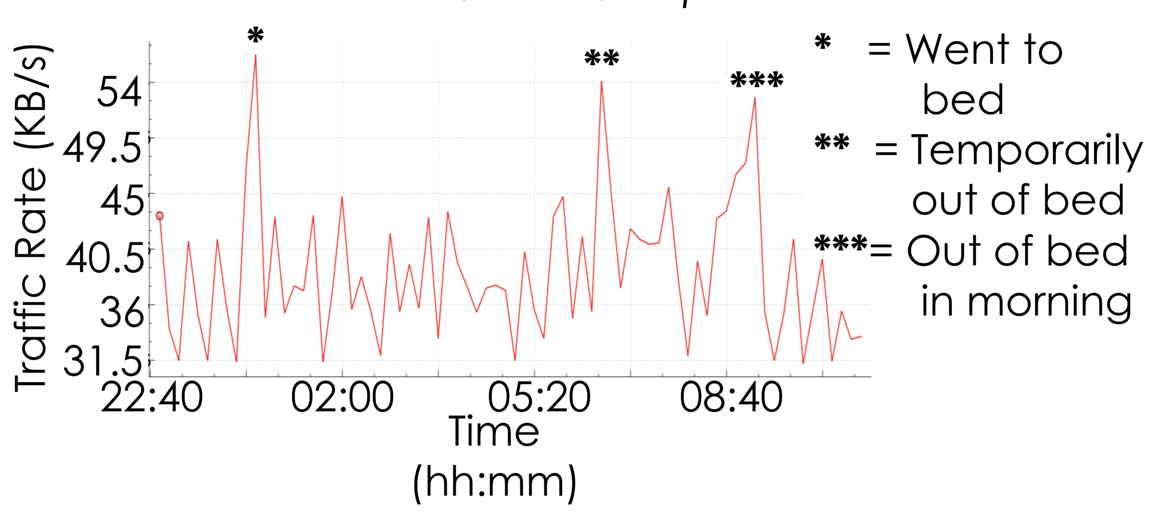
Privacy-Preserving Traffic Obfuscation for Smart Home IoT Devices

Trisha Datta (tdatta@princeton.edu)

Motivation and Goal

- With proliferation of IoT devices, adversaries are capable of inferring user events just by looking for peaks in traffic flows

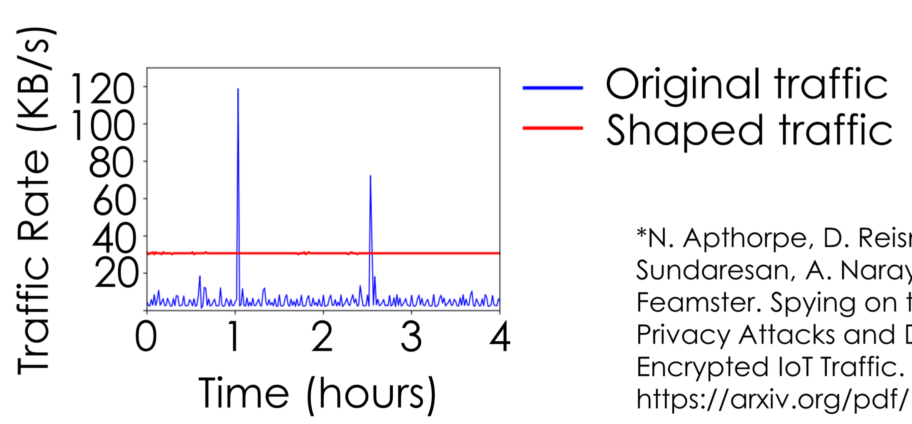
Traffic from a Sense Sleep Monitor



- **GOAL: Protect user privacy by creating library for IoT developers to obfuscate traffic**

Related Work

- Constant rate traffic shaping solution on router using Raspberry Pi WiFi access point and two priority queues*



