XIA: An Architecture for a Trustworthy and Evolvable Internet

Peter Steenkiste
Dave Andersen, David Eckhardt, Sara Kiesler, Jon Peha,
Adrian Perrig, Srini Seshan, Marvin Sirbu, Hui Zhang
Carnegie Mellon University

Aditya Akella, University of Wisconsin

John Byers, Boston University

Advanced Networking Symposium
UMD, March 23, 2012

Carnegie Mellon

BOSTON UNIVERSITY

THE UNIVERSITY of WISCONSIN MADISON

Outline

- Background
- XIA principles
- XIA architecture
- Building XIA
- Conclusion

NSF Future Internet Architecture

- Fundamental changes to the Internet architecture
  - Avoid constraints imposed by current Internet
  - Long-term, multi-phase effort
- Four teams were selected in the second phase:
  - Named Internet Architecture: content centric networking - data is a (the) first class entity
  - Mobility First: mobility as the norm rather than the exception – generalizes delay tolerant networking
  - Nebula: Internet centered around cloud computing data centers that are well connected
  - eXpressive Internet Architecture: focus on trustworthiness, evolvability

XIA Vision

We envision a future Internet that:

- Is trustworthy
  - Security broadly defined is the biggest challenge
- Supports long-term evolution of usage models
  - Including host-host, content retrieval, services, …
- Supports long term technology evolution
  - Not just for link technologies, but also for storage and computing capabilities in the network and end-points
- Allows all actors to operate effectively
  - Despite differences in roles, goals and incentives

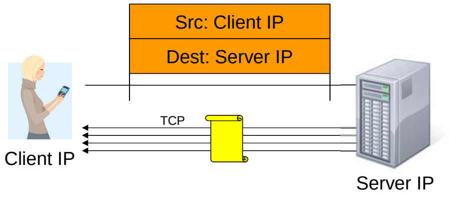
Predicting the Future is Hard!

– A lot of really smart people don't agree:

- – Named Data Networking: content centric networking - data is a first class entity
- – Mobility First: mobility as the norm rather than the exception – generalizes delay tolerant networking
- – Nebula: Internet centered around cloud computing data centers that are well connected

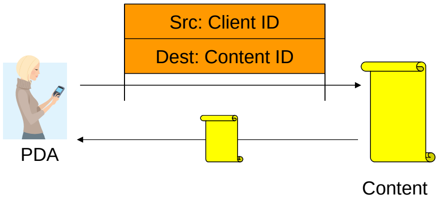
We love all of them!

Today's Internet



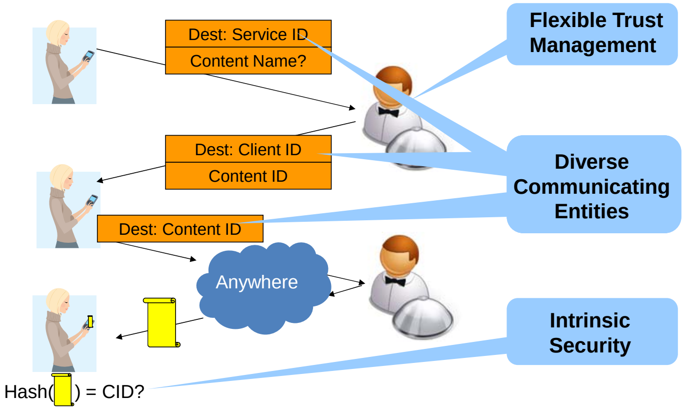
- Client retrieves document from a specific web server
  - But client mostly cares about correctness of content, timeliness
  - Specific server, file name, etc. are not of interest
- Transfer is between wrong principals
  - What if the server fails?
  - Optimizing transfer using local caches is hard
    - Need to use application-specific overlay or transparent proxy – bad!

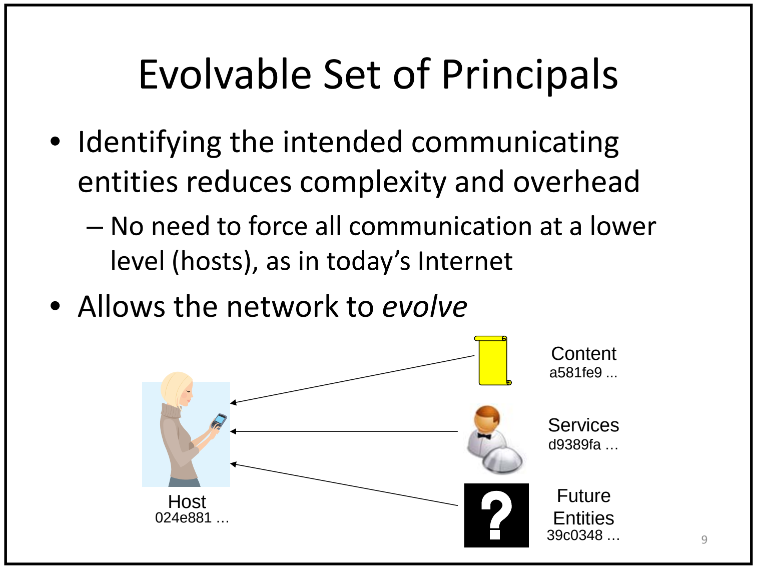
eXpressive Internet Architecture



- Client expresses communication intent for content explicitly
  - Network uses content identifier to retrieve content from appropriate location
- How does client know the content is correct?
  - Intrinsic security! Verify content using self-certifying id: hash(content) = content id
- How does source know it is talking to the right client?
  - Intrinsic security! Self-certifying host identifiers

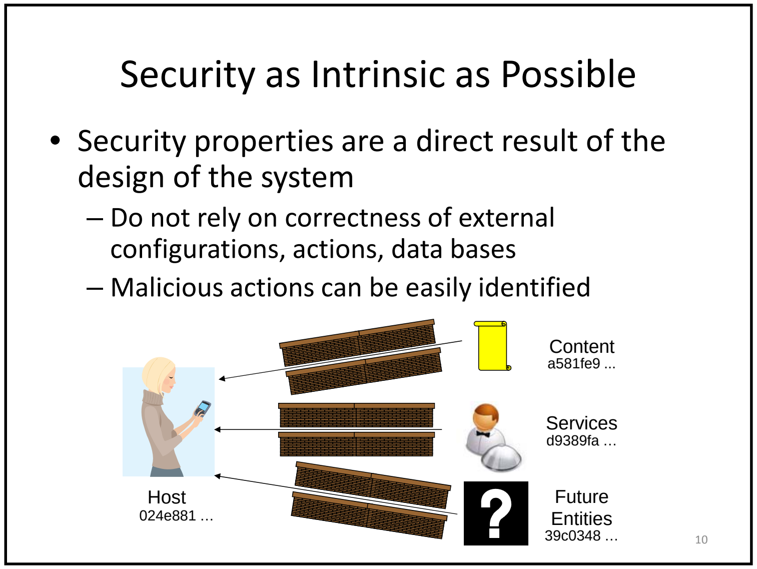
A Bit More Detail …





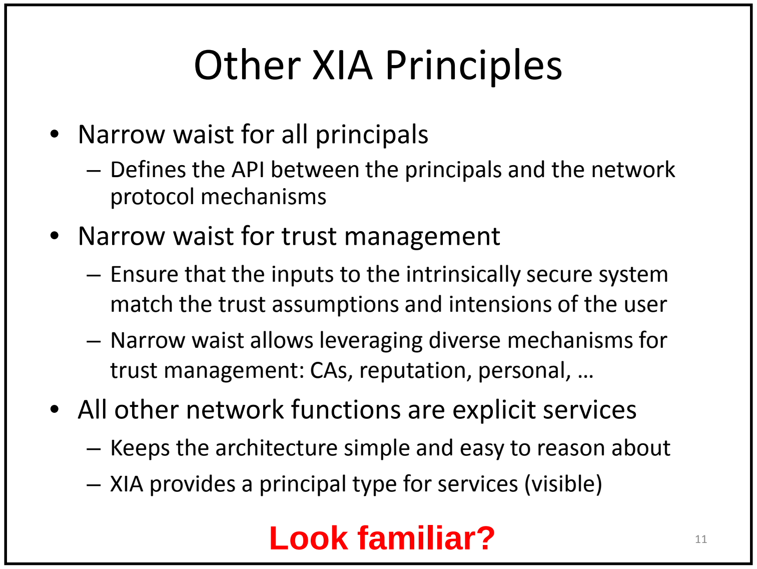
Evolvable Set of Principals

- Identifying the intended communicating entities reduces complexity and overhead
  - No need to force all communication at a lower level (hosts), as in today’s Internet
- Allows the network to *evolve*



Security as Intrinsic as Possible

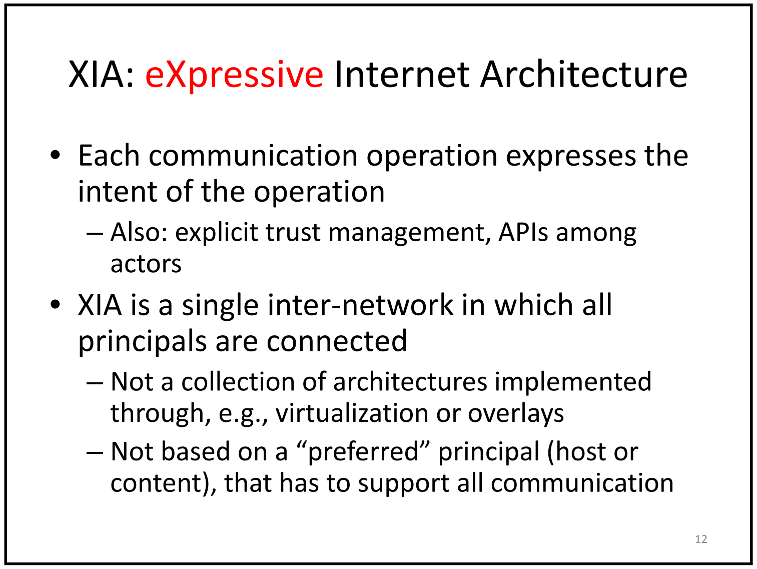
- Security properties are a direct result of the design of the system
  - Do not rely on correctness of external configurations, actions, data bases
  - Malicious actions can be easily identified



Other XIA Principles

- Narrow waist for all principals
  - Defines the API between the principals and the network protocol mechanisms
- Narrow waist for trust management
  - Ensure that the inputs to the intrinsically secure system match the trust assumptions and intensions of the user
  - Narrow waist allows leveraging diverse mechanisms for trust management: CAs, reputation, personal, …
- All other network functions are explicit services
  - Keeps the architecture simple and easy to reason about
  - XIA provides a principal type for services (visible)

Look familiar?



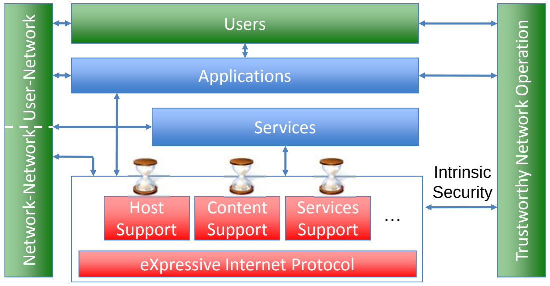
XIA: eXpressive Internet Architecture

- Each communication operation expresses the intent of the operation
  - Also: explicit trust management, APIs among actors
- XIA is a single inter-network in which all principals are connected
  - Not a collection of architectures implemented through, e.g., virtualization or overlays
  - Not based on a “preferred” principal (host or content), that has to support all communication

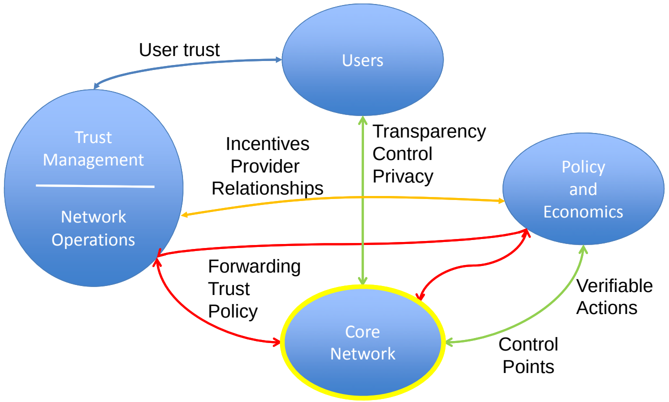
What Applications Does XIA Support?

- Since XIA supports host-based communication, today's applications continue to work
  - Will benefit from the intrinsic security properties
- New applications can express the right principal
  - Can also specify other principals (host based) as fallbacks
  - Content-centric applications
  - Explicit reliance on network services
  - Mobile users
  - As yet unknown usage models

XIA Components and Interactions

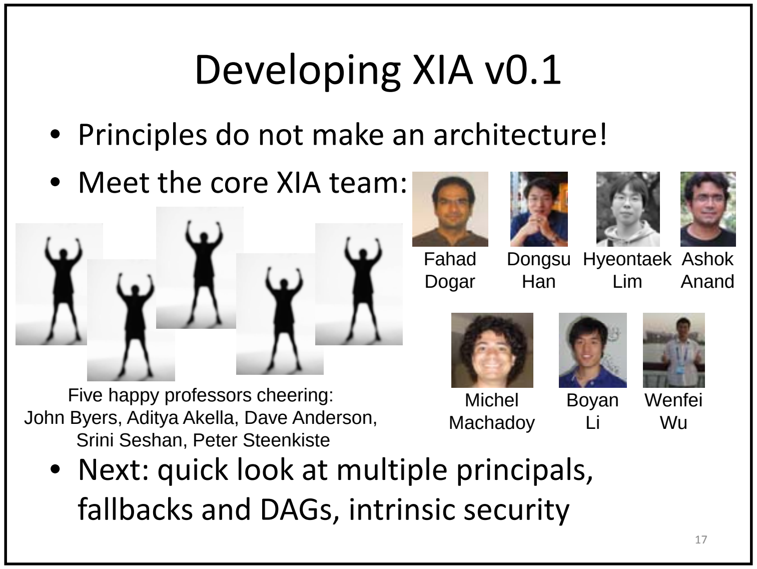

Network-Network | User-Network
Users
Applications
Services
Host Support
Content Support
Services Support
…
eXpressive Internet Protocol
Intrinsic Security
Trustworthy Network Operation

How about the Real World?



Outline

- Background
- XIA principles
- XIA architecture
  - Multiple principals
  - DAG-based addressing
  - Intrinsic security
- Building XIA
- Conclusion



