XIA: eXpressive Internet Architecture - A Proposal for a Future Internet Architecture

15-441/641: Computer Networking

Lecture 25: What is Next?
Peter Steenkiste

Fall 2013
www.cs.cmu.edu/~prs/15-441-F13

Outline

• Background
• The eXpressive Internet Architecture – a proposal
  – Example and concepts
  – Research thrusts
• XIA building blocks:
  – AIP
  – Tapa

NOTE: this lecture describes a research project
This material will not be on the final exam

Key Internet Features

What we learned about the current Internet:
• Simple core with smart endpoints
• The IP narrow waist supports evolution
• Packet based communication
• All IP hosts can exchange packets
• Non-essential functions are services
• End-to-end transport protocols
• Security is not part of the architecture

But may be there are better ways ...

Outline

• Background
• The eXpressive Internet Architecture – a proposal
  – Example and concepts
  – Research thrusts
• XIA building blocks:
  – AIP
  – Tapa
“Narrow Waist” of the Internet Key to its Success

- Has allowed Internet to evolve dramatically
- But now an obstacle to addressing challenges:
  - No built-in security
  - New usage models a challenge – content and services, not hosts
  - Hard to leverage advances in technology in network
  - Limited interactions between network edge and core
- But where do we get started?

Multiple Principal Types

- Identifying the intended communicating entities reduces complexity and overhead
  - Have different forwarding semantics
- Set should be *evolvable*

Three Simple Ideas

- Support multiple types of destinations
  - Not only hosts, but also content, services, etc.
  - Not having to force communication at a lower level (e.g., hosts) reduces complexity and overhead
- Intrinsic security guarantees security properties as a direct result of the design of the system
  - Do not rely on external configurations, data bases, ..
- Flexible addressing gives network more options for successfully completing communication operations
  - Include both “intent” and “fallback” address
  - Supports evolvability, network diversity, fault recovery, mobility, ..

Multiple Principal Types - Example
Many Alternatives!

Using Principal Types

- Content and service addresses directly support cross-application service selection and caching
  - Complex today: overlay indirection infrastructure, deep packet inspection, transparent proxies, etc.
- Routing protocols for hosts, content and services
  - Metrics driving by context, different concerns
  - Public internet: policies, business, ...
  - Intra-networks: usage models, super fast recovery, ...
- Add new (custom) functionality to the network
  - E.g., caching + service -> diverse multicast variants
  - Dealing with disruptions

Security as Intrinsic as Possible

- Communication security properties are a direct result of the design of the system
  - Do not rely on correctness of external configurations, actions, data bases

Device 922504A ...
Correct content

Content 581109A ...
Correct host

Services 931399F ...
Correct service

Future Entities 3BC0344 ...

Use of Intrinsic Security

- Name-> address look automatically provides public key associated with the address
  - May not need for separate key management infrastructure
  - Can help, e.g., with network partitioning
- Changing of addresses in session in network layer
  - Sign change with private key associated with old address
- New types of intrinsic security that might
  - Variants for services, contents and hosts; new types
  - Support for existing key management processes
- Simplify comprehensive security mechanisms
Supporting Evolvability: Flexible Addressing

- Introduction of a new principal type will be incremental – no “flag day”!
  - Not all routers and ISPs will provide support from day one
- Creates chicken and egg problem - what comes first: network support or use in applications
- Solution: provide an *intent* and *fallback* address
  - Intent address allows in-network optimizations based on user intent
  - Fallback address is guaranteed to be reachable

<table>
<thead>
<tr>
<th>CID</th>
<th>AD.HiD</th>
<th>AD.HID</th>
<th>.....</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest</td>
<td>Src</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Our Solution: DAG-Based Addressing

- Uses direct acyclic graph (DAG)
  - Nodes: typed IDs (XID; expressive identifier)
  - Outgoing edges: possible routing choices
- Simple example: Sending a packet to HID₅

Support for Fallbacks with DAG

- A node can have multiple outgoing edges
- Outgoing edges have priority among them
  - Forwarding to HID₅ is attempted if forwarding to CID_A is not possible – Realization of fallbacks

Addressing Requirements

- Fallback: intent that may not be globally understood must include a backwards compatible address
  - Incremental introduction of new XID types
- Scoping: support reachability for non-globally routable XID types or XIDs
  - Needed for scalability
  - Generalize scoping based on network identifiers
  - But we do not want to give up leveraging intent
- Iterative refinement: give each XID in the hierarchy option of using intent
It Is Not Just About Architecture!

- End-to-end transport over heterogeneous networks
  - TCP works well over wired segments
  - How to better support wireless mobile users, insertion of services, vehicular, DTNs, ...
- Trustworthy network operations
  - Improve “security” broadly defined by leveraging the intrinsic security properties of XIA
  - Focus on systematic approaches to trust management and availability

Outline

- Background
- The eXpressive Internet Architecture – a proposal
  - Example and concepts
  - Research thrusts
- XIA building blocks:
  - AIP
  - Tapa

A Couple of XIA Building Blocks

- The Accountable Internet Protocol
  - Example of intrinsic security for host-based communication
- The Transport Access Point Architecture
  - Segment based Internetworking to Accommodate Diversity at the Edge, Fahad Dogar, Peter Steenkiste, CMU CSD technical report, CMU-CS-10-104, February 2010
  - Transport services for mobile and wireless users
  - Not part of the architecture, but can leverage many of its features
AIP Motivation

- Many security challenges are a result of not being able to unambiguously determine who is responsible for a specific action
  - Source spoofing, denial-of-service attacks, untraceable spam, ..
- Add accountability to the Internet architecture
- Key idea is to use self-certifying addresses for both hosts and domains
- Avoid dependence on external configurations
  - E.g. global trust authority

Self-Certifying Identifiers

- Identifier of object is public key of object
  - Convenient to use hash of object (e.g. fixed size)
  - Need way of securely mapping user readable name into the identifier
- AD is hash of public key of domain
- EID is hash of public key of host
- Provides a means of verifying the correctness of the "source" identifiers in a packet
  - Effectively by sending a challenge to the source that it must sign with its private key

Example: AD verification

- Addresses are hierarchical, similar to today’s Internet
  - But each level has a flat address, i.e. no CIDR
- Until packet reaches destination AD, intermediate routers use only destination AD to forward packet
  - Effectively uses a pointer in a stack of domain identifiers
- Upon reaching destination AD, forward based on EID

- Identify received AD: X.
- In accept cache?
- Forward packet
- Trust neighboring AD?
- Drop packet
- Send V to source
- Pass uRPF?
Verification Packet

- Router sends a packet V to Source containing:
  - Source and destination identifier
  - Hash of the packet P
  - Interface of the router
  - A secret signed by R
- Source signs V with its private key and send it back to R
  - But only if it recognizes the hash
- R verifies that it was signed correctly using the public key from the source field
- If they match, R add S to its cache

AIP Discussion

- AIP adds complexity to routers ...
  - Crypto support, caches, larger forwarding tables, ...
- ... but accountability helps address number of security challenges
  - Reduces complexity and cost in rest of networks
- Research question
  - Fast look up in large tables of flat identifiers
  - Managing keys (revocation, minting, ...)
  - Evolving of the crypto

Wireless and Mobile Challenges

- Network and device heterogeneity
  - “Wired” protocols stack may not work
- Diverse network services
  - Content retrieval, mobility services
- Relaxed synchronization end-points
  - Intermittent connectivity common case
- Topology control
  - Handoff, multi-path

Transport Access Points

- Tapa supports visible middleboxes (TAPs) that break up e-e connections in segments
- Each segment uses custom solutions for congestion, error, and flow control
- Transfer, transport layers glue segments into e-e path
  - Operate on self-certifying chunks of data (ADUs)
Unbundling the Transport Layer

- Tapa unbundles the “thick” Internet transport layer
  - Motivated by the “dumb middle” idea
- Segments support best effort delivery of “chunks”
  - Must support congestion, flow, and some error control in way that is appropriate for that segment
  - Chunks are a few KB and self-certifying
- Transfer layer supports best effort end-to-end delivery of chunks by stitching segments together
  - Naturally supports insertion of network services
- Thin end-to-end transport supports e-e semantics
  - Also flow, error, congestion control across segment path

Tapa Prototype

- Leverages Data-Oriented Transport (DOT)
  - Uses self-certifying chunks of data
  - Supports application-independent caching
- Uses diverse protocols for wireless segment
  - TCP is convenient solution for wired backbone
- Intelligent end-end transport intelligence is implemented on mobile host and TAP
  - Vehicular communication
  - Catnap battery savings

Vehicular Example

- Vehicle-infrastructure suffers from frequent interruptions, short periods of connectivity
- Vehicle optimizes transfers by explicitly managing server-TAP and TAP-vehicle transfers
  - Leverages self-certifying content identifiers

BW Discrepancy in typical end-to-end transfers

- Idle period = 3.7ms – too small for PSM
- Packet Transmission Time = 4ms

Catnap leverages this opportunity to provide up to 2-5x energy savings during data transfers
Catnap Design Overview

1. Decoupling of Wired and Wireless Segments
   2. ADU Hint -- Length of ADU
   3. Scheduler -- Decides when to send data to client

How much can the NIC sleep?

- TCP transfers remain in active state
- Transfer times do not increase with Catnap
- Sleep time with Catnap increases as transfer size increases

Tapa and XIA

- Content-centric optimizations in Tapa can be pushed “into the network”
  - Tapa can use content XIDs rather than host XIDs
  - Old APs can be listed as hints (rather than server)
- Tapa needs support from services on/near APs
  - Simple “decoupling services”, content optimization, Catnap, higher level services
- Tapa will benefit from intrinsic security properties

XIA Project

- More information:
  - http://www.cs.cmu.edu/~xia
- XIA faculty
  - Peter Steenkiste, CS/ECE, Carnegie Mellon
  - Dave Andersen, David Eckhardt, Srinivasa Seshan, Hui Zhang, CS, Carnegie Mellon
  - Sara Kiesler, HCII, Carnegie Mellon
  - Jon Peha, Marvin Sirbu, EPP, Carnegie Mellon
  - Adrian Perrig, ETH/Carnegie Mellon
  - Aditya Akella, CS, University of Wisconsin
  - John Byers, CS, Boston University