

Future Internet Architecture (FIArch)

Rationale and Methodology

Dimitri Papadimitriou
Alcatel-Lucent Bell

Approach



Methodology: 4 Steps

Step 1: Motivations and design objectives

Step 2: Common architectural principles

Step 3: Architectural components

Step 4: Putting components into common model

Step 1: Motivations and design objectives



Goals:

- Understanding actual motivations and rationalization: identification and characterization of presumed **architectural "limitations"** (with quantitative and qualitative arguments)
 - Setting design objectives (and associated metrics) starting from current architecture objectives of the Internet, **recognizing thus its potential**
-

Architecture – What does we mean ?



Architecture

- Formal grouping of a function space, a state space and objects/information, as well as their respective distribution that characterizes its domain of applicability
- Specification of the associated functional, object/informational and state models leads to a set of architectural components (i.e. procedures, data structures, state machines, etc.) together with the characterization of their interactions (i.e. messages, calls, events, etc.) and their distribution

Note:

Principles (often) drive the design of these models and their associated components

What is an Architectural Limitation ?

- Limitation refers to functional, structural or performance restriction or constraint that
 - **cannot be resolved** with current or clearly foreseen paradigms as far as our understanding/knowledge goes
 - Fundamental limitation (of Internet architecture)
 - **could be resolved** (as far as our understanding/knowledge goes) by replacing and/or adding/removing a component of the architecture that would in turn **change its global properties**
 - Challenging limitation
 - Example: to overcome IP address overload : loc/id split
- >< **Re-engineering:** technique/method for overcoming limitation by replacement of an instance of an existing architectural component that would not change its global properties
- Examples: replacement of IPv4 by IPv6 (no change to "IP") – over-dimensioning/over-provisioning
-

Fundamental Questions

- **Question 1:** might the architecture itself (and its properties) become the limiting factor of Internet growth and deployment of new applications ?
- **Question 2:** Internet architecture incrementally and reactively **extended but up to which limit?** vs impact of Internet architecture **limits but up to which extend?**

- Studies on research results impact have shown that better performance or functionality define **necessary but not sufficient conditions for change** in the Internet architecture (and/or its components)

=> Demonstrate architectural limits and characterize them is a **MUST**

What's different ...



- ... from EU FP7 (FIA, EIFFEL), US (NewArch, FIND, GENI, etc.), Asia (AKARI, etc.), etc.
 - *That's not the actual question:* here trial to consolidate material to define a **common problem statement with shared design objectives**
 - Note: it's easy to specialize (e.g. IoT vs Core of the Internet) ... the real exercise is to identify the **commonly shared baseline**
-

This being said

Is the Internet architecture under pressure ?

... Think of

- Emergence of new needs at both functional and performance level (scaling, reliability, availability)
 - Increasing cost and complexity, as well as performance decrease resulting from Internet growth
 - Existing and foreseen (functional and performance) limitations resulting from the Internet architectural principles, model, and components...
 - Increasing heterogeneity of running conditions as well as the increasing occurrence of unattended and unexpected events
 - Real time systems, etc.
-

Base Functionality

- Starting point of architectural analysis is functionally-driven
 - 4 base functionality
 - Processing/handling
 - Storage
 - Transmission
 - Control
- } Operating on Data

“*data*”: refers to any structured group of bits
a.k.a. packets, traffic, information, content
(audio, video, multimedia), etc.

Base Functionality

- **Processing/handling** of “data”
 - Refers to forwarders (e.g. routers, etc.), computers (e.g., terminals, servers, etc.), CPUs, etc. and handlers (software programs/routines/web services) that generate and treat data
 - **Storage** of “data”
 - Refers to memory, buffers, caches, disks, etc. and associated structures (words, pages, files, etc.)
 - **Transmission** of “data”
 - Refers to physical and logical transferring and exchange of data
-

Base Functionality



- **Control** of processing, storage, transmission
 - refers to action of observation (input), analysis, and decision (output) whose execution affects the running conditions of these functions

Note

- Communication function \equiv combination of processing, storage, transmission, and control functions
 - Data communication \equiv combination of processing, storage, transmission (and their control) applied to data
- => Communication (function) and "network" are not equivalent terms
-

Step 1 In a nutshell...



- Characterize
 - What are Internet architecture fundamental limitations wrt base functionality ?
 - -as much as we can- with currently available qualitative and quantitative data how the current architecture performs wrt base functionality ?
 - Derive design objective
 - Identify those that could possibly be met using current architecture and those that appear to demand a new or extended architectural foundation
 - **design objectives** = functional and performance properties as well as structural and quality criteria that architecture is expected to meet
-

Step 2: Common Architectural Principles



- Internet architecture driven by design principles
 - Suggest rules on how a designer/an architect can best structure the various architectural components
 - Describe the fundamental and time invariant laws underlying execution of engineered artifacts
 - Goal: Identify **common** design principles for architecture involving (distributed and dynamic) data processing, storage, transmission, and control functions
 - Determine additional / improvement of current architecture principles (implies also identification of hidden relationships)
-

Step 3: Architectural Components

- Architectural components (i.e. procedures, data structures, state machines, etc.) together with the characterization of their interactions (i.e. messages, events, etc.) and their distribution
 - Approach by decomposition to decrease complexity of the global problem space -> which decomposition ?
 - For each of these components there is a-priori no unique instantiation or realization possible !!!
 - Distinction between time invariant components (performance over time metrics) vs. evolvable/adaptable components (lifetime over performance metrics)
 - Performance and functionality criteria and utility to be evaluated by means of measurable metrics and results compared to their resulting cost & complexity
-

Step 4: Design model



- Combine designed components into common model baseline following common design principles
... the BIGGEST challenge !
-