Personalizing the Home Network Experience using Cloud-Based SDN

Hassan Habibi Gharakheili (UNSW)

Jacob Bass (Newtworks)

Luke Exton (UNSW)

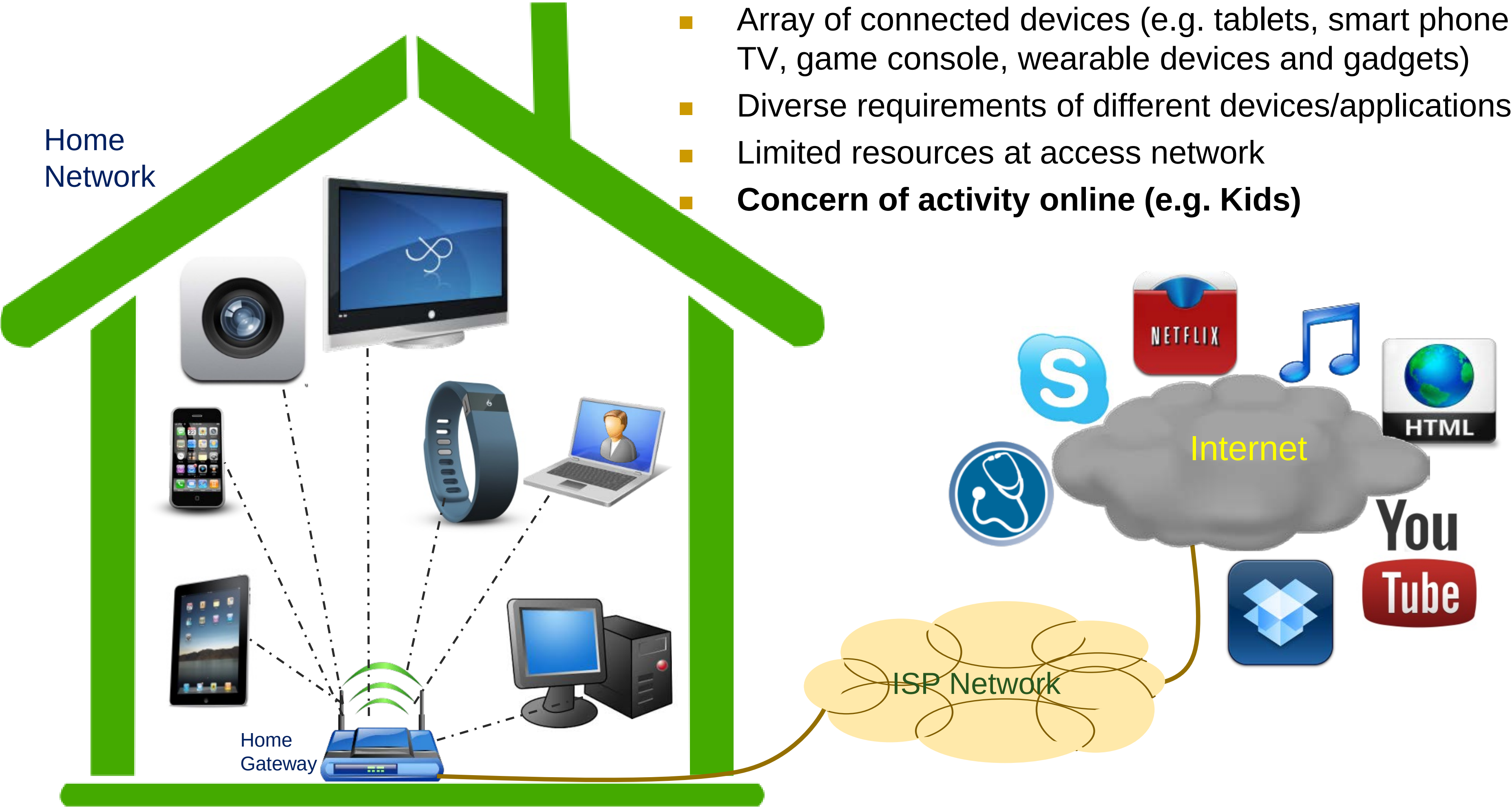
Vijay Sivaraman (UNSW)

