (in Kbps)

- **JSON API**: 
  ```json
  {hello: 'juke-box', type: 'minbw', numQueue: 2, qid: 1, 
   nwsrc: '203.5.76.0/24', nwdst: '149.171.37.162/32', 
   bw: 5500, qid: 2, nwsrc: '0.0.0.0/24', 
   nwdst: '149.171.37.115/32', bw: 4000}
  ```

- **For Parental control**: 
  ```json
  {\ldots\ldots\ldots Actions:drop}
  ```
Test-bed @ UNSW

- OpenvSwitch over Dell server
  - Flow queue per API call, HTB slicing
- Floodlight (Java) controller
  - Restful API
- AP: TP-LINK
- Clients: PowerShell scripted
  - C1: HD video or browsing; C3: large download
- Video: YouTube / Web-page: facebook
Experimental Results

Best effort performance

- C1: Browsing, Video
- C2: Download
Device level slicing

- C1: Browsing, Video  
  C2: Download
- Perfect video MOS of 3.25
- Page load times unaffected

![Graphs showing device level slicing results](image)

(a) Web browsing  
(b) Video stream
Application level slicing

- C1: IDM + Browsing ; IDM + Video
- Individual applications on C1 using independent minimum BW queues
Iperf measurement

- Effect of TCP, UDP cross traffic
Conclusions

- Access network remains a bottleneck
- Ultimate-goal:
  - make network dynamic so it can be exposed programmatically to outside entities
- Future work:
  - Richer and more programmable home network
    - Security feature
    - More application aware