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
- Euro−FIRE
- Euro−NGI
- Asia Future Internet Forum

- Pouzin Society
- NSF GENI
- NSF FIA
- Euro−4WARD

Architectural Support for Internet Evolution and Innovation
Challenges with Current Architecture

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
Our View

- 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 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
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
Downward information passing
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

Service

Method

Control agent

Policies

Application

Silos

Physical layer
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
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

Input

Output
The SILO Hourglass
The SILO Hourglass
SILO Software Prototype

```c
class SILO_API {
public:
    SILO_API() {
        //siloid = 0;
        _request_id_init = 0;
    }

    //build up a set of requires/forbids 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);
    ...}

    // register a silo, load the needed DBS
    RecipeId RegisterRecipe(const string &recipe);
    // remove a recipe
    void DestroyRecipe(const RecipeId &recipeId);
    // create a new silo given a recipe ID
    Siloid CreateSilo(const RecipeId &recipeId);
    // delete a silo
    void DestroySilo(const Siloid &siloid);
    Siloid GetRecipeIdForSiloid(const Siloid &siloid);

    // 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 ProcessDataForSiloid(const Siloid &siloid,
    unsigned char *buf);
    void ProcessDataForSiloid(const Siloid &siloid,
    unsigned char *buf);

    class SILO_Knob {
public:
    SILO_Knob() {};
    _knob_int fMax;
    _knob_int fValue;
    inline int get_max() {return
    _knob_int fMax;}
    inline int get_value() {return
    _knob_int fValue;}
    inline bool set_value(int nValue) {
    _knob_int fValue = nValue;
    return true;
    }
    ...}

    class SILO_Config {
private:
    ...}
```
SILO As a Research Tool

Control the substrate

Provide information on substrate measurement capabilities

Sliver substrate measurement capabilities

Moderate access to the slice

Export unified measurement interface

Request specific measurements

Provide toolkit for cross-layer experimentation

Perform experiment in a slice

Experimenter

Substrate w/ programmable measurement devices

Integrated Measurement Framework

GENI Control Framework

SILO
IMF Physical Infrastructure

- VOA
- SOA
- PCB
- SOA-Control Prototype Board
- NetFPGA
- Polatis Switch
- Infinera DTN
IMF Cross-Service Demo

Legend:
- BER – Bit Error Rate
- DTN – Digital Transport Node
- SOA – Semiconductor Optical Amplifier
- VOA – Variable Optical Attenuator
- SCPI - Standard Commands for Programmable Instruments
- GPIB - General Purpose Interface Bus

Physical Devices
- SOA
- VOA
- IF switch
- Network FPGA
- SSH

Software
- XMPP
- XMLRPC
- Management Network
- Optical Data Path @ BEN

Architectural Support for Internet Evolution and Innovation
IEEE ANTS, December 16-18, 2010 – p.23
IMF Demo – Results
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 \(\rightarrow\) breakable experimental net (BEN)
Upcoming Book

Next-Generation Internet Architectures and Protocols

Edited by Byran Jansamorothy, George N. Bourkas Bruskas and Krishna Moorthy Sivlingam