Overview

- This paper is about service management
  - Empowering home network user to;
    - Self-customize the network experience
    - Service personalization
    - e.g. *father’s laptop prioritized over kid’s iPad*
  - Using SDN-enabled architecture
    - Abstract the network, Simplify and
    - Exposed via automated interface

Motivation: Home Network

- **Complexity**
- Array of connected devices (e.g. tablets, smart phone, TV, game console, wearable devices and gadgets)
- Diverse requirements of different devices/applications
- Limited resources at access network
- **Concern of activity online (e.g. Kids)**



Challenges

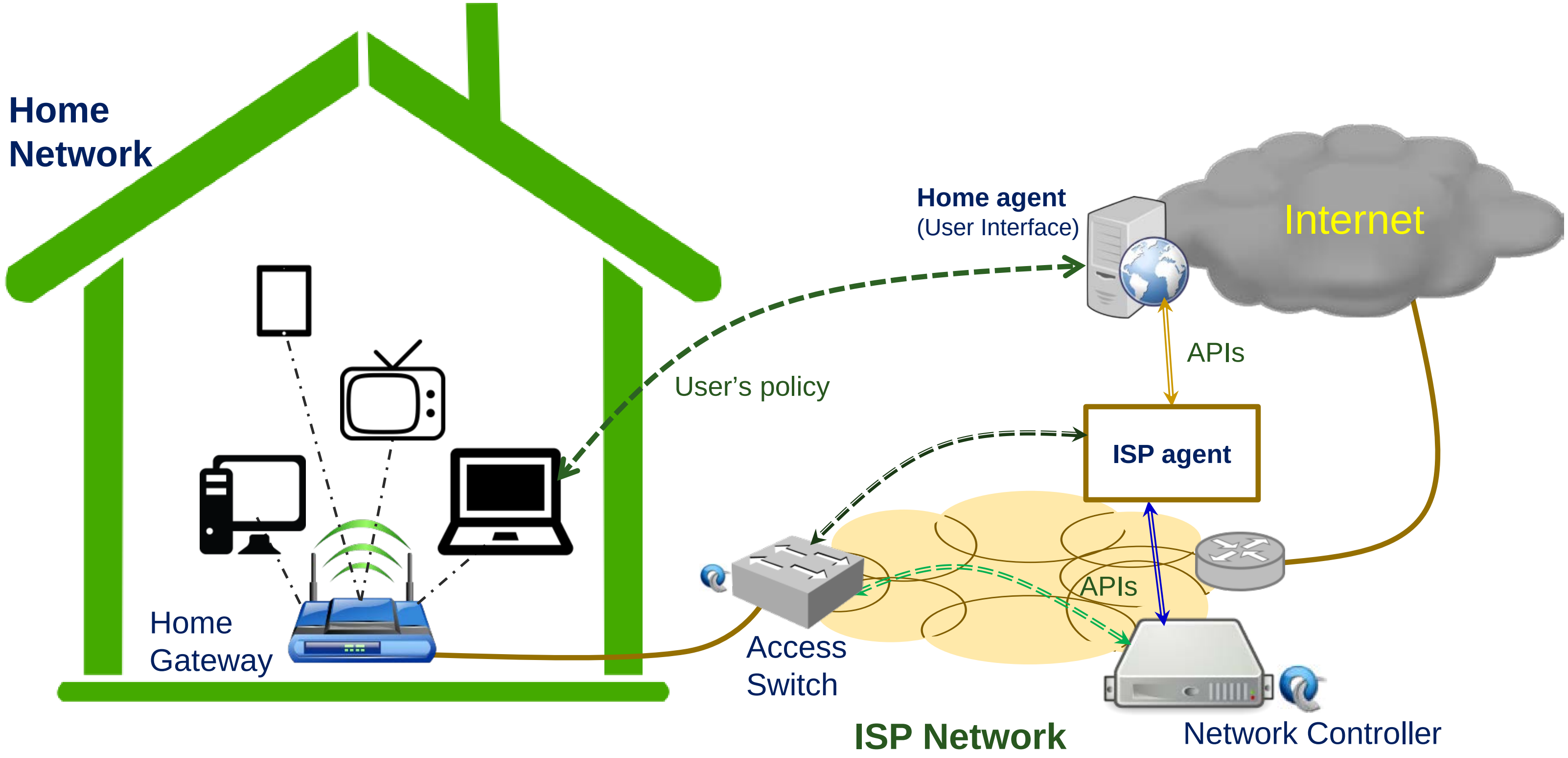
- Indeed users want control!
  - But typically are unskilled
  - “Automated self-provisioning” is a key point
- Some featured home gateways allow customization
  - Requires user sophistication
  - Static and non-uniform solutions
  - Not address the bottleneck link coming into home
- ISP is best positioned, but:
  - Managed services require manual configuration
  - Traffic discrimination may raise “net neutrality” issue
  - Invisible into home network (NAT)

