Network Function for Reliable and Secure Control Messaging over Commodity Transport

Deniz Gurkan, Stuart Baxley, and Nicholas Bastin

March 3, 2017

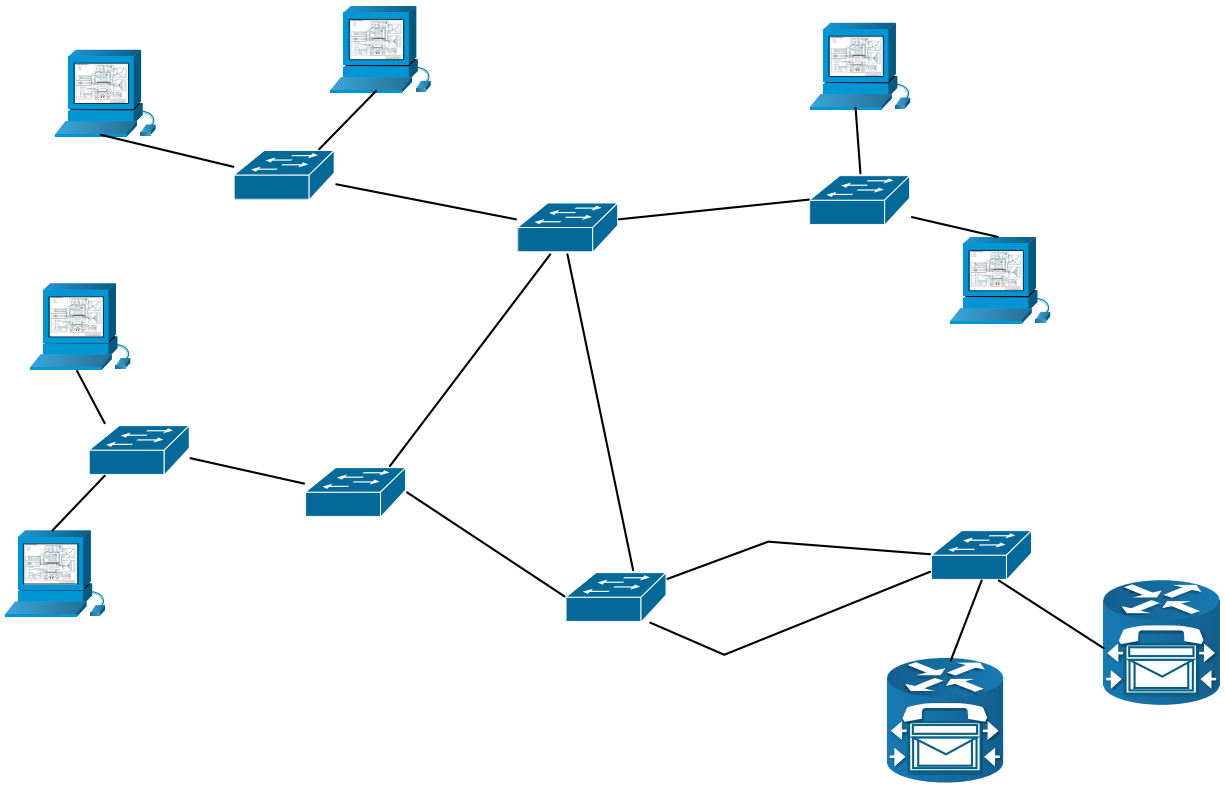
dgurkan@uh.edu

Agenda

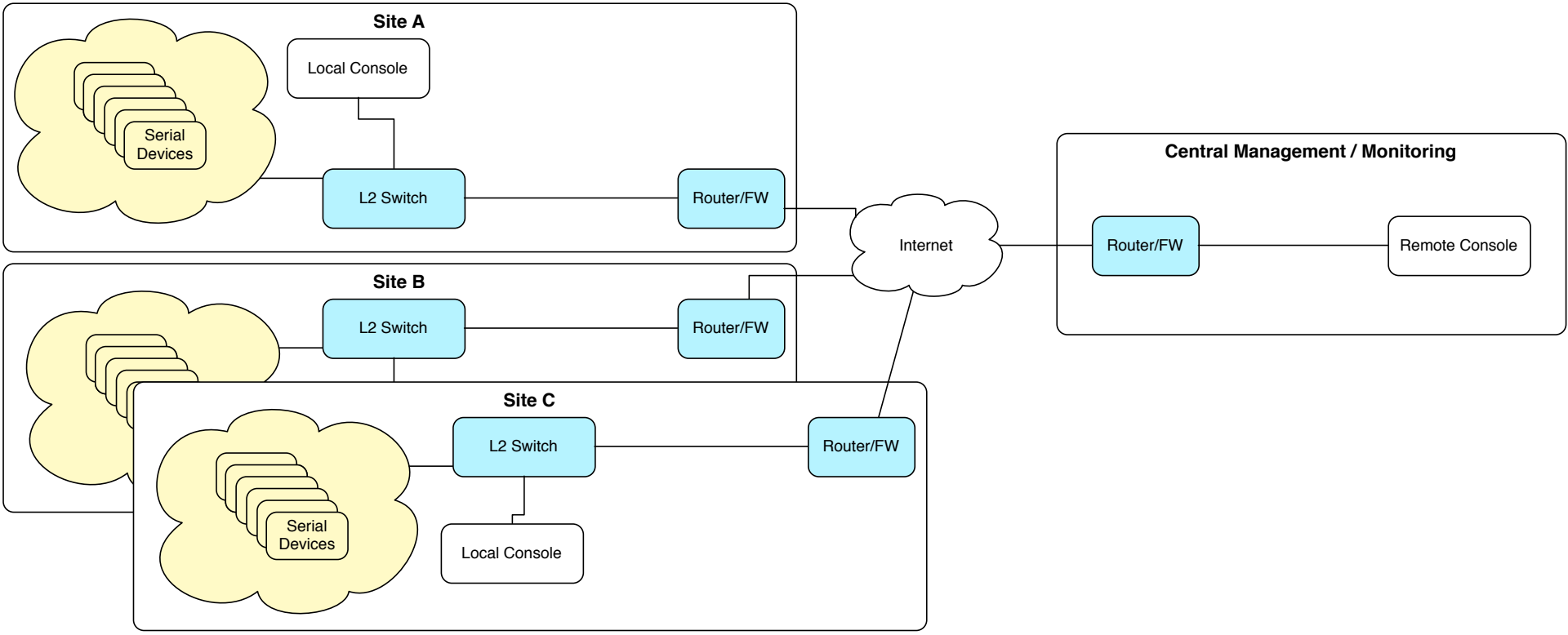
- Research and development of a network function
- Oil/Gas ICS
- Network function research
- Summary
- Testbed overview

