Elastic Switch
Managing network services at the edge

Joint work with collaborators at
U. Washington, U. Pisa, CMU and EPFL
• SDN has its roots in the network control plane
  – Traditionally complex, closed, hard to evolve

• This talk: a data-plane centric viewpoint
Early Internet architecture tells a story of a simple and minimal data plane.

The story is flawed:
- minimalism is not what we want
- simplicity is not what we have

Reality: The network data plane is a mess.
The Traditional IP Data Plane

• IP as the narrow waist
  – endpoints create packets with a destination address
  – routers deliver packets to their destinations
The Traditional IP Data Plane

• IP as the narrow waist
• A simple model with many desirable properties
  – no (data plane) state at routers
  – each packet is processed independently
  – end systems don’t see the gory details of federation
The Traditional IP Data Plane

- IP as the narrow waist
- A simple model with many desirable properties
- But increasingly, “simple” just seems naïve
  - want to attribute/diagnose problems but IP hides details of federation
  - want to control what traffic is allowed but IP offers universal reachability with no authentication
  - want to optimize for different apps and customers but IP offers no hooks for differentiation
The Traditional IP Data Plane

- IP as the narrow waist
- A simple model with many desirable properties
- But increasingly, “simple” just seems naïve
- And minimalism hinders competition
  - innovation offers no competitive advantage if we all provide exactly the same service

Enter “middleboxes” and the modern Internet data plane
The Data Plane Today

• A proliferation of one-off “middleboxes”
  – standalone and specialized devices
  – largely transparent to protocols, routers, apps
The Data Plane Today

• A proliferation of one-off “middleboxes”

Case-study at a large enterprise: ~636 middleboxes, 900 switches/routers
The Data Plane Today

• A proliferation of one-off “middleboxes”

Many kinds of devices, vendors, management tools
Recap

• Data planes in the past:
  – minimal processing
  – simple
Recap

• Data planes in the past: today:
  – minimal rich processing
  – simple complex
    • closed
    • inefficient
    • custom management APIs
    • limited/no extensibility

• An increasingly well-recognized problem...
Many ideas on how to do better

• Middleboxes as a cloud service [APLOMB, Sigcomm’12]
  – case study: outsource 90% with avg 1.1ms penalty

• AT&T Domain 2.0 [Anschutz 2013]
  – Central Office as the cloud platform

• Middlebox APIs for cloud tenants [CloudNaaS, SOCC’11]
Common Theme

“Network Function Virtualization” (NFV)
The emerging NFV vision

So, are we done?
Two (potential) problems

(1) We’re replacing monolithic HW with monolithic SW
   – each app still comes with custom management
   – apps still far from agile/extensible/composable
   – each developer reinvents the wheel on common fns.
Contrast/Inspiration

• Modern data analytics systems
  – a programming model (map-reduce)
  – but also a runtime *framework*

• **Framework** takes care of
  – scheduling
  – high availability
  – policy
  – ...

• Automates management for operators
• Frees app developers to focus on app logic
Two (potential) problems

(1) We’re replacing monolithic HW with monolithic SW
Two (potential) problems

(1) We’re replacing monolithic HW with monolithic SW

Need a framework for NFV applications!

– decouple app logic from techniques for <HA, scaling,...>
– general techniques for <HA, ...>
– may need: programming model / guidelines
Framework Wishlist

• General techniques for:
  – Fault tolerance
  – Scheduling
  – Dynamic scaling / load-balancing
  – Performance optimization
  – Performance QoS / isolation
  – Network configuration
  – Service composition
  – Programming abstractions*
Two (potential) problems

(1) We’re replacing monolithic HW with monolithic SW

(2) We need to blur the boundary between SDN & NFV

“Elastic Switch” aims to provide this framework, with joint application-network management
Elastic Switch

• Platform for NFV services
  – edge device in SDNv2
  – rack in CO (Domain 2.0) or EC2 (APLOMB)*

• Built on commodity CPUs + switches

• Provides framework for NFV apps
  – general solutions for scheduling, scaling, etc.
  – programming abstractions to aid above
  – tools for code verification, model checking, etc.
Elastic Switch

- "SoftNIC"
  - runs on 1+ dedicated core(s)
  - packet or byte-stream abstraction
Elastic Switch

Unified (net+app) controller

OpenFlow
Elastic Switch

Unified (net+app) controller

Richer network abstractions

Loose integration
Ongoing projects (1)

Foundation: fast, flexible, predictable SW packet processing

• Fast and modular software traffic processing
  – SoftNIC: a fast yet flexible SW switch (UCB, KAIST)

• Predictable performance in SW middleboxes
  – Dobrescu et al. NSDI 2012 (EPFL, UCB)
  – Leveraging hardware support (UCB, U. Toronto, Intel)

• Formal verification of software dataplanes (EPFL)
  – Dobrescu et al. NSDI 2014 proved code-level verifiability
Ongoing projects (2)

- Joint network-application scheduler/optimizer
- Fault-tolerance SW middleboxes
  - Justine’s talk to follow
- FlowTags: Abstractions for service composition (CMU)
  - Fayaz et al. NSDI’14
Ongoing projects (3)

- **APLOMB**: Traffic processing as a cloud service
  - Sherry et al., SIGCOMM 2012 (UCB, UW)

- Open Interfaces for Carrier Services
Thank you.