Use-cases

- Enhanced QoE for:
  - Streaming video (e.g. YouTube)
    - large share of downstream Internet traffic
    - Suffering of variable bandwidth available
      - Start-up delay and rebuffering
  - Video conferencing (e.g. Skype)
    - Becomes popular means of communication
    - Interactive communication; more sensitive
      - Loss, latency, and bitrate
- Parental control for:
  - Web content (e.g. social networking)
  - Need of dynamic and customized
    - Based on age of kids, values, priorities
    - E.g. restriction of web access while studying or social networking for elementary years

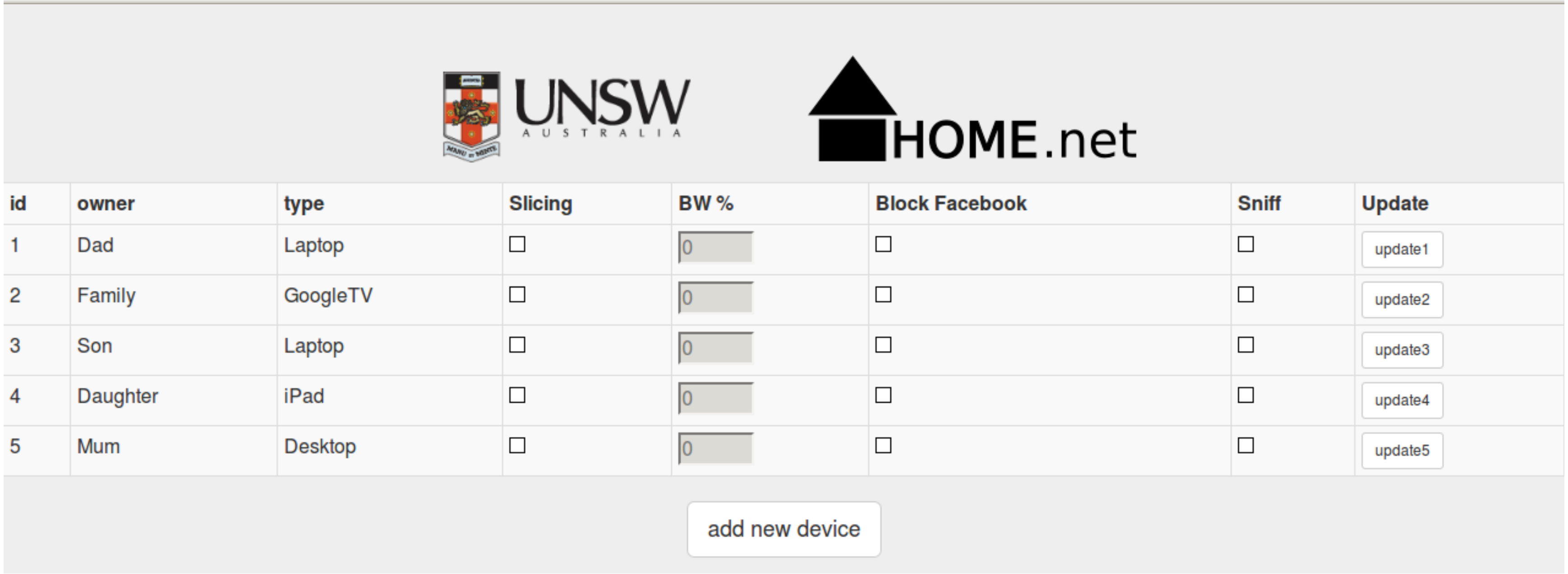
System Architecture

- Front-end user agent
  - Hosted on the cloud