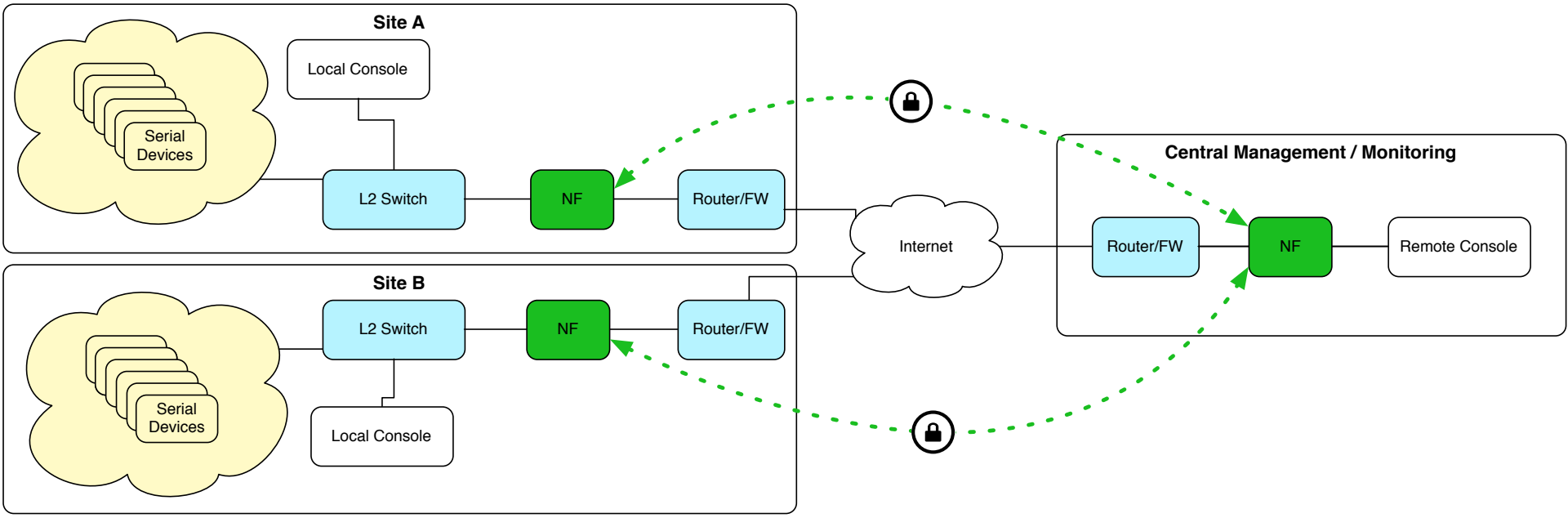
Development of a Network Function

- **Network Function:** a logical operation to be performed on particular flow(s) at particular vertices on a network graph