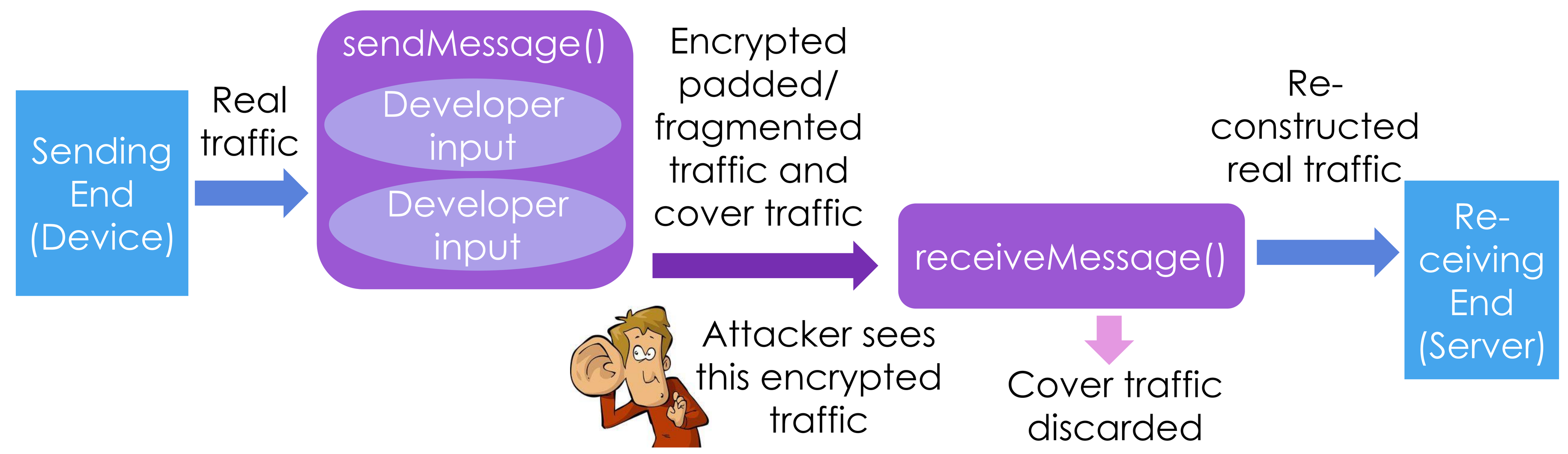
*N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster. Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic. https://arxiv.org/pdf/1708.05044.pdf.

- Traffic shaping studied in anonymity context
- Multiple ML algorithms used to simulate realistic IP traffic
- Limitations of state of the art
  - Limited scalability
  - Limited flexibility
  - Does not protect against sniffing inside the home

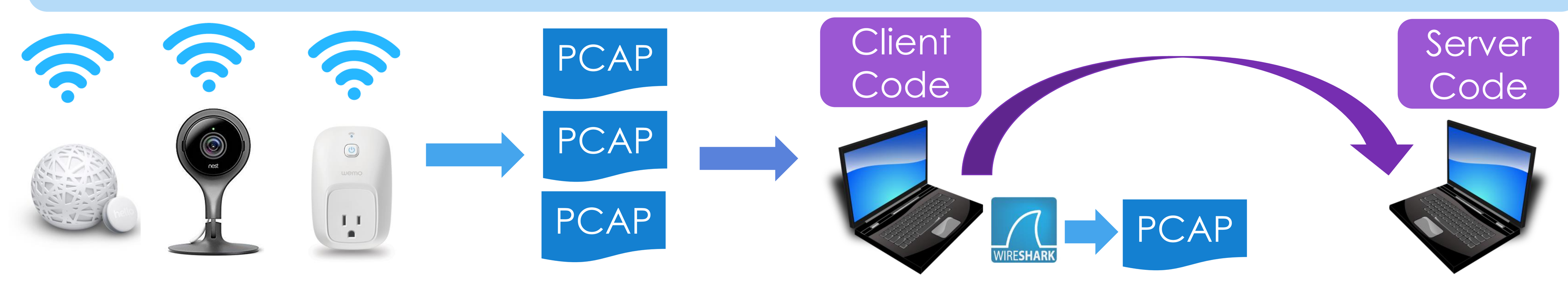
Approach

- Narrow scope to non-general-purpose devices
- Create solutions for three categories of devices
  - High-latency devices – functionality not affected by long delays
  - Low-bandwidth low-latency devices – low bandwidth consumption during user events, functionality affected by long delays
  - High-bandwidth low-latency devices – high bandwidth consumption during user events, functionality affected by long delays
- Traffic shaping and injection of cover (fake) traffic are main conceptual tools
- Develop new send()/receive() functions to facilitate easy integration into code
- Develop new recovery protocol for constructing/re-constructing messages at the sending and receiving ends

