Net SILOs: An Architecture to Enable Software Defined Optics

George N. Rouskas

Department of Computer Science
North Carolina State University

http://net-silos.net/

Joint work with: Ilia Baldine (RENCI), Rudra Dutta (NCSU), Dan Stevenson (RTI), Anjing Wang (NCSU), Manoj Vellala (NCSU)
Outline

- Context: The Clean-Slate Debate
- Motivation: Software Defined Optics
- SILO Network Architecture: The Story So Far
- Summary and Future Directions
The Internet is broken!
The Internet is broken! (has ossified / reached an impasse)
The Internet is broken!

Security is a mess: it is hard to

- identify users
- prevent them from causing harm
- hold them accountable
The Internet is broken!

Middleboxes violate end-to-end principle:

- firewalls
- NAT
- proxies
The Internet is broken!

Fixed layer architecture is outdated
The Internet is broken!

Fixed layer architecture is outdated

```
   App    App    App
   Transport
       ssh
   Network
   Data Link
   Physical
```
The Internet is broken!

Fixed layer architecture is outdated

Diagram:
- Physical layer
- Data Link layer
- Network layer (MPLS)
- Transport layer
- Application layer
The Internet is broken!

Cross-layer interactions difficult: TCP over wireless
The Internet is broken!

Clear need for clean-state initiatives → NSF FIND, EU FIRE, · · ·

1. research in new network architectures
2. large-scale experimental facilities → GENI
The Internet is doing just fine, thank you!
The Internet is doing just fine, thank you!

- Biological metaphor: mutation and natural selection
- Evolutionary designs: more robust, less expensive
- Mid-layer protocols must be conserved – not ossified
  → innovation at lower/upper layers of architecture
The Internet is doing just fine, thank you!

- Biological metaphor: mutation and natural selection
- Evolutionary designs: more robust, less expensive
- Mid-layer protocols must be conserved – not ossified
  → innovation at lower/upper layers of architecture

→ Evolution beats revolution
Our View

Internet architecture successful in accommodating change
Our View

- Internet architecture successful in accommodating change
- But: current practice of patches/tweaks cannot continue forever
Internet architecture successful in accommodating change

But: current practice of patches/tweaks cannot continue forever

New architecture must be designed for adaptability/evolvability
Our View

- Internet architecture successful in accommodating change
- **But**: current practice of *patches/tweaks* cannot continue forever
- New architecture must be designed for *adaptability/evolvability*
- SILO objective:
Internet architecture successful in accommodating change

But: current practice of patches/tweaks cannot continue forever

New architecture must be designed for adaptability/evolvability

SILO objective:

The goal is not to design the “next” system, or the “best next” system, but rather a system that can sustain continuing change
OBS And The Layer Stack

Where does OBS fit in the stack?
OBS And The Layer Stack

Transport
Network
Data Link
Physical

Apps

?
OBS And The Layer Stack

App   App   App

Transport

Network

Data Link

Physical

?
OBS And The Layer Stack

Transport

Network

MPLS

Data Link

Physical

App App App

?
Does “TCP Over OBS” make sense?
Does “TCP Over OBS” make sense?

Yes!

- TCP carries $\approx 95\%$ of Internet traffic
- Good understanding of TCP performance is crucial
Does “TCP Over OBS” make sense?

No!

- which TCP flavor?
- which OBS flavor?
- transport and OBS layers must be optimized for each other
- not as straightforward as “TCP over wireless”
Optical substrate can no longer be viewed as black box
Software Defined Optics

*Optical substrate can no longer be viewed as black box*

*Collection of intelligent and programmable resources:*

- Optical monitoring
- Sensing mechanisms
- Amplifiers
- Impairment compensation devices
- Tunable optical splitters
- Configurable add-drop
- Programmable mux-demux (e.g., adjust band size)
- Adjustable slot size
Optical substrate can no longer be viewed as black box

Collection of intelligent and programmable resources:
- optical monitoring, sensing mechanisms
- amplifiers, impairment compensation devices
- tunable optical splitters
- configurable add-drop
- programmable mux-demux (e.g., adjust band size)
- adjustable slot size
- . . .
Cross-Layer Interactions

- Impairment-aware routing
- Traffic grooming
- Network resiliency
- ...
SILO Architecture Highlights

- Generalizes traditional layer stack:
  - **services**: building blocks of fine-grain functionality
  - **silo**: per-flow vertical composition of services
  - decoupling of layers and services

- Enables inter-layer interactions:
  - **knobs**: explicit control interfaces

- Facilitates introduction of new services:
  - **ontology**: describes services and their relationships
  - **composition algorithm** to construct silos
  - standard ontology languages and reasoning engines may be used
SILOs

Cross-Service Tuning

Tuning strategies, hints

Physical Channels

silo & service mgmt

Composability Constraints

Net SILOs: A New Network Architecture

Net SILOs: A New Network Architecture

Ontology

- service
- method
- control agent
- policies

silos
- application
- physical layer

Net SILOs: A New Network Architecture

Service Composition

Constraints on composing services $A$ and $B$:

- $A$ requires $B$
- $A$ forbids $B$
- $A$ must be above (below) $B$
- $A$ must be immediately above (below) $B$
- Negations, AND, OR

Minimal set:

- Requires, Above, ImmAbove, NotImmAbove

All pairwise condition sets realizable

- Forbids = $(A$ above $B)$ AND $(B$ above $A)$
- Above = NOT Below
Composition Problem

- Given: a set of essential services ← application
- Obtain a valid ordering of these and additional services
  - or, identify conflicts with constraints
- Simple composition algorithm
SILO Software Prototype

http://net-silos.net/
Vision – enable flexibility, evolution: “design for change”
- fine-grain, reusable services, explicit control interface
  - enables experimentation, flexibility, community of innovation
- per-flow service composition (silos)
  - ease of evolution, policies

Framework – provide architectural support to vision:
- constrained composition
- commoditize cross-layer interaction / optimization

Ongoing efforts:
- extend the prototype
- new research directions: software defined optics, virtualization
- influence GENI development efforts