- Back-end SDN (switch, controller, ISP agent and APIs)



Operational Scenario

- Device specific, high-level demand (policy) is taken via UI
- Translated into low-level network semantics
- Communicated to network controller
- Applied into the switch



id	owner	type	Slicing	BW %	Block Facebook	Sniff	Update
1	Dad	Laptop	□	0	□	□	update1
2	Family	GoogleTV	□	0	□	□	update2
3	Son	Laptop	□	0	□	□	update3
4	Daughter	iPad	□	0	□	□	update4
5	Mum	Desktop	□	0	□	□	update5

APIs

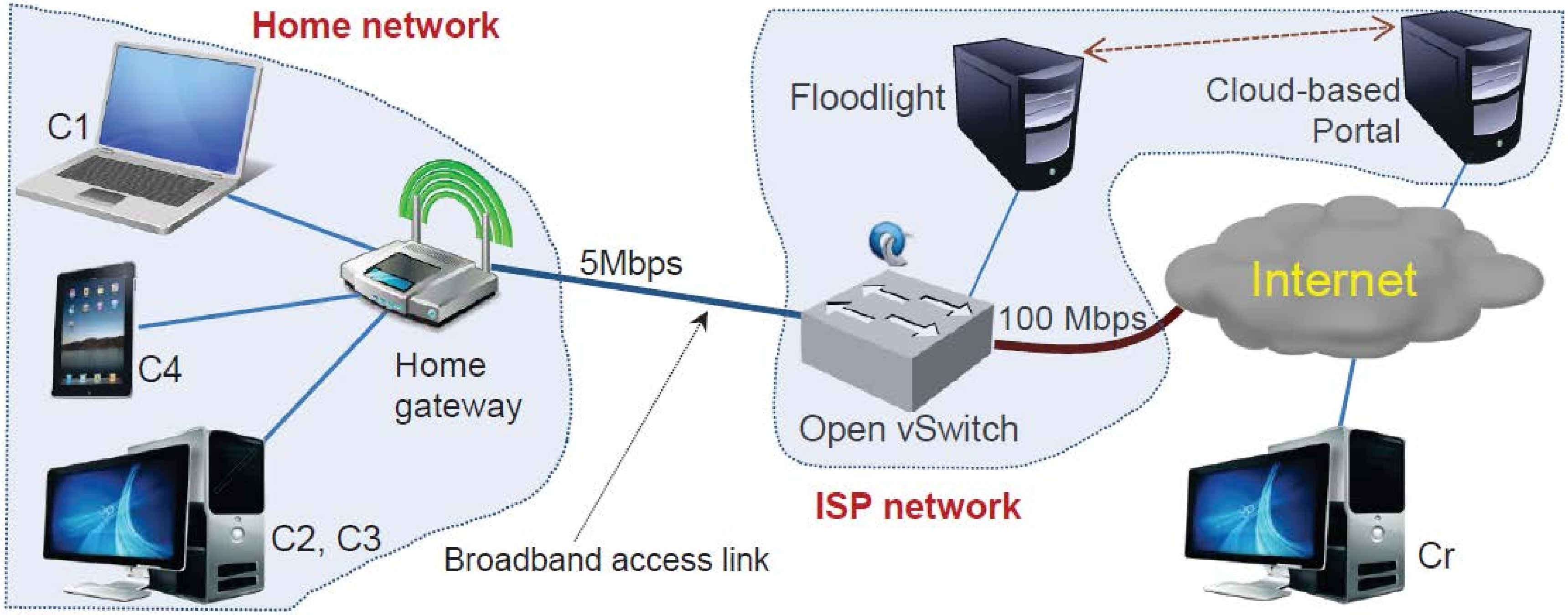
- Bandwidth assurance
  - `“Policy”: “minBW”`
  - `“Device_ID”: “MAC”`
  - `“rate”: X` (minimum rate of queue)
  - Creates/updates a queue on the switch and pushes the flow into the respective queue

