Towards a User-Centric Internet Architecture

Mark Allman
International Computer Science Institute

Youngstown State University
September 2010

“Seven hundred tons of metal a day and sir you tell me the world’s changed,
Once I made you rich enough, rich enough to forget my name”
Towards a User-Centric Internet Architecture

Mark Allman

*International Computer Science Institute*

Youngstown State University

September 2010

“Seven hundred tons of metal a day and sir you tell me the world’s changed,
Once I made you rich enough, rich enough to forget my name”
Collaborators

• Aditya Akella
• Tom Callahan
• Kevin Ditraglia
• Fredrick Douglas
• Andrei Gurtov
• Joakim Koskela
• Benjamin Kuperman

• Chitra Muthukrishnan
• Pete Naegele
• Vern Paxson
• Michael Rabinovich
• Mitch Rackovan
• Michael Slattery
• Nicholas Weaver
What is “Architecture”? 

• Abstractions
  • e.g., layering

• Foundational Services
  • e.g., Domain Name System (DNS)

• Organizing Principles
  • e.g., the end-to-end principle
  • e.g., engineering for tussle
Abstractions

• General computer science principle of complexity hiding
• Applied broadly within the discipline
• Network are no different
  • e.g., protocol layering
  • e.g., AS numbers for routing
Layering

<table>
<thead>
<tr>
<th>Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
</tr>
<tr>
<td>Session</td>
</tr>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Data Link</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>
Layering (cont.)

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical</td>
<td>Physical layer deals with physical connections and operational aspects.</td>
</tr>
<tr>
<td>Data Link</td>
<td>Focuses on the transmission of data over the physical medium.</td>
</tr>
<tr>
<td>Network</td>
<td>Manages the network architecture and data transport across networks.</td>
</tr>
<tr>
<td>Transport</td>
<td>Handles the reliable transfer of data over a network, including error control.</td>
</tr>
<tr>
<td>Session</td>
<td>Defines the interaction between applications, including negotiations.</td>
</tr>
<tr>
<td>Presentation</td>
<td>Presents data in a format understandable to users.</td>
</tr>
<tr>
<td>Application</td>
<td>Enables direct user interaction and processing.</td>
</tr>
</tbody>
</table>

- **Remote Interprocess Communication (e.g., TCP/IP)**: Communication between processes on different hosts.
- **Direct Host-to-Host Communication (e.g., Ethernet)**: Communication between hosts using the same network interface.
- **Process-to-Host Communication (e.g., SMTP)**: Communication between processes and a host.
- **Host-to-Host Communication (e.g., IP)**: Communication between hosts using IP addresses.
- **App-to-App Communication (e.g., HTTP, XMPP, SMTP)**: Communication between applications.

Power, voltages, attenuation, etc. (i.e., EE stuff!)
Layering (cont.)

<table>
<thead>
<tr>
<th>Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Religion</td>
</tr>
<tr>
<td>Politics</td>
</tr>
<tr>
<td>Money</td>
</tr>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Data Link</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>
Foundational Services

• Protocol stack isn’t enough

• Need additional elements to add flexibility, functionality, scalability, etc.
  • e.g., DNS to name hosts
  • e.g., DHCP for host configuration
  • e.g., NIS/LDAP for configuration information
DNS

- Maps human understandable hierarchical names to IP addresses
- e.g., www.icir.org == 192.150.187.12
DNS (cont.)

<table>
<thead>
<tr>
<th>Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
</tr>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Data Link</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>

DNS
DNS (cont.)

<table>
<thead>
<tr>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Link</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>DNS</td>
</tr>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>Application (DNS)</td>
</tr>
</tbody>
</table>
Organizing Principles

• In addition to specific aspects of technology we develop for networked systems we also need overarching ways to think

• E.g., the end-to-end principle
• E.g., engineering for tussle
End-to-End Principle

- Keep the middle of the network simple
- Put the “smarts” at the edges
- Allows for innovation to be built on top of simple and ubiquitous core
Hop-By-Hop Example

Sender

Router 1

Router 2

Receiver
End-to-End Example
Reconciling Interests

- Observation: different entities in the network have different interests
- Observation: no one-size-fits-all way to address competing interests
- Conclusion: engineer the system to deal with competing interests
Current Architecture

- Current architecture has obviously been useful
  - formed the foundation of a system that has scaled in terms of hosts, people and content

- Is the current architecture enough?
- Can / should we evolve it to make the Internet “better” in some way?
Trends

1. Users generate the content
2. Users access the Internet from a variety of computing platforms
3. Breadth of applications is every increasing
4. Users and service providers (broadly defined) have inconsistent goals
Evolving Architecture

- Trends are very *user focused*
- Current architecture is very *host focused*

- Can we evolve the *Internet architecture* to include *users* as first-class entities across services, protocols, etc.?
Our Approach

• We have a multi-pronged approach to adapting the architecture to be user-centric

• Establishing identity

• Meta-information storage
  • e.g., naming

• Transparent networking
Establishing Identity

- Crucial problem: how do we identify people and validate transactions?

- Employ usernames, passwords, crypto (oh my!)

- Well, yes, but what about *host compromise* or *man-in-the-middle snooping*?
  - easy to lose the “keys to the kingdom”
Path Vulnerabilities

Server

Router 1

Router 2

Client
New “Paths”

- **Objective**: secure identity and transactions regardless of the state of the path

- **Two key constructs**:
  - trusted path to the user
  - independent path to the user
Trusted Path

- USB fob
  - holds users’ crypto material
  - fits on users’ physical keyrings
  - has input/output
    - speaker + button (say)
  - limited functionality
    - i.e., a few crypto functions
    - (reduced attack surface)
A = “do you really want to transfer $1M?”
Trusted Path (cont.)

