On the Foundations of Network Programmability

Aurel A. Lazar
Mahesan Nandikesan
Xbind, Inc. and Columbia University
Overview

- State of the art
- Observations leading to the foundations
- The foundations
- Benefits
State of the Art

• Collection of works on
  – Programmable switches and routers
    • Expose specific APIs/protocols for controlling the switching or routing table
    • Very elaborate, with no distinction between basic and derived capabilities, e.g., GSMP
  – Programmable services
    • Expose APIs for customizing services
  – Programmable architectures
State of the Art

• Issues
  – Network programmability not defined
    • The only common thread is that “something is being programmed”
  – Fundamental concepts mixed with auxiliary and derived concepts
Goals of the Foundations

• Provide a systematic development that
  – Minimally captures the basics of network control along with simple peak-rate-based QoS
  – Places the system of concepts in a well-defined hierarchy
• Capture a very general class of networks
• Define network programmability
Observations Leading to the Foundations

- Basic data transport
  - Forwarding packets/frames/streams
  - Multiplexing packets/frames/streams
  - Policing packets
    - In packet-switched networks that support QoS
Observations Leading to the Foundations

• Basic element control
  – Writing the forwarding table
    • Applies to both connection-oriented and connection-less networks
  – Writing the policing table
    • Applies only for packet-switched networks that support QoS
Observations Leading to the Foundations

• Basic network control
  – Discovering the network topology
  – Creating graphs (communication channels)
    • Applies both to connection-oriented and connectionless networks
    • In connectionless networks, graph creation is initiated by the routing system (e.g., OSPF).
    • In connection-oriented networks, graph creation is typically initiated by end-users.
Observations Leading to the Foundations

• Common Framework
  – The previous slides presented the minimal set of operations necessary to support a network with simple peak-rate-based QoS.
  – The minimal set is common to various multiplexing technologies
    • Packet multiplexing
    • Time-division multiplexing (TDM)
    • Wavelength-division multiplexing (WDM)
Observations Leading to the Foundations

• Extensions
  – Could be made for supporting other features, e.g., for
    • Exploiting multiplexing gain
Structure of the Foundations

• Network programmability is any approach to network control that takes the following hierarchical development
  1. Network resource model
  2. APIs to the resource model (vertical APIs)
  3. Algorithms for resource discovery and graph building
  4. APIs for distributed interaction (horizontal APIs)
Structure of the Foundations

1. Network resource model
2. APIs to the model
3. Distributed algorithms
4. APIs for distributed interaction

1. Network resource model
2. APIs to the model
3. Distributed algorithms
4. APIs for distributed interaction
Network Resource Model

• Data Transport Resources
  – Labels
    • A finite sequence of bits
    • Captures VPI/VCI, Ethernet addresses, IP addresses, DLCI, TDM time-slots, and wavelengths in WDM.
    • Not intended as an overlay concept as in MPLS
  – Bandwidth
  – Buffer space
Network Resource Model

- **Forwarding fabric**
  - Contains a table of entries of the form
    - (port, label stack) → (port, label stack, buffer)
- **Multiplexer**
  - Contains buffers, scheduler, buffer manager
  - Represents a time-slot interchanger in the case of TDM
  - Wavelength multiplexer in the case of WDM
Network Resource Model

• Policier
  – Contains a table of entries of the form
    • (port, label stack, peak-rate)
  – Does peak-rate policing for each entry
  – Can police aggregate streams by
    • Specifying a smaller label stack
Network Resource Model

A switch/router model

A multiplexer model for packet-switching and TDM
APIS TO THE MODEL

• APIs for reading the
  – Number of ports, port bandwidths, label types for each port, label ranges.

• APIs for writing and clearing the
  – Forwarding table
  – Policing table
Algorithms for Resource Discovery and Graph Building

- Topology discovery
  - E.g., link-state technology
- Graph building
  - Algorithms such as those used by PNNI, RSVP, but with altered syntax to match the model.
**APIS FOR DISTRIBUTED INTERACTION**

- Topology discovery and graph building algorithms require peer-to-peer communication
  - APIs for peer-to-peer communication instead of protocols
  - These APIs are algorithm-specific
The above development

- Supports the minimal operations required for network control with peak-rate allocation
- It can be extended to support
  - Multiplexing gain
  - Capacity and usage based routing
  - Automatic protection switching
  - Other features
Benefits

- Conceptual clarity
- Simplicity of writing network-control software
- Separation of control algorithms from network state
Comparison with Active Networks

• Active Networks
  – Have a program execution environment
    • Allow for fairly general programs to be uploaded
  – Result: Flexibility but also Complexity

• Programmable networks
  – Restricted goal: build graphs (communication channels)
  – Result: Simplicity
Summary

• Foundations
  – Based on a well-defined hierarchical framework
  – Presented in great generality
  – Answers the question
    • What is network programmability?