- Parental Control
  - `“Policy”: “PC”`
  - `“Device_ID”: “MAC”`
  - `“black-list”: {IPadd1, IPPadd2,.. }`
  - creates static flow that drop all traffic originating from the pre-defined range of IP block destined to the related device

Prototype

- **Home Agent**: the web portal runs as a standard HTML web-site, and is served by an off-site web-server run by ISP

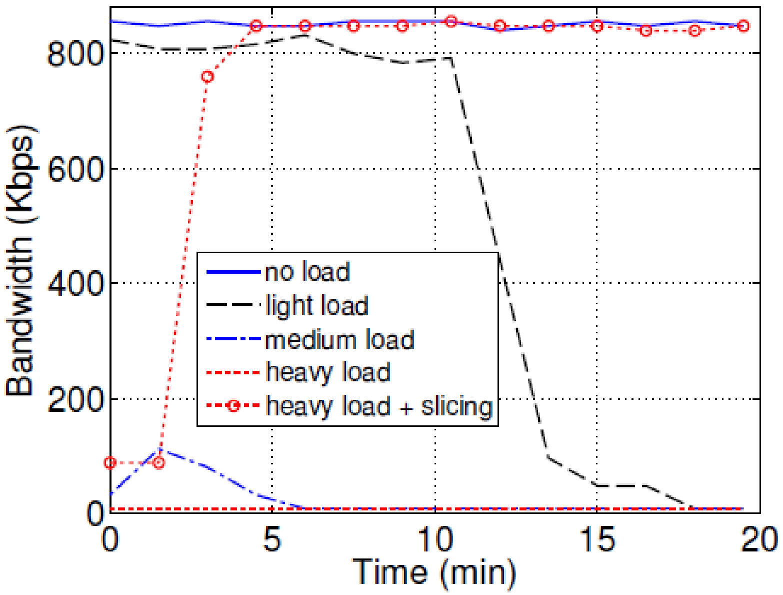
- **ISP Agent**: runs as a java program on an internal ISP server, communicating over HTTP with the Home agent