Library Implementation



IoT Device Simulation



Definition of Privacy: Based on Differential Privacy

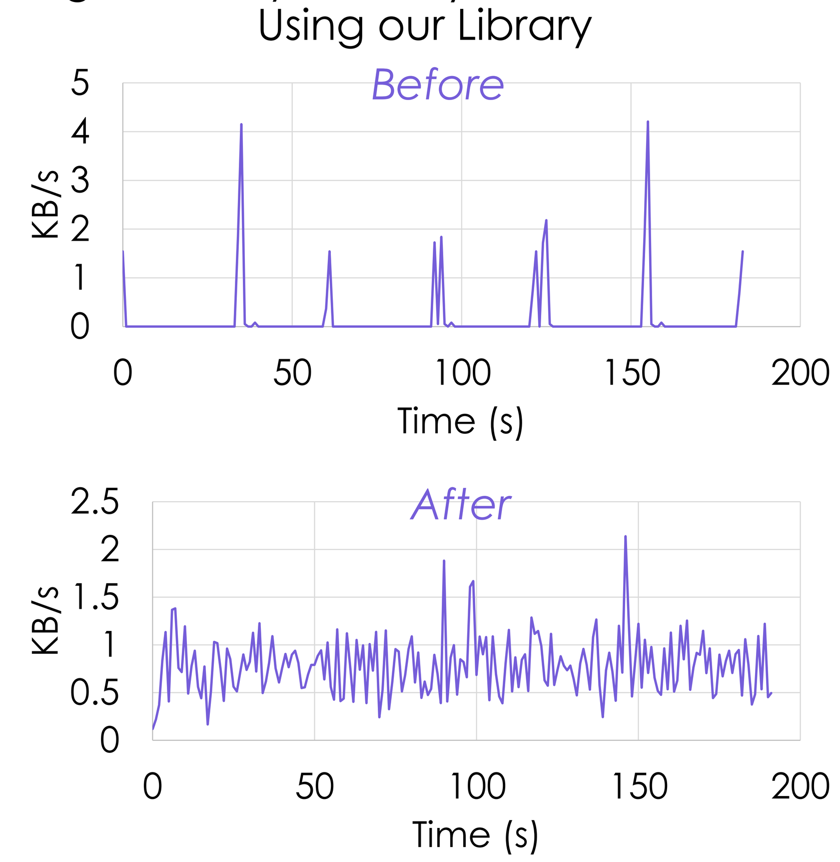
- Differential privacy
  - Databases D and D' that differ in one row
  - Function Q applied to D and D'
  - Query results on Q(D) and Q(D') should be statistically indistinguishable
- Analog to our problem
  - Database rows ⇔ User event peaks
  - Function Q ⇔ Traffic shaping/cover traffic injection solution
  - Query results ⇔ Traffic flows
- Secure if traffic flow with given event is statistically indistinguishable from traffic flow without event