- Any alterations of the audio are detectable
- Only the fob can authorize the transaction
- The only thing the network path can do is prevent communication
Trusted Path (cont.)

• We have an initial design
  • generic API to work across services
  • bill of materials: $30
    • (likely lower now)
Independent Path

• Rather than try to secure the in-band communication we rely on a second independent path to relay out-of-band confirmations
Ind. Path (cont.)

Verify User X

Bank Web Site

V-Server

Code = Y

Code = Y

Laptop

User X

Cell

Xfer

$1M

Code = Y
Meta-Information

- Observation: lots of meta-information floating around
  - names (URLs, email addresses, etc.)
  - social graph
  - configuration information
  - application state

- Storage and management are ad-hoc
  - bookmarks, address books, rc files, etc.
MISS

• We developed the Meta-Information Storage System (MISS) as a service to coherently store meta-information

• each user gets a space to populate with their information

• flat namespace

• outside specific hosts and applications

• Goal: provide a foundation to both deal with the mess and enable new functionality
Naming

- Naming network resources and services is a big mess
Naming Problems

• Problem #1: names are obtuse
• Problem #2: names are hard to share

http://www.flickr.com/photo_zoom.gne?id=1131208946&size=0&context=photostream
Naming Problems (cont.)

• Problem #3: names are globally unique, but ambiguous to people

• What is ou.edu?
  • Ohio University ??
  • University of Oklahoma ??
Naming Problems (cont.)

• Problem #4: names are intolerant of location change

mallman@cs.ohiou.edu  mal37591@ohiou.edu

mallman@lerc.nasa.gov  mallman@bbn.com

mallman@grc.nasa.gov

mallman@icir.org

mark.allman@case.edu
Naming Problems (cont.)

• Problem #5: naming is under nobody’s control
  • service providers play a part
  • e.g., “www.blogspot.com”
  • content providers play a part
  • e.g., “MyGreatVacationPictures.html”
  • consumers play a part
  • e.g., “Joe’s Blog” in the bookmarks list
A Naming Layer

• Perhaps what we need is a new over-arching namespace

• just an abstraction to existing namespaces

• A “personal namespace” that can be contained in MISS
A Naming Layer (cont.)

• Give users’ a way to name their own resources
  • independent of resource/service location
  • with context sensitive names
  • public-vs-private scoping defined by the user
Name Types

• Simple names
  • e.g., “calendar = webcal://cal.mallman....”
  • e.g., “email = mallman@icir.org”
  • e.g., “aim = myAIMhandle”

• Pointers to other namespaces
  • e.g., “Joe = NID:7a6b623df1”
Example

Mark Allman NID

Dad

web

calendar

vacation-pix

Wes Allman NID

Mark

blog
Example (cont.)

• Wes can use:
  • Mark:vacation-pix
  • Mark:web

• Mark can use:
  • Dad:blog
Implementation

• Backend MISS has been built
• Plugins to implement the naming scheme have been developed for Thunderbird and Firefox
• Open question: What would you do with a MISS-like service?
User-v-Network

• The Internet architecture calls for the network to be application-agnostic, but that is not operational reality in modern networks

• Some decry such *non-neutral* treatment

• However, these practices are reality and rooted in compelling business, economic and civic concerns

• so, the tension is likely here to stay

• Represents a tussle-space we must accommodate and not resist
Traffic Discrimination

• Typical scenario:

  • service provider takes issue with some use of the network, buys or implements some way to find the offending traffic and limits it in some way (dropping, throttling, etc.)

  • users (/applications) take issue with discrimination by the network and encode, layer and generally obfuscate their traffic to circumvent detection

• Rinse and repeat

  • standard arms race
Transparency

• We don’t need (or even want) a *neutral* network we need a *transparent* network

• I.e., users / applications can understand network policies

• I.e., the network can understand users’ / applications’ intentions
Warning

• This is a *thought experiment*
Typing

- Move away from network handling opaque blobs of bits
- Rather, the type of the bits is also exposed
  - in terms of the semantics of how those bits will be used
- Extensive set of types
  - from atomic (IP addresses) to higher-level constructs (URLs) to aggregated objects (HTTP responses)
- Exhaustive typing
  - everything is typed without exception
Dialog

• Provides a way for users and applications to communicate with the network to understand policies and adapt to particular requirements

• E.g., email
  • a user may wish to keep an email transaction private
  • an institution may require email be exposed for virus scanning
Choice

- Dialog leads to choice

- E.g., users / applications can decide to expose the required information

- E.g., users / applications can decide to use a different path (or virtual path)
Verification

- Problem: how do we know the payloads will be used as advertised?

- In the limit, this is unknowable

- We can gain confidence by using *attesters* to verify types
  - e.g., TPMs
  - e.g., TTPs
“Realization”

- XML blobs to encode messages
- Crypto to scope actors who can view a message
- But, yeah, there are issues .....
Other Ideas

• Opportunistic personas
  • better security through crypto + track records

• Better information sharing for energy-sensitive networking

• Purpose-built social networks
  • e.g., for use during emergencies
Next Steps

• Integrating the social graph across protocols, services, etc.

• User-directed protocols ("hooks")

• Networking with context

• *Can and should we evolve the network architecture to be more user-centric?*
Questions? Comments?

Mark Allman

mallman@icir.org
http://www.icir.org/mallman/

“... and I believe in the promised land”
References


• Mark Allman. Personal Namespaces. ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets), November 2007.

• Joakim Koskela, Nicholas Weaver, Andrei Gurtov, Mark Allman. Securing Web Content. ACM CoNext Workshop on ReArchitecting the Internet (ReArch), December 2009.


• Tom Callahan, Mark Allman, Michael Rabinovich, Frederick Douglas. On Grappling with Meta-Information in the Internet, July 2010. Under submission.
