eXpressive Internet Architecture: Overview

MobilityFirst Workshop

Organized as a part of:
Project: Engaging undergraduates in research that speaks their language
Sponsoring Agency: The Thurgood Marshall College Fund
Sponsoring program: Undergraduate Research to Retain and Graduate Students in STEAM

Shweta Jain, PH.D.
Assistant Professor and Doctoral Faculty
sjain@york.cuny.edu

York College of CUNY

Slides based on: XIA: An Architecture for an Evolvable and Trustworthy Internet, A. Anand et al. Technical Report CMU-CS-11-100, Department of Computer Science, Carnegie Mellon-University, February 2011.

Overview

Background and Overview of XIA

XIA is one of the 5 projects funded by the National Science Fund in 2010.[1] The project builds upon visions of content, service, host and user centric ideas[2].

1. Evolvable design for the future Internet that supports communication between/with any entity: content, service, host, user and anything else
2. Keeps the narrow waist model of the current Internet but adds "evolvability" to the waist
3. Intrinsically supports security by enabling verification of the communication principal through its name

[1] http://www.cs.cmu.edu/~./xia/

[2] See references: 11, 17, 30, 31 and 35 in the original paper

IP based narrow-waist: Good, Bad and a Compromise

1. Good: Narrow-waist model has provided room for innovations at upper and lower layers – reason for success of the Internet
2. Bad: Limits the network to use host addresses and does not provide a clean way to support new capabilities such as name based networking
3. Compromise: Application needs have led to the design of middle-boxes, such as firewall and proxy caches, to support some desired functionalities

IP based narrow-waist: Good, Bad and a Compromise

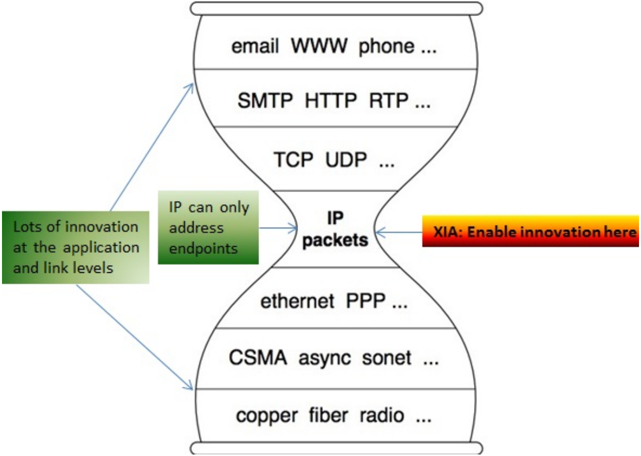


Figure: Narrow-waist model has been successful. XIA brings innovation to the narrow-waist to include X communicating principals:

Prior Motivating research

1. Content centric network design – such as the Van Jacobson's CCN motivates name based content retrieval
2. Service centric architecture – such as Jennifer Rexford's Scaffold that allows flexible object location
3. A user centric design – such as Bryan Ford's connectivity architecture for personal mobile devices
4. Host centric design – such as the IP based currently prevalent architecture

XIA Goals: The network should be able to natively support all these usage models and be evolvable..

Because we cannot predict/see the future.

What is missing in other competing projects?

1. Named Data Network: Designed only for content
2. Nebula: Designed from cloud based services
3. MobilityFirst: Supports named content and mobility
4. IP: Supports end-to-end host based communication

All except IP, support intrinsic security

The XIA advantage: The network should be able to natively support all these usage models and then some more......

Because we cannot predict/see the future.

The XIA Vision

Preserve the strength of today's existing architecture while...

1. Substantially improving security
2. Building the ability for evolution in network functionality to support the unforeseen future

How?

A *principal* centric approach that allows the communicating application to specify the type of communicating end-point:

- A piece of content addressed by name
- A service identified by a service ID which is capable of generating the desired content
- A host reachable at an address or a name
- Anything else..