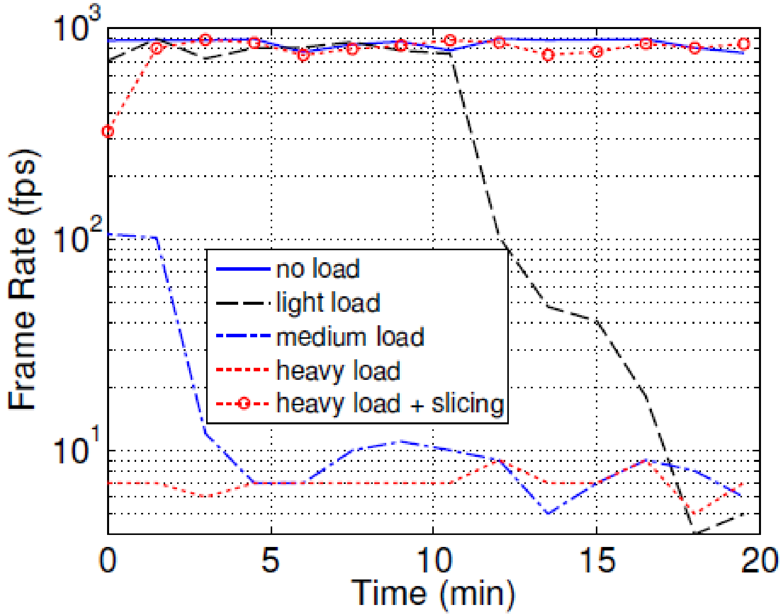


Experimental Evaluation

- Test suite:
  - Skype video call
  - Pytomo tool
    - open-source YouTube crawler and analyzer
  - MOS measurement for an HD video
- Network condition:
  - No b/g load
  - Light b/g load
  - Medium b/g load
  - Heavy b/g load (aggressive IDM)
  - Heavy b/g load with service assurance

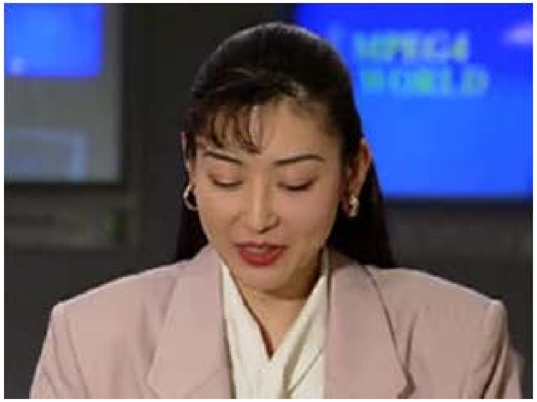
Skype: technical metrics



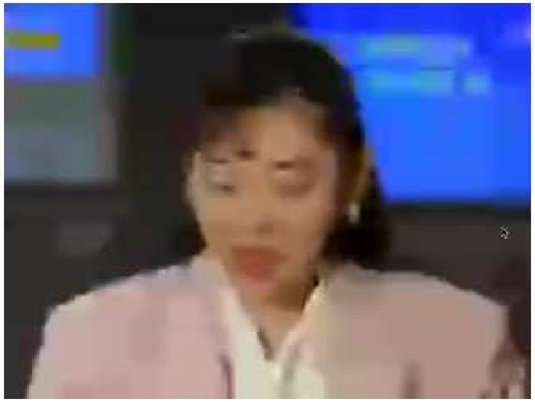


No load	Light load	Medium load	Heavy load	Heavy load + Slicing
3 ms	86 ms	686 ms	1331 ms	2 ms

