Quick Overview of SDN/NFV Research at Berkeley

A Combination of Perspectives

- PIs: Sylvia Ratnasamy and Scott Shenker
- Ratnasamy: ~9 years at Intel, leading Routebricks
  - Resurgence of commodity-based packet processing
  - Enabled the rise of NFV
- Shenker: Heavily influenced by years at Nicira
  - SDN, with focus on modularity and insertion
- Resulting synergy: synthesize SDN and NFV
  - SNDv2 is unified approach that embraces both

SDNv2 Leverages "Old" Ideas…

- New functionality deployed via "middleboxes"
  - On-path packet processing key to many new services
- Commodity-based packet processing
  - Already widely used in middleboxes
- The edge-based "fabric" approach
  - Used by Nicira (later VMware) and others
  - And by MPLS long before that….

…To Tackle New Challenges

- Carrier networks (rather than just datacenters)
  - Numerical scale, geographic scope, diverse requirements
- Ability to support third-party (tenant) services
  - SDN focuses on operator interfaces
  - Should allow others to build services on carrier networks
- Incorporate edge-systems that go beyond simple forwarding or even general packet processing
  - Self-managing edge systems that handle job placement, load balancing, fault tolerance, and scale up/down

Basic SDNv2 Design

- Edge/Core split
  - Core: handles packet delivery
    - Including failure recovery and traffic engineering
  - Edge: handles all other functionality
    - Isolation, network virtualization, middleboxes,…
- Edge offers:
  - Hybrid HW/SW packet processing
  - Self-managing clusters (unified processing/network cont'l)
- Open Interfaces for third-party services
  - Service virtualization

Service Virtualization

- Set of low-level per-tenant interfaces
  - E.g., local discovery, name lookup, edge caching, etc
  - Interfaces are safe, self-service, flexible, and useful
- Tenants leverage interfaces to offer global services
  - No need to build their own global infrastructure
- This is “virtualization” because tenant only specifies high-level application interaction, not details
  - Carrier infrastructure ties it all together

Service Virtualization Examples

- Tenant would invoke VPN with the following:

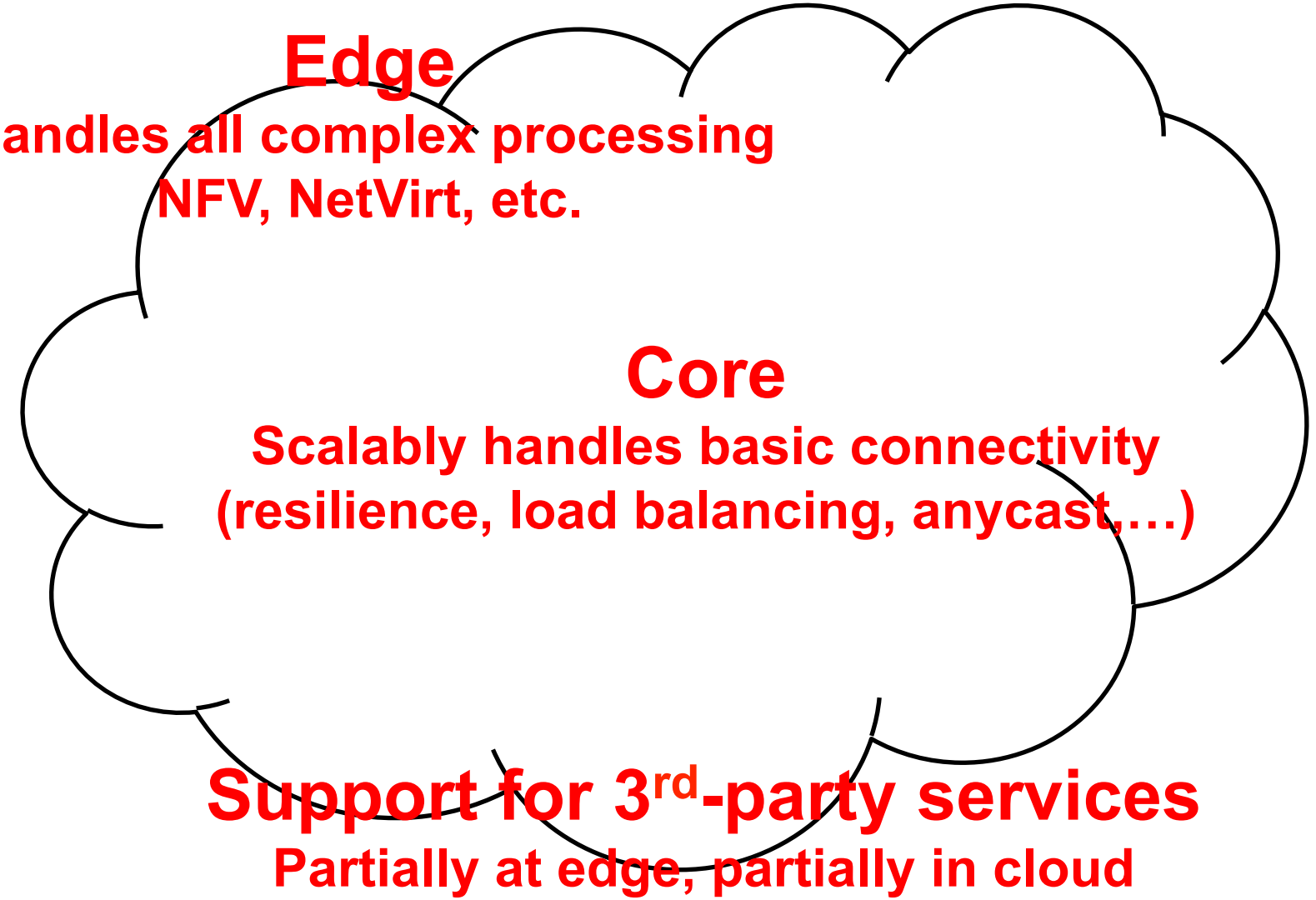
  client ➔ authentication ➔ services

- Client supplies metadata for client/auth/services
  - Carrier uses this metadata to send packets to right place
  - Requires no manual involvement from carrier

- Other examples:
  - CDNs, Dropbox, VoIP, IoT, etc.
  - *Any service that could benefit from edge processing*

- No technical novelty, but enables greater innovation

Architectural Components of SDNv2



SDNv2 Controller

- Hands pipelines to edge
  - Maps packet classes to sequence of processing modules
    - E.g, customer X's traffic gets L2, L3, FW, Proxy…
  - Implementation and management of these pipelines handled by new class of edge device
- Hands address mappings to core
  - Core then handles packet delivery
  - Internal core management could use SDN
- Handles tenant invocations
  - As previously described

Forwarding Interfaces

- Core: MPLS-like
  - But could be legacy networks for now
- Edge: commodity processor, accessed via
  - Downloading code to switch
  - Various new programming models (e.g., SoftFlow)
  - Module invocation (as in NFV)
  - *Can use ASICs, but not limited to ASICs*
- Key point:
  - No need for a single ***universal*** forwarding model, but all forwarding models should be ***open***

SDNv2 Changes Innovation Model

- New functionality deployed in software
  - Hardware can later accelerate
  - But new functionality not gated on ASIC support
- New functionality can be introduced by 3rd parties
  - Not all innovation introduced by operators or vendors
- New functionality does not require changes to core
  - As long as the core supports adequate packet delivery, it need not be changed

Other SDNv2 Advantages

- Simplifies management
  - Controller merely specifies high-level pipelines
  - Core and edge do the rest
- Increases modularity
  - Core and edge have cleanly separated tasks
  - Detailed management of processing offloaded to edge
- Easier to deploy
  - No need to change core for many features
  - Not gated on ASICs or OpenFlow

SDNv2: Heresy or Evolution?

- Some view the diminution of OpenFlow as heresy
- We think this is just a natural design evolution
  - Incorporating new requirements (e.g., NFV, services)
  - And new capabilities (e.g., x86 forwarding)
- Strongly believe an open switch interface is crucial
  - Our point: there should be at least ***two*** forwarding models
  - *Simple and fast* (MPLS-like); *general and not-as-fast* (x86)
  - Just like existing deployments in virtualized datacenters
- This isn't heresy, this is progress…..

Ensuing Talks

- Elastic Switch: self-managing edge systems
  - Framework to support edge services
- Fault-tolerant packet processing
  - An example of what this framework provides
- Recursive SDN
  - Framework to support scalable core packet delivery
  - Not a specific routing/TE solution, but a recursive framework for providing scalability/locality/repair

Areas Not Covered Today

- Efficient and predictable packet processing
- Details of service virtualization
- Verification in networks with middleboxes
- Troubleshooting for distributed SDN software
- Control-plane networks and new consistency models
- Application to wireless (rural, virtualization)
- Middlebox outsourcing

Questions?