Software Defined Networking

Dr. Nick Feamster
Professor

In this course, you will learn about software defined networking and how it is changing the way communications networks are managed, maintained, and secured.

This Lesson: The Road to SDN

- Excitement about SDN has increased over the past few years.
- Yet, many of the ideas have evolved over the past twenty years.
- The term “SDN” was coined in 2009
- Many ideas have roots in earlier technologies (e.g., phone network)

Three Stages

- **Active Networking:** Programmable networks
- **Control and Data Plane Separation:** Open interfaces between control and data planes
- **OpenFlow API and Network OSes:** First instance of widespread adoption of an open interface

Active Networking

- More diverse applications and greater use
  - Researchers wanted to deploy new ideas
  - First attempt to make networks programmable
- **Technology push:** Reduction in computing costs, Funding agency interest
- **Use pulls:** Operator frustration with deployment challenges

Active Networking: Intellectual Contributions

- Programmable functions in the network
- Network virtualization
- Demultiplexing to software programs
- Vision of a unified architecture for middlebox orchestration

Active Networking: Myths

- **Myth:** End-user would program packets
- **Reality:** This programming model would be rare

- **Myth:** Packets must carry Java code
- **Reality:** Active networking had a programmable router/switch model

Control/Data Separation

- **Pragmatism** (narrower scope)
  - Attempt to solve traffic engineering problems
- **Technology push:** Open interfaces between control and data planes (e.g., ForCES), logically centralized control (e.g., RCP)
- **Use pull:** Pressing network management problems

Control/Data Separation: Intellectual Contributions

- Logically centralized control using an open interface to routers and switches
- Distributed state management (of controllers)

Control/Data Separation: Myths

- **Myth:** Logically centralized route control violates fate sharing
- **Reality:** Conventional distributed routing solutions already violated these principles (e.g., OSPF areas, BGP route reflectors)
  - Separation allowed researchers to think about cleaner ways to do distributed state management

OpenFlow

- **Generality:** More functions than earlier route controllers, building on switch hardware
  - More limited flexibility, but immediate deployability
- **Technology push:** "Perfect storm" between operators, vendors, chipset designers, and researchers
- **Use pull:** Initially campuses, then data centers

OpenFlow: Intellectual Contributions

- Generalizing network devices and functions
- The vision of a network operating system
  - Data plane with open API
  - State management layer
  - Control logic
- Distributed state management techniques

OpenFlow: Myths

- **Myth:** First packet must go to the controller.
- **Reality:** No assumptions about granularity of rules or whether the controller handles traffic.

- **Myth:** Controller must be physically centralized.
- **Reality:** Deployments have distributed controllers.

- **Myth:** SDN is OpenFlow.
- **Reality:** OpenFlow is an instantiation of SDN.

Lessons

- Balance between vision and pragmatism
- OpenFlow “took off” in part because of a balance between vision and support from existing hardware
- The balance remains tenuous
  - Commodity servers
  - Programmable hardware