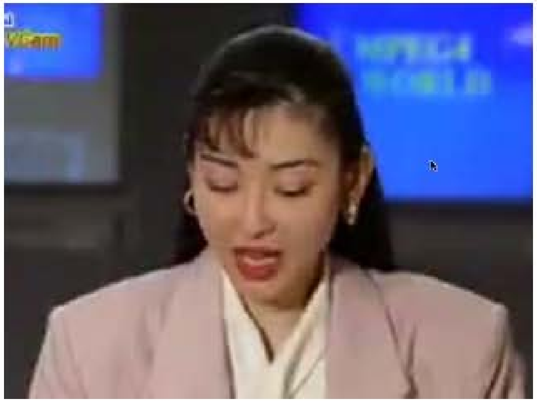
Skype: visual perception



No b/g load

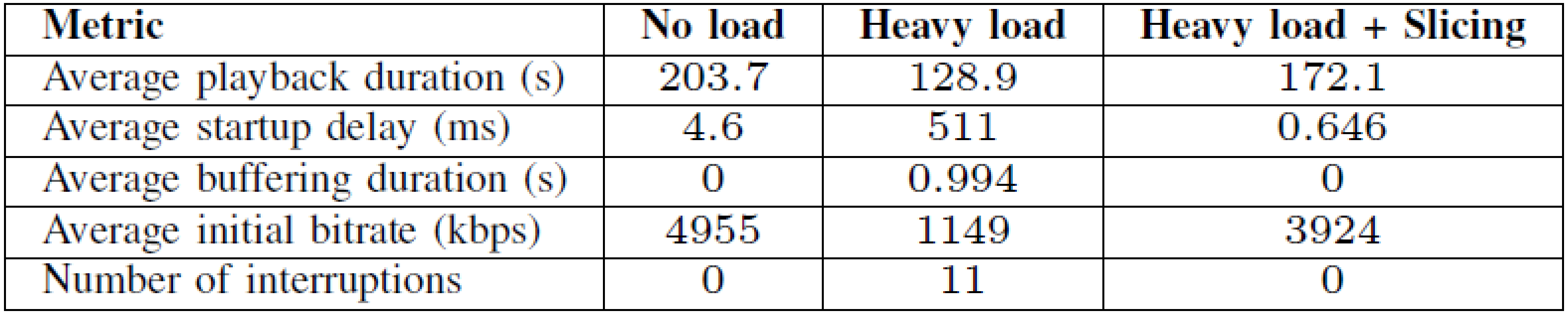


Heavy b/g load



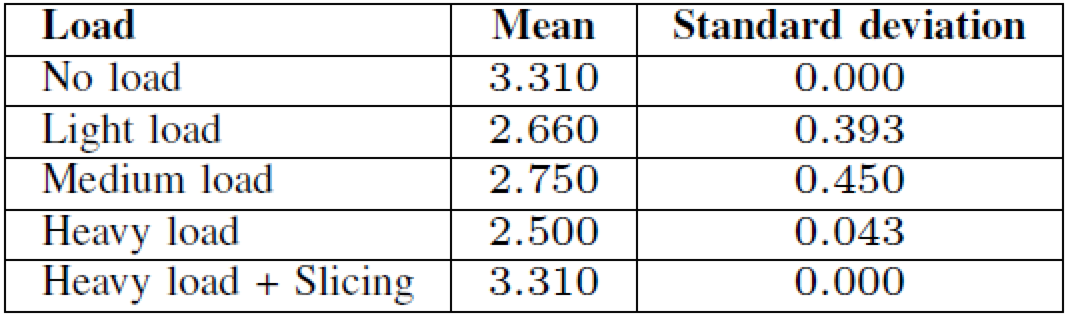
Heavy b/g load with service assurance

Pytomo



Metric	No load	Heavy load	Heavy load + Slicing
Average playback duration (s)	203.7	128.9	172.1
Average startup delay (ms)	4.6	511	0.646
Average buffering duration (s)	0	0.994	0
Average initial bitrate (kbps)	4955	1149	3924
Number of interruptions	0	11	0

YouTube MOS



Load	Mean	Standard deviation
No load	3.310	0.000
Light load	2.660	0.393
Medium load	2.750	0.450
Heavy load	2.500	0.043
Heavy load + Slicing	3.310	0.000

- The MOS value gradually drops while more loads are introduced to the network

- The last row shows how the user can benefit from managed service quality realizing similar performance as no load

- Indeed, this QoE improvement comes at the cost of slowing down the other unimportant downloads

Parental Control:

- FaceBlock!
  - Used the publicly available block of IP addresses provided by Facebook to populate a blacklist
  - Enable/disable static flows instructing the switch to drop the associated flows
  - Once enabled, the blocking takes place immediately

Conclusions and Future Directions

- Access network remains a bottleneck
- User benefits from self-customisation and enhanced QoE
- ISP benefits from user satisfaction and monetization opportunity
- End-goal: make network dynamic so it can be exposed programmatically to outside entities
- Future Work:
  - Offering more features
  - Dynamic negotiation via Apps interface
  - Over legacy networks