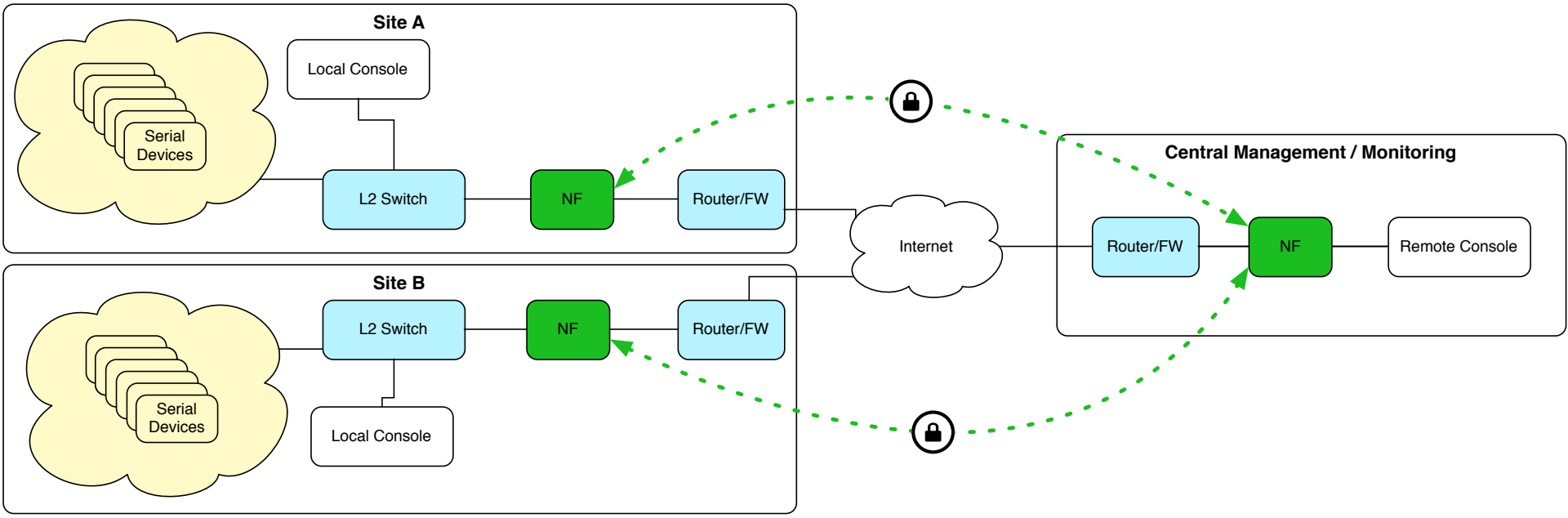


Oil/Gas ICS under Consideration
Local and Remote Sites





- No need to replace existing equipment to gain functionality
- Single point of testing and certification
- Single point of configuration for an entire site
- No need for power-hungry CPU or added cost to individual devices



- No need to replace existing equipment to gain functionality
- Single point of testing and certification
- Single point of configuration for an entire site
- No need for power-hungry CPU or added cost to individual devices

- Guarantee in-order delivery of individual messages
- Per-device/flow or per-site ordering
- Wide options for site-to-site encryption
- Possible transparent (port-mirroring) operation

Network Function Research

- Agree on a problem definition
- Define deployment constraints
- Identify solution requirements
- Logical function design
- Validation of approach
- Implementation
- Testing

Network Function Research

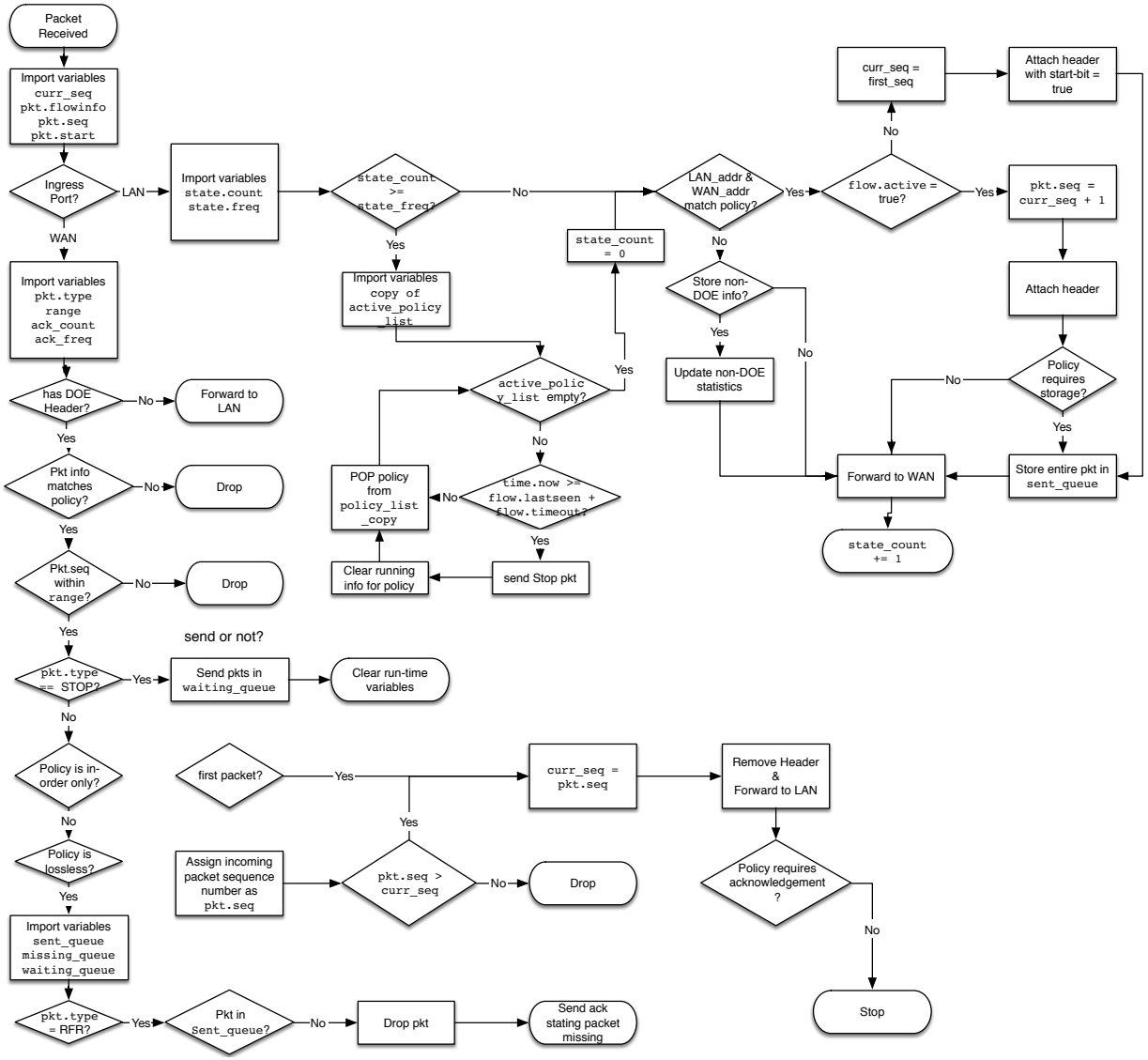
- **Problem**: without changing the existing infrastructure, insert a network function that will ensure in-order and lossless delivery of packets to the central management/ monitoring from sensors at sites in a secure manner