Customized Implementations for Different Device Types

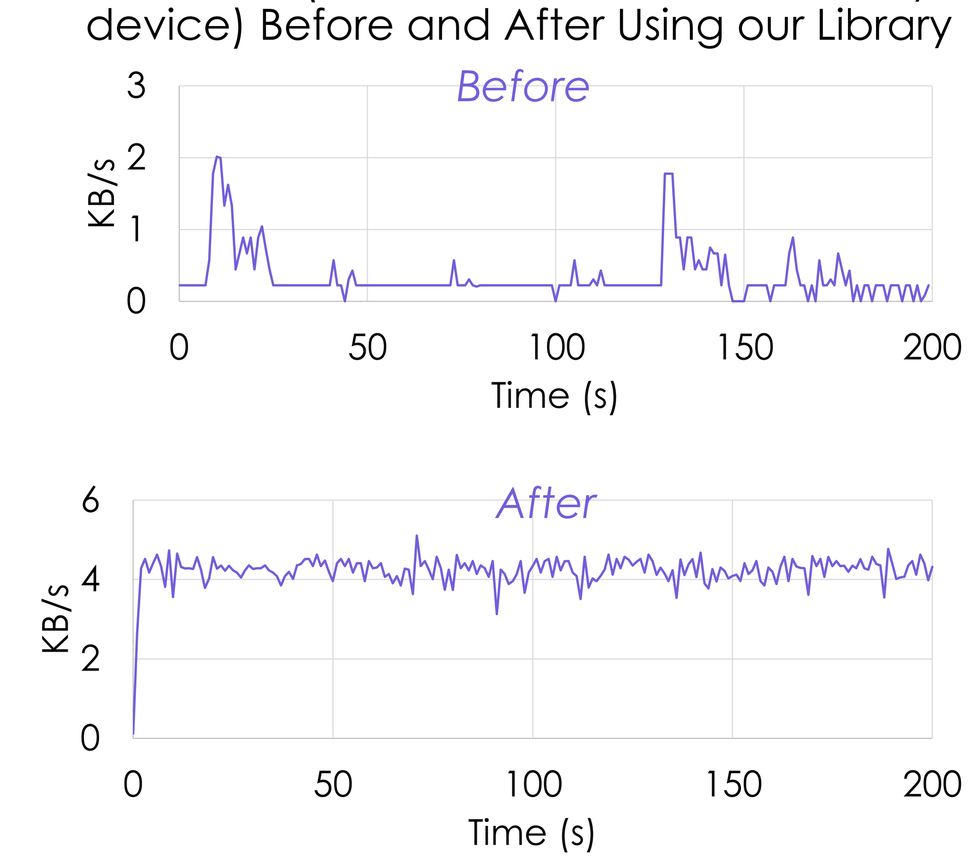
- High-latency devices: enforce specific distributions of payload sizes and interpacket delays
- Low-bandwidth low-latency devices: enforce constant traffic rate
- High-bandwidth low-latency devices: simulate realistic user events with Hidden Markov Models so that real and simulated events become indistinguishable

Evaluation of Privacy Preservation

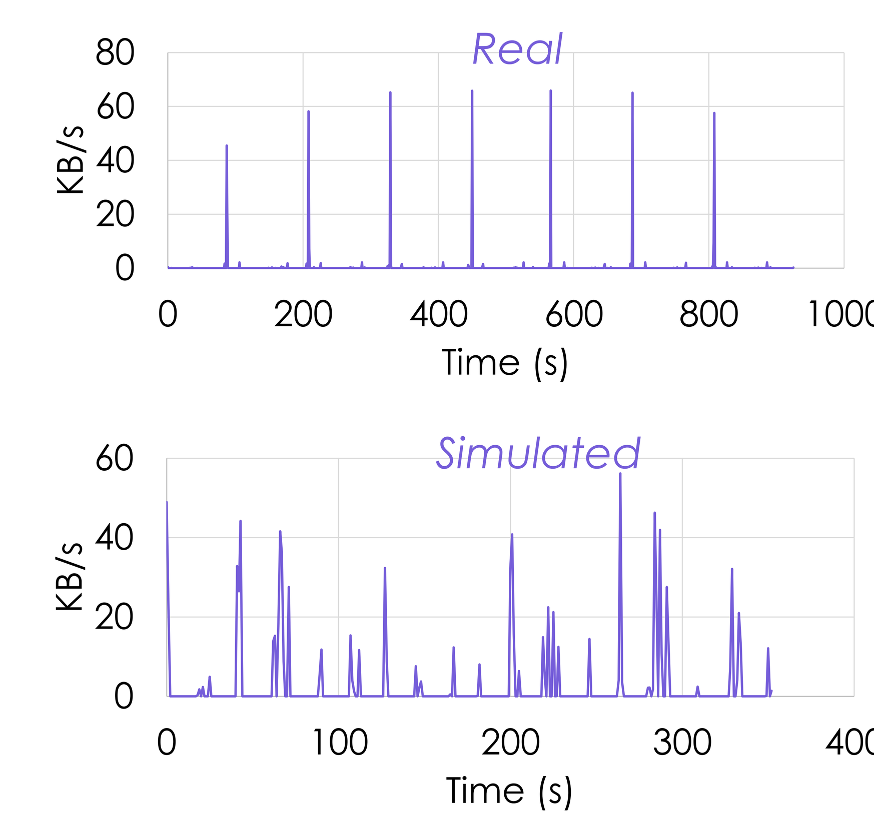
Traffic from a Sense Sleep Monitor (a high-latency device) Before and After Using our Library



