Two Real-World Deployments

- **Usage control in home networks**
  - Implementation of user controls (e.g., usage cap management, parental controls) in home networks
  - **Today:** Not possible
  - **With SDN:** Intuitive, simple

- **Access control in enterprise networks**
  - Re-implementation of access control on the Georgia Tech campus network
  - **Today:** Complicated, low-level
  - **With SDN:** Simpler, more flexible
Deployment Status

• Over 300 routers deployed in home networks “in the wild”

• Collaboration with Measurement Lab on monitoring network performance from various regions and ISPs.

• Ongoing trials with several ISPs as part of private deployments

• Firmware
  – OpenWrt, with luci web interface
  – IPv6-capable

• Netgear 3800 router
  – Atheros chipset
  – MIPS processor, 16 MB flash, 64 MB RAM
  – Gigabit ethernet
  – 2.4 GHz and 5 GHz radio
Ongoing Extensions

• **More measurements**: Denser deployments (e.g., apartments)

• **Broader scope**: More measurements (e.g., integration with Tor’s OONI project)

• **Sensor fusion**: Tighter integration with other in-home, *in situ* sensing capabilities (e.g., phones)

• **Open programming interface**: Enable other researchers to perform measurements
The Need for a Policy Language

• Network policies
  – Are dynamic
  – Depend on temporal conditions defined in terms of external events

• Need a way to configure these policies without resorting to general-purpose programming of a network controller

• Intuitive user interfaces can ultimately be built on top of this language
The Need for Reactive Control

• Simple policies are doable in FML: “Ban the device if usage exceeds 10 GB in the last 5 days”

\[
\begin{align*}
\text{deny}(& 
\text{Us, Hs, As, Ut, Ht, At, Prot, Req} \leftarrow \text{over}(
\text{Hs})
\text{over}(
\text{Hs}) \leftarrow \text{usage}(
\text{Hs}, \text{lastDays}(5), \text{amt}), \text{amt} > 10.
\end{align*}
\]

• But, adding **temporal predicates** is difficult!
  – “Remove the ban if usage drops below 10 GB.”
  – “Remove the ban when an administrator resets.”

• Each condition requires a new predicate.

\[
\text{over}(
\text{Hs}) \leftarrow \text{usageOnceExceeded}(
\text{Hs}, \text{lastDays}(5), 10).
\]
Language Design Goals

• **Declarative Reactivity:** Describing when events happen, what changes they trigger, and how permissions change over time.

• **Expressive and Compositional Operators:** Building reactive permissions out of smaller reactive components.

• **Well-defined Semantics:** Simple semantics, simplifying policy specification.

• **Error Checking & Conflict Resolution:** Leveraging well-defined, mathematical semantics.
Procera: Programming Reactive, Event-Based Network Control

- **Controller:** signal functions and a flow constraint function
- Receives **input signals** from environment
- Periodically updates a **flow constraint function** that controls the forwarding elements

Define a signal function for a device going over (or under) the usage cap:

```
overUnderEvent =
  proc env → do
  capMap ← capTracker ← env
  usageDb ← usageTracker ← env
  usageChanges ← usageChangesTracker ← env
  let now = calendarTime env
  let over src =
    monthlyUsage usageDb now > capMap ! src
  condSplit over ← usageChanges
```

Define the set of devices over the cap:

```
overSetStream =
  proc env → do
  (over, under) ← overUnderEvent ← env
  toSetStream ← (over, under)
```
Next Steps: Faster, Programmable Data Plane

• Augment OpenFlow switches with custom packet processors
• **Device abstraction layer** to allow programmability of this substrate
  – Single device
  – Network wide
• **Applications**
  – Big data applications
  – On-the-fly encryption, transcoding, classification
  – Selective deep packet inspection
Summary

• Software Defined Networking can simplify network monitoring and management, but we still need new control models.

• **BISmark**: Better visibility and control of home networks

• **Lithium**: Event-based network control

• **Procera**: Policy language for SDNs

• Next
  – A fast, programmable data plane for SDN