Security is built-in through naming e.g., names and addresses are cryptographic hash of the content allowing content receivers to verify the content through the name

Example: IP

Get a web based content such as `www.yahoo.com`

1. Contact DNS service to obtain the host IP address
2. Send request for content to the host
3. Receive *html* page, parse and send requests for embedded static and dynamic objects
4. Content delivery networks such as `www.akamai.com` provide static content such as images through DNS redirection
5. Dynamic content such as Ads are generated by certain services that run on end-hosts

XIA Vision: The browser should interact with services using service IDs, hosts using host IDs and get content by name.

Example: NDN

Get a web based content such as `www.yahoo.com`

1. Send interest for the content in the network
2. Routers enter the interest in Pending Interest Tables (PIT), look up Forwarding Information Base (FIB) and forward the interest
3. Eventually when the content is found, it returns along the interfaces through which interest had propagated
4. NDN does not give a separate mechanism to reach a service or a host
5. What happens when routers do not have entries for the content in the FIB?

XIA Vision: The network should have a mechanism to redirect content requests to end-hosts if the content entry is not in the FIB

XIA Vision: Each principal i.e., service, host, content etc. should be handled in a way that the objective is achieved optimally rather than using another principal as a tunnel

Example: XIA

Get a web based content such as `www.yahoo.com`

1. Browser finds the **service identifier** for `www.yahoo.com` using web search, bookmarks, secure DNS etc.
2. The browser interacts with a *service* that provides `www.yahoo.com` by sending a packet whose **destination ID is the service identifier** rather than host identifier. Thus, the network can find the nearest service provider instead of going to a particular host
3. The service makes up a response containing the list of content IDs of all the chunks of data that make up the web site.
4. The browser sends a sequence of requests with content IDs as destination addresses to retrieve content from the network

Example: XIA - Security

Get a web based content such as `www.yahoo.com`

- The **service identifier** is a hash of a public key to verify authenticity of the service provider
- The service response is addressed to the requester's secure host identifier so that service goes back to the host that requested it.
- The service response itself is digitally signed for verification that the information is authentic
- Each chunk of content is verifiable by matching the hash of the content with its name/ID so the content is authentic regardless of where it came from.

What happens when a router does not know how to reach a piece of content?

Each content ID has a "fallback" option which is the ID of the service that originally provided the content ID. The service is supposed to know exactly where the content is located (end host)

XIA Design Requirements

1. Users and applications must be able to *eXpress* their intent:
   - Supports evolvability by making scope for adding any type of intent (principal)
   - Does not require all routers to implement/understand all intents.
   - Communicating parties must be able to specify secondary intents/alternate actions if routers cannot operate oon the primary intent
2. Principal types must be able to evolve
   - It should be possible to add new intents/principals
   - Modification of actions taken in existing principals should be supported
3. Principal identifies should be intrinsically secure
   - Communicating parties should be able to validate that it is interacting with a valid/correct principal
   - Example: Validate hosts using secure host ID and content through content hash

XIA Data Plane

Two components that achieve the XIA vision and implement the design requirements:

1. **eXpressive Internet Protocol:** The internetwork layer protocol that defines the common addressing, header formats and per-hop processing of all principals
2. **Principal type-specific support:** A customizable component that specifies treatment for each specific principal type e.g., host, content, service

Applications can also use multiple principal types through well-defined, principal specific interfaces (API).

eXpressive Internet Protocol

Basic building block that specifies common headers, packet formats and processing:

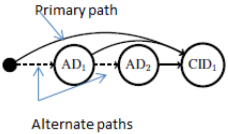
1. **XIP Addresses:** Structured as directed acyclic graphs (DAG) that support intrinsic security and scalable global routing
2. **XIP header formats:** Traditional fields such as version, length, TTL etc and variable length DAG fields
3. **Per-hop forwarding:** Source specific procession, next destination lookup and principal specific forwarding mechanism

XIP Addresses: DAG

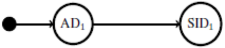
Properties of the XIP Address

- Each node is an XID (of any type), each edge is the next hop along a path
- Routing begins at the entry node that does not have an XID and ends at any node with out-degree zero
- The address is a single connected component
- Outgoing edges are tried in the order they are listed
- Node out-degree has an upper limit to facilitate fast processing at routers