Traffic from a Nestcam Security Camera Motion Detector (a low-bandwidth low-latency device) Before and After Using our Library



Real Traffic from a Belkin WeMo Switch (a high-bandwidth low-latency device), Traffic Simulated Using Hidden Markov Models, and Tables Showing Results of T-Tests and KS-Tests



Results of T-Tests

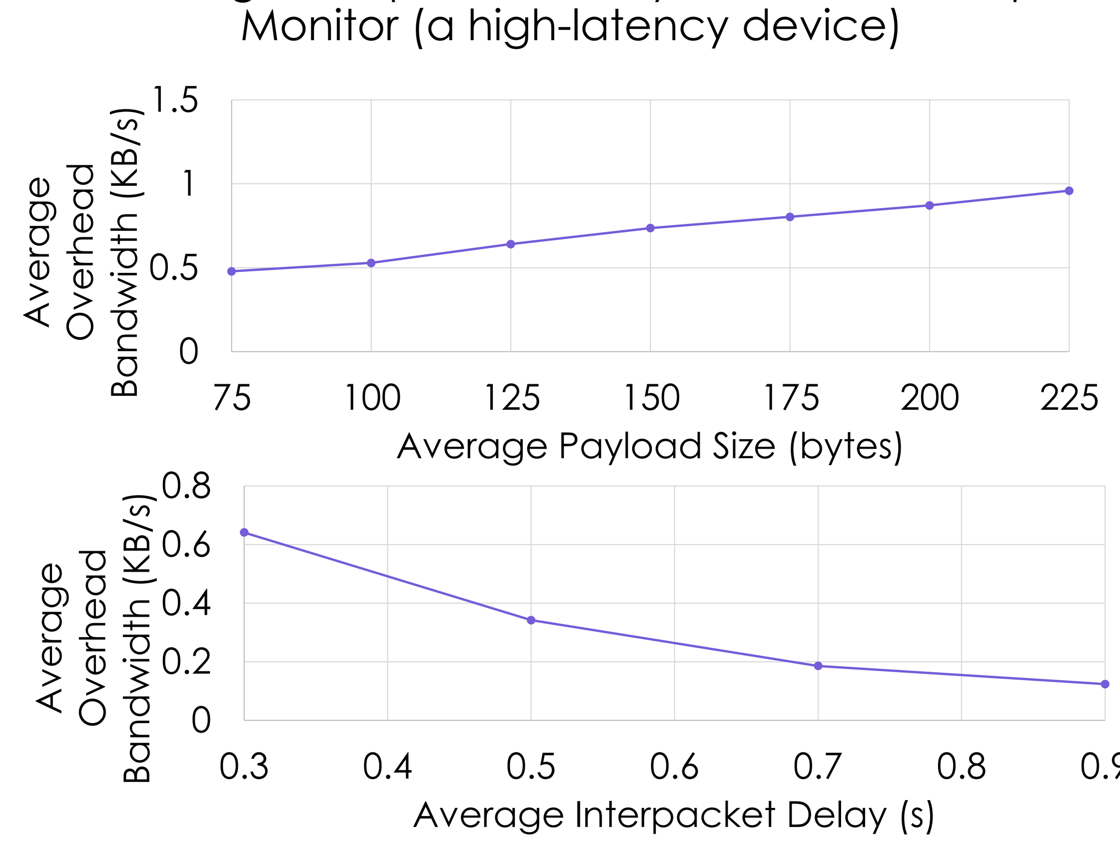
Distribution	t-statistic	p-value
Payload Sizes	-0.304	0.761
Interpacket Delays	0.862	0.389

