Net SILOs: A Network Architecture for Advanced Cross-Layer Experimentation

George N. Rouskas

Department of Computer Science
North Carolina State University
http://net-silos.net/

Joint work with: Ilia Baldine (RENCI), Rudra Dutta, Anjing Wang, Mohan Iyer (NCSU)
In Search of Next Generation Internet

Early Pioneer Work

NewArch (DARPA)  SIGCOMM FDNA  NSF FIND  NSF GENI  Pouzin Society

2000  01  02  03  04  05  06  07  08  09

Euro–NGI  Euro–4WARD  Euro–FIRE  Asia Future Internet Forum
Outline

Motivation: The Layering Conundrum

Context: The Clean-Slate Debate

Net SILOs: The Story So Far

Summary and Demo
The Internet is Broken!
The Internet is Broken!

Fixed layer architecture is outdated

- Physical
- Data Link
- Network
- Transport
- App
- App
- App
Challenges with Current Protocol Stack

1. **Evolution**: function-heavy protocols with built-in assumptions
2. **Experimental implementations**: require kernel modifications
3. **Cross-layer design**: lack of inter-layer interactions/controls
Protocol Evolution: Transport

Several distinct functions:
- identify application endpoints (ports)
- e2e congestion control
- multi-homing (SCTP)
- reliability semantics (TCP, RDP, SCTP, etc)

→ evolution of individual functions affects entire transport layer

Built-in assumptions about IP addresses
→ transition to IPv6, support for mobility difficult
Experimental network designs crucial for:
- gaining insight
- understanding protocol operation
- discovering new knowledge firmly rooted in physical world

Implementations on commodity HW/SW remain challenging:
- require modification of OS kernel
- involve significant expertise
- limit ability to “play” with network stack
Cross-layer design a major research theme over last decade:
- wireless networks
- TCP congestion control
- optical networks (next slide)
- ...

Adoption of ideas in operational networks quite slow:
- no interfaces for inter-layer interactions/cross-layer controls
- lack of experimental work
  $\rightarrow$ reliance on simulation with invalid assumptions
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:
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
  - ...
Cross-Layer Interactions

- Impairment-aware RWA and network design
- Placement of optical sub-systems (converters, amplifiers, regenerators)
- Traffic grooming
- Inter-layer QoS and traffic engineering
- Optical layer multicast
- Multi-layer failure localization and recovery
- ...
The Internet is Doing Just Fine, Thank You!

Net SILOs: A Network Architecture for Advanced Cross-Layer Experimentation

NGI 2009, July 3, 2009 – p.11
The Internet is Doing Just Fine, Thank You!

- Biological metaphor: mutation and natural selection [Dovrolis 2008]
- Evolutionary designs: more robust, less expensive
- Mid-layer protocols must be conserved – not ossified
  → innovation at lower/upper layers of architecture
Clean-State Initiatives

Design the Internet from scratch:

- Research in new network architectures $\rightarrow$ FIND (US)
  - security, addressing, protocols, routing, economics, theory, · · ·
- large-scale experimental facilities $\rightarrow$ GENI (US)
  - control plane, virtualization, slicing, · · ·
New abstraction: organize protocols in **heaps**, not stacks

Richer interactions among protocols → flexibility

Require new system-level implementations

**Role-Based Architecture (RBA) [BFH 2003]**
Meta-protocol: generic protocol layer with basic services

Each layer in stack → appropriately configured instantiation

Allows reuse, cleaner cross-layer interactions, dynamic composition

---

MP-4

MP-3

MP-2

MP-1

Physical
**Recursive Network Architecture (RNA) [TP 2008]**

- **Meta-protocol**: generic protocol layer with basic services
- Each layer in stack $\rightarrow$ appropriately configured instantiation
- Allows reuse, cleaner cross-layer interactions, dynamic composition

![Diagram showing meta-protocol layers: MP-4, MP-3, MP-2, MP-1, Wireless]
Recursive Network Architecture (RNA) [TP 2008]

- **Meta-protocol**: generic protocol layer with basic services
- Each layer in stack → appropriately configured instantiation
- Allows reuse, cleaner cross-layer interactions, dynamic composition

---

Layering As Optimization Decomposition

- Protocol layers integrated into mathematical framework
  [CLCD 2007] [LSS 2006]
- Global optimization problem: network utility maximization
- Decomposition into subproblems → layering
  - optimal modules (protocols) map to different layers
  - interfaces between layers coordinate the subproblems
Layering As Optimization Decomposition

Clean-state optimization $\rightarrow$ layered network architecture
- optimal layering $\neq$ TCP/IP stack
- various representations of optimization problem $\rightarrow$ different layered architectures
- (loose) coupling among layers $\rightarrow$ cross-layer considerations
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
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
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
- New architecture must preserve/generalize layering
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
- New architecture must preserve/generalize layering
- SILO objective:
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*
- New architecture must *preserve/generalize* layering
- 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
SILO Architecture Highlights

**Building Blocks:** services of fine-grain functionality

**Design Principles:**
1. Generalize traditional layer stack
2. Enable inter-layer interactions:
   - **knobs:** explicit control interfaces
3. Design for change:
   - facilitate introduction of new services
4. Separate **control** from **data** functions
Generalization of Layering

- **Silo**: vertical composition of services
  - preserves layering principle

- **Per-flow** instantiation of silos
  - introduces flexibility and customization

- **Decoupling** of layers and services
  - services introduced at point in stack where necessary
Silos: Generalized Protocol Stack Stacks

Cross-Service Tuning

Knobs

Silo & Service Mgmt

Composability Constraints

Physical Layers

App

S1

S4

S5

S7

S8

S2

S3

S6

S7

S9
Knobs: explicit control interfaces
- adjustable parameters specific to functionality of service
- enable info exchange among services

Algorithms may optimize jointly the behavior of services in a silo
Upward information passing
Downward information passing
Inter-Layer Interactions (2)

Up-and-down information passing

[Diagram showing up-and-down information passing between layers]
Silo-wide optimization/calibration
Design for Change

- Architecture does not dictate services to be implemented
- Provide mechanisms for:
  - introduce new services
  - compose services into silos
- **Ontology** of services: describes
  - service semantics $\rightarrow$ function, data/control interfaces
  - relationship among services $\rightarrow$ relative ordering constraints
Ontology – Networking Knowledge

Net SILOs: A Network Architecture for Advanced Cross-Layer Experimentation

NGI 2009, July 3, 2009 – p.23
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
Service 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 implemented
- Ongoing research in formalizing the problem
The SILO Hourglass
The SILO Hourglass

Applications

SILO Universe

SILO

Transport technologies

SONET
OTN
PPP
802.11
802.16
Ethernet

Physical interfaces
Net SILOs: A Network Architecture for Advanced Cross-Layer Experimentation

http://net-silos.net/
SILO As a Research Tool

- Deploys in a slice
- Researcher brings:
  - custom services
  - tuning algorithms
  - ontology updates
- Connect to measurement framework → cross-layer protocol experimentation tool
Summary

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

- New research directions
  - silos in the core
  - software defined optics
  - virtualization and slicing
- Extend the prototype
  - portfolio of reusable services
  - optical testbed deployment → breakable experimental net (BEN)
- Explore synergies with other (FIND) projects