Constraints and Requirements

- Vendor agnostic
- Limited available compute resources
- No impact on existing redundancy

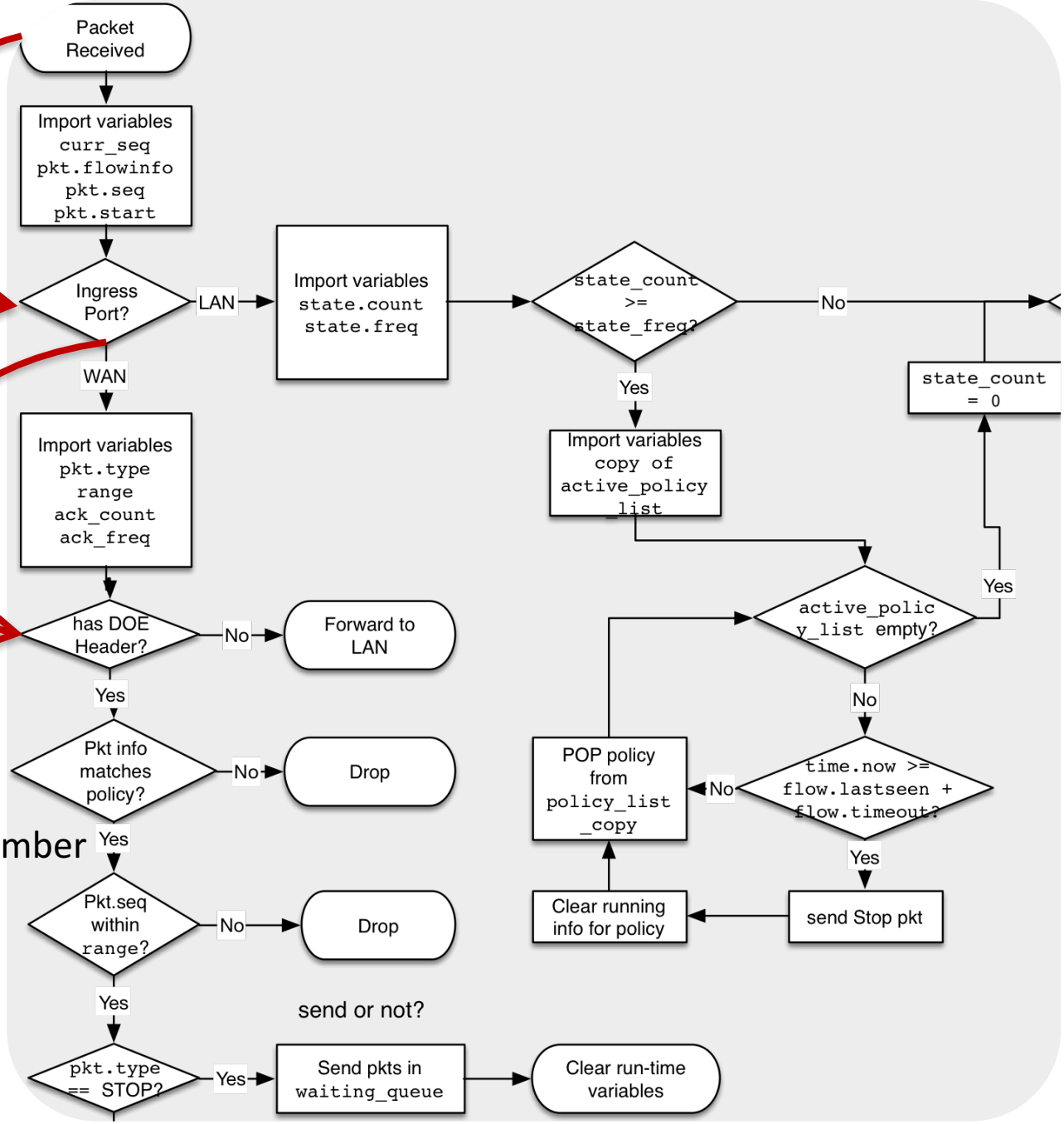
- Reliable delivery over lossy infrastructure
- Secure delivery over public networks

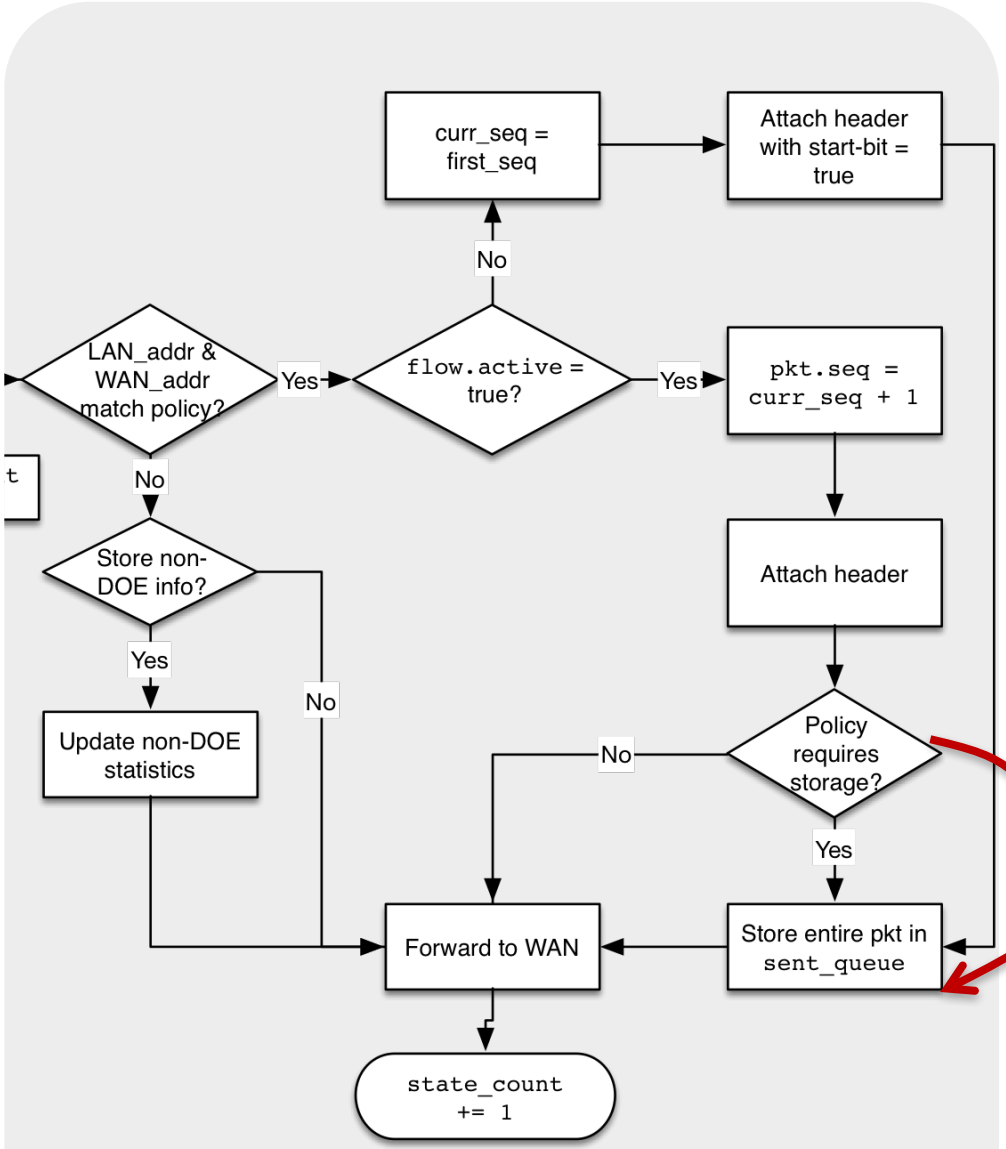
Logical Function Design



Local packet
vs
Remote packet

Correct format
of sequence number

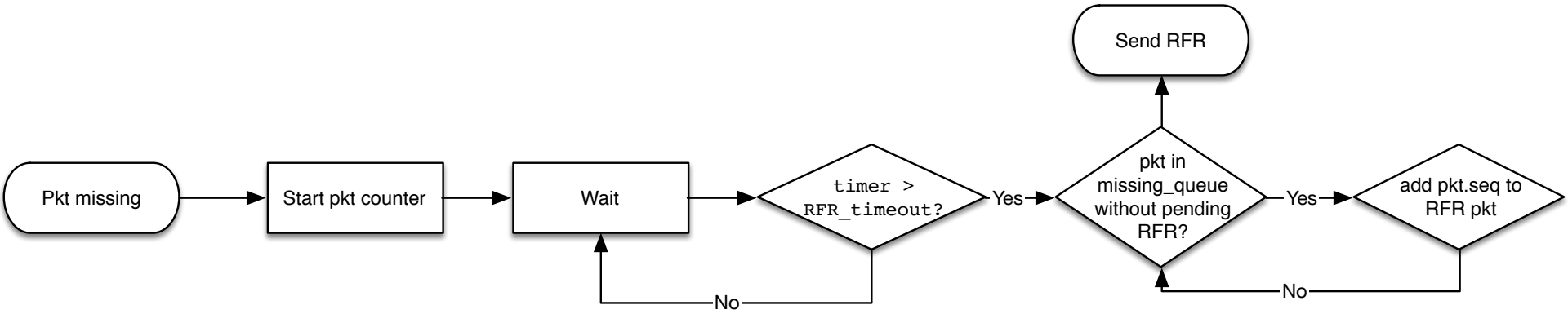




Store per policy

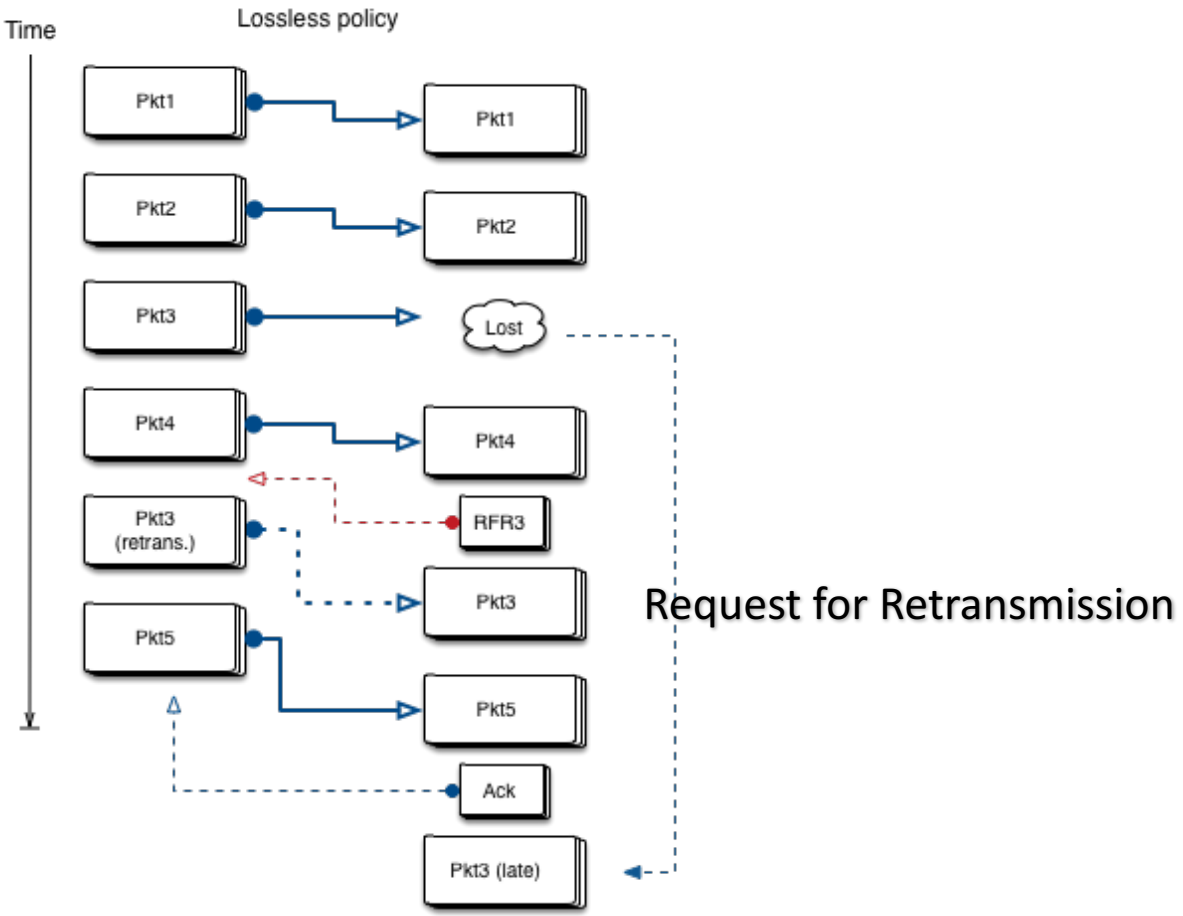
Map out <u>all policy</u> at the development phase on a flow chart for NF implementation

Policy on Request for Retransmission Tracking the State

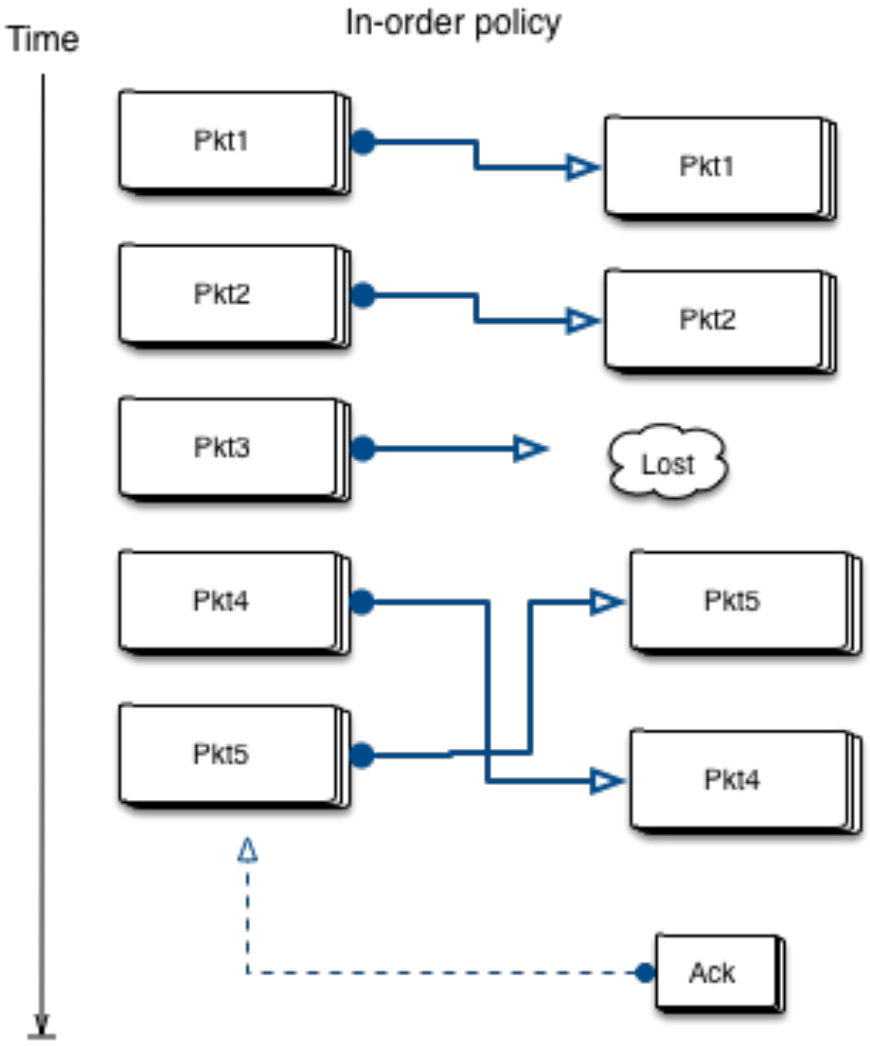


Keep track of state of missing packets in the NF while handling the request for retransmission packets and sequence number of missing packets

Validation of Approach



Validation of Approach



Network Function Development

- Lossless:
  - Store a number of packets at the local NF to be ready to retransmit when there is a loss reported from the remote NF
  - While transmission is successful, free up storage buffer at local site
  - Keep track of retransmissions and acknowledgements to guarantee smooth operation

Network Function Development

- In-order delivery:
  - Number sensor flow packets using a sequence number field in packet header
  - Keep track of sequence numbers in transit for a number of packets in the flow
  - When remote site reports an out-of-order/loss, retransmit to ensure ordering of packets

Implementation: Testing and Validation of NF

- Create emulated topologies with end hosts to create sample unreliable transfer scenarios in oil/gas ICS
- Observe and examine loss and other conditions that impact resilience of the network
- Validate the cybersecurity goals on the University of Houston's network research testbed

Network Science and Engineering Research Testbed

- Programmable network function surfaces
- Instantiate network graphs:
  - with extensive control, management, and programmability over vertices
  - various impairment emulation on the edges
- Advanced orchestration of resource allocation
- Real network behavior for applied research
- Realistic network function behavior observations for instruction of computer networking

Summary and Future Outlook

- We are working on providing a network function to provide solutions to the lossy commodity transport issues in industrial control systems of oil and gas industries.
- Our logical function design addresses loss and in-order delivery of messages in ICSs.
- We use our testbed to validate our approach.

University of Houston
SDN Lab Testbed Overview

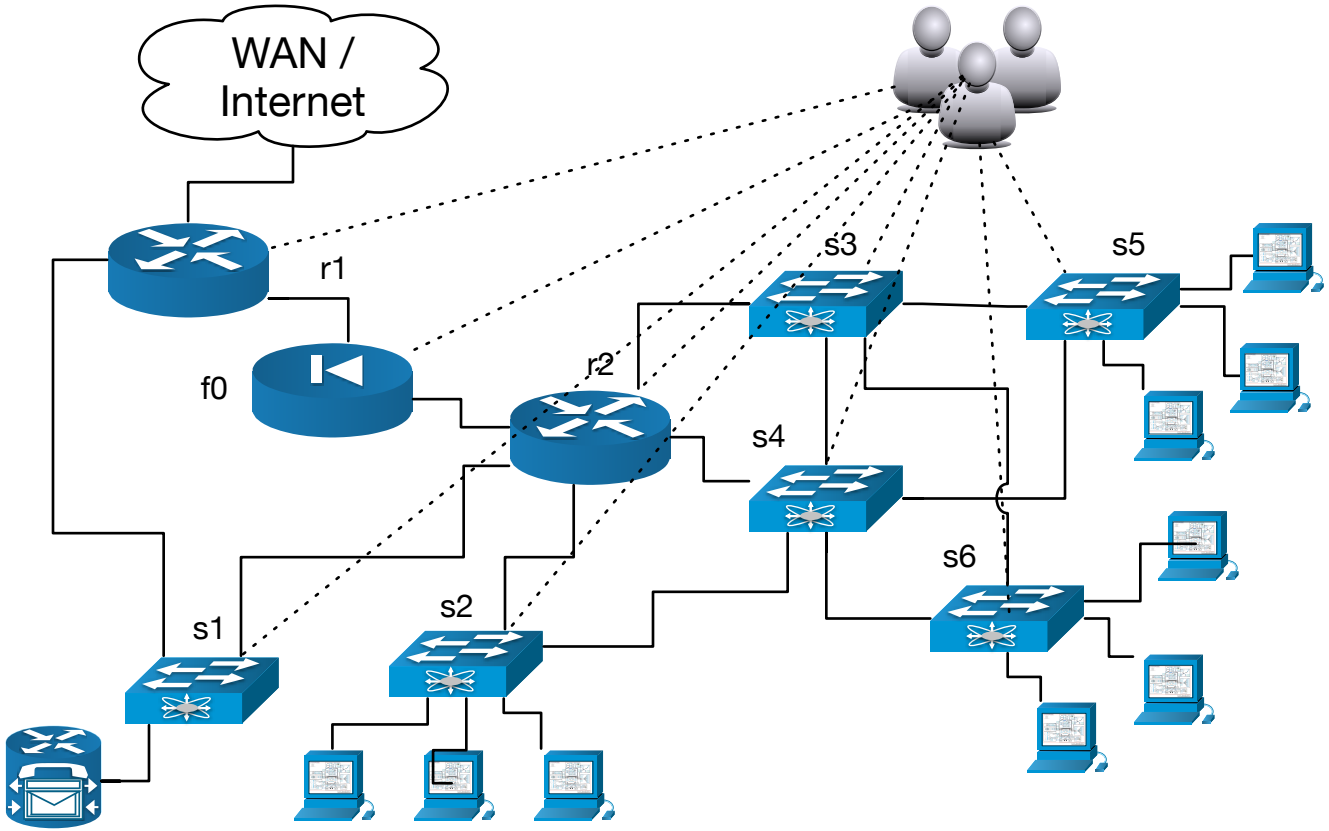
Testbed Infrastructure

- Over 1000 1Gb and 10Gb switch ports from Brocade, Cisco, Dell/Force10, HP, Intel and Pica8
- Over a dozen SDN switches
- A variety of specialized forwarding devices (NPUs, hybrid server-switches, etc.) from Caros, Cavium, Freescale, Intel, and Znyx
- Over 250 general purpose CPU cores and 1.5TB of ram across two dozen servers
- Over 100TB of raw storage capacity and 24 line-rate taps

SDN Lab Software Solutions

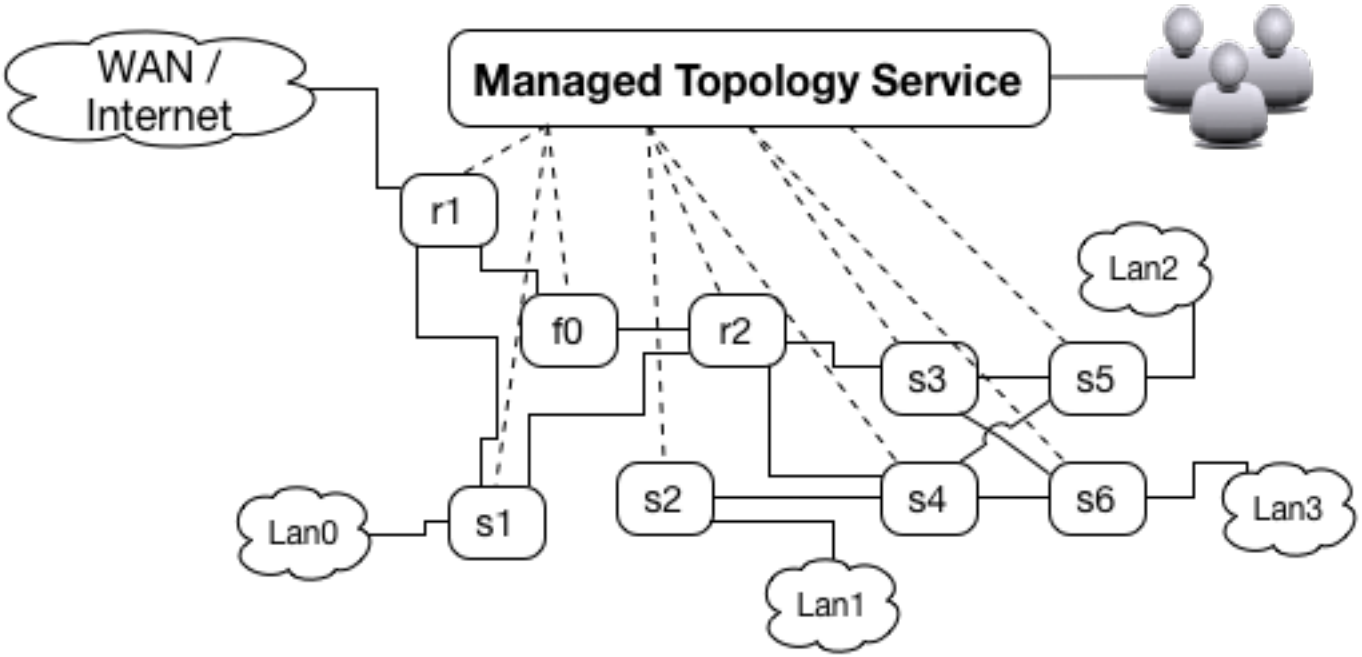
- Multi-vendor network orchestration system
- Network performance data analysis tools
- Experiment harness for repeatable and verifiable research data

Network Orchestration System



A manual configuration and management of devices lead to rigid and vendor-dependent solutions that limit the flexibility and are error-prone when ICS environments evolve

Network Orchestration System



Our resource orchestration system can manage network graphs with NFs at vertices and programmable links (i.e., at-will impairment emulation) at edges