Results of KS-Tests

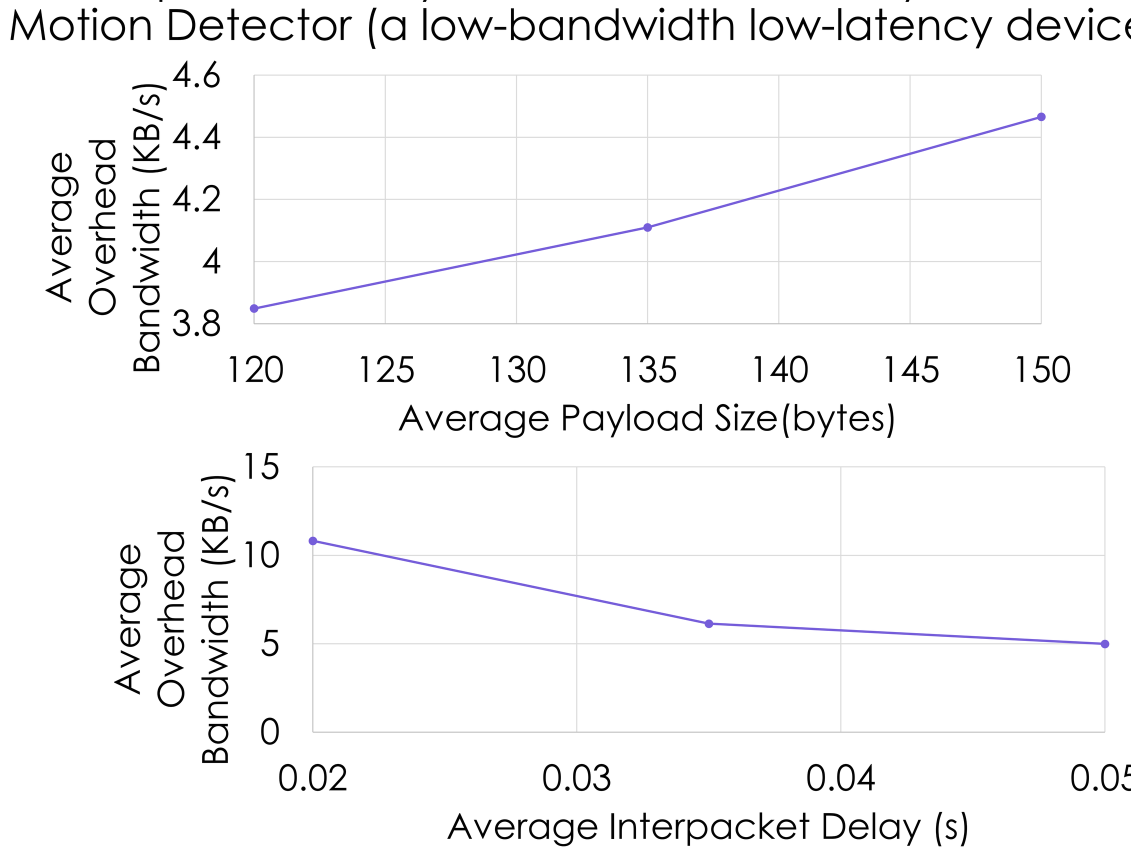
Distribution	ks-statistic	p-value
Payload Sizes	0.040	0.342
Interpacket Delays	0.040	0.334

Evaluation of Overhead Bandwidth Consumption

Effect of Changing Average Payload Size and Average Interpacket Delay for a Sense Sleep Monitor (a high-latency device)



Effect of Changing Average Payload Size and Average Interpacket Delay for a Nestcam Security Camera Motion Detector (a low-bandwidth low-latency device)



Bandwidth Consumption for Belkin WeMo Switch (a high-bandwidth low-latency device)

- Measured an average of 2.56 KB/s of average overhead bandwidth with same model

Comcast and Cox Communication have 1 TB/month caps

- Device that uses 4 KB/s of overhead bandwidth will use 1% of this cap