Architectural Support for Internet Evolution and Innovation

George N. Rouskas

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

Joint work with: Ilia Baldine (RENCI), Rudra Dutta (NCSU), Anjing Wang (Ericsson), Mohan Iyer (Oracle)
Outline

- Motivation: Challenges with Internet Architecture
- SILO: A Meta-Design Framework
- SILO as Research Tool: Cross-Layer Experimentation
- Summary
In Search of Next Generation Internet

- Early Pioneer Work
  - NewArch (DARPA)
  - SIGCOMM FDNA
- NSF FIND
- NSF GENI
- NSF FIA
- Pouzin Society
- Euro-FIRE
- Euro-NGI
- Euro-4WARD
- Asia Future Internet Forum
1. **Evolution**: function-heavy protocols with built-in assumptions
2. **High barrier to entry**: for new data transfer protocols
3. **Cross-layer design**: lack of inter-layer interactions/controls
Accommodating New Functionality

- Deploy half-layer solutions (MPLS, IPSec)
  → layers become markers for vague functional boundaries
- Adapt existing implementation to new situations
  → TCP over wireless/large bw/delay product networks
- Implement own UDP-like data transfer
  → no reuse or kernel optimizations
- Abandon the old: new implementations for sensor networks
  → Internet balkanization
Internet architecture houses an effective design

But: it is not itself effective in enabling evolution

New architecture must be designed for adaptability/evolvability

New architecture must preserve/generalize layering

SILO objective: design for change
What is Architecture?

- Fundamental elements/principles vs. design decisions
What is Architecture?

- Fundamental elements/principles vs. design decisions
- Diverse points of view → FIND projects target: addressing, naming, routing, protocol architecture, security, management, economics, communication technologies (wireless, optical), · · ·
What is Architecture?

- Fundamental elements/principles \textit{vs.} design decisions
- Diverse points of view → FIND projects target: addressing, naming, routing, protocol architecture, security, management, economics, communication technologies (wireless, optical), · · ·
- Our definition:
What is Architecture?

- Fundamental elements/principles vs. design decisions
- Diverse points of view → FIND projects target: addressing, naming, routing, protocol architecture, security, management, economics, communication technologies (wireless, optical), · · ·
- Our definition:

  it is precisely the characteristics of the system that does not change itself, but provides a framework within which the system design can change and evolve
Meta-Design Framework

- Obtain a meta-design that explicitly allows for future change
- Not a particular design or arrangement of specific features
Obtain a meta-design that explicitly allows for future change

Not a particular design or arrangement of specific features

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 Stacks

Cross-Service Tuning

Knobs

Silo & Service Mgmt

Composability Constraints

Physical Layers

App

S1

S4

S5

S7

S8

S1

S3

S6

S7

S9

S2

S3

S6
**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
Inter-Layer Interactions (2)

Upward information passing
Inter-Layer Interactions (2)

Downward information passing
Inter-Layer Interactions (2)

Up-and-down information passing
Silo-wide optimization/calibration
Architecture does not dictate services to be implemented

Provide mechanisms to:
- 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
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

SILO Universe

SILO

Transport technologies

SONET

OTN

PPP

802.11

802.16

Physical interfaces
SILO Software Prototype

```
class SILO_API {
public:
    SILO_API() {
        //siloi_id = 0;
        _request_id_init=0;
    }
    //build up a set of required/forbidden constraints
    //such as: APP requires/forbids services A\->B
    ERROR_CODE create_required_srv (int request_id
    List<srv_ID srv_list),
    ERROR_CODE create_forbidden_srv (int request_id
    List<srv_ID srv_list),
    ERROR_CODE release_request (int request_id);
//...
};

class SILO_Knob
{
public:
    SILO_Knob() { ; } 
    inline int get_max() { return _knob_intf_max; }
    inline int get_value() { return _knob_intf_value; }
    inline bool set_value(int mValue) {
        _knob_intf_value = mValue;
        return true;
    }
    //...
};
```

// manages a collection of silos and passes
// data through them

```
class SiloManager {
public:
    typedef unsigned int RecipeId;
    typedef unsigned int SiloId;
    static SiloManager *instance();

    // process data
    void ProcessData(RecipeId &recipeId, 
    SiloId &siloId, 
    unsigned char *buf);
    void ProcessXBuffer(RecipeId &recipeId, 
    SiloId &siloId, 
    unsigned char *buf);
```
Prototype Architecture

- Application
  - SILO API
  - SILO Management Agent
  - SILO Tuning Agent
  - SILO Construction Agent

- Tuning Strategies Storage

- Data and control flow:
  - Packet traffic: data and control channel
  - Silo request/recipe
  - Knob descriptions
  - Ontology access
  - Optimization policies
SILO As a Research Tool

- Control the substrate
- Provide information on substrate measurement capabilities
- Sliver substrate measurement capabilities
- Moderate access to the slice
- Act on behalf of the experimenter
- Export unified measurement interface
- Request specific measurements
- Provide toolkit for cross-layer experimentation
- Perform experiment in a slice

Substrate w/ programmable measurement devices

Integrated Measurement Framework

GENI Control Framework

SILO

EXPERIMENTER
IMF Demo – Results

![Graphs showing data over time](image-url)
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 and scalability
  - policy enforcement through composition constraints
  - (generalized) virtualization as a service

- Extend the prototype
  - portfolio of reusable services
  - optical testbed deployment → breakable experimental net (BEN)