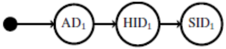
XIP DAG Address: Examples



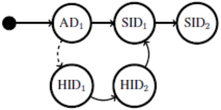
(a) General address format



(b) Reaching a service through an intermediate network



(c) Reaching a service through a host



(d) Fallbacks options in the address

Figure: XIP Address format: Directed Acyclic Graphs *Picture source:*[3]

[3] XIA: An Architecture for an Evolvable and Trustworthy Internet, A. Anand

XIP Headers

Two components in the header:

- **Traditional fields:** Single byte fields for header version, next header, payload lengths, hop limit (TTL)
- **Variable-length DAG fields:**
  - Single byte DAG fields (NS and ND) specify lengths of the source and destination DAG Addresses
  - LN points at the last visited node in the DAG
- The list of nodes in the DAG

Ver	NxtHdr	PayLen	HopLimit	ND	NS	LN

0:	XidType	ID	P[N]
...	...	...	...
ND-1:	XidType	ID	P[N]
0:	XidType	ID	P[N]
...	...	...	...
NS-1:	XidType	ID	P[N]

Figure: XIP packet header *Picture source:*[4]

[4] XIA: An Architecture for an Evolvable and Trustworthy Internet, A. Anand

Details of the DAG fields

DAGs are stored in topological order (order in which they will be visited)

- Each node in the DAG is an element of size 28 bytes
- First 4 bytes specify the principal type (XidType)
- 20 bytes are used for the principal's ID
- Four 1 byte pointers are the outgoing edges of the node, limiting the out-degree to 4

XIP Routers: Per-hop Forwarding

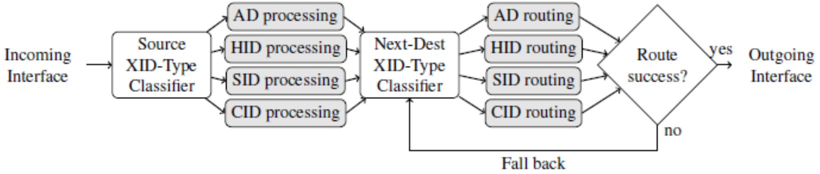


Figure: Simplified diagram of an XIP router *Picture source:*[5]

Steps in packet forwarding

1. When a packet arrives, the router first invokes the specific processing for the source-XID type
2. The Next-Dest classifier determines the appropriate forwarding actions, if none found, the router returns an error
3. The router invokes the principal specific processing module for the next destination before sending the packet forward

[5] XIA: An Architecture for an Evolvable and Trustworthy Internet, A. Anand

Web Browsing in XIA

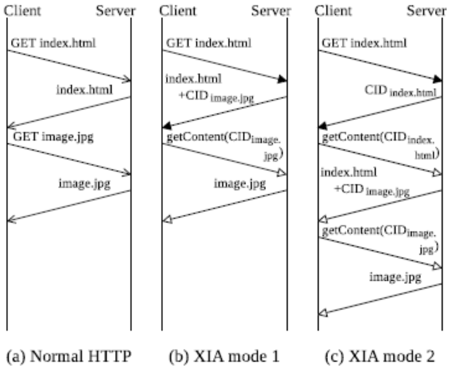


Figure: Timing diagrams of today's HTTP vs XIA with and without caching
Picture source:[6]

[6] XIA: An Architecture for an Evolvable and Trustworthy Internet, A. Anand et al.

Steps for retrieving web content in XIA

1. Send a request with a Service ID
2. Service returns either the html page with CID of remaining content or CID for the html page
3. Send *getContent* requests containing CIDs to retrieve each content

Comments or Questions?

Homework

Read the paper that was summarized in this slide

Blog about what you like, dislike or think about the paper and this presentation

Resources

- XIA website http://www.cs.cmu.edu/~./xia/
- The XIA Publications http://www.cs.cmu.edu/~./xia/publications/publications.html
- The XIA Presentations http://www.cs.cmu.edu/~./xia/Event-News/Talks.html
- The XIA Code http://www.cs.cmu.edu/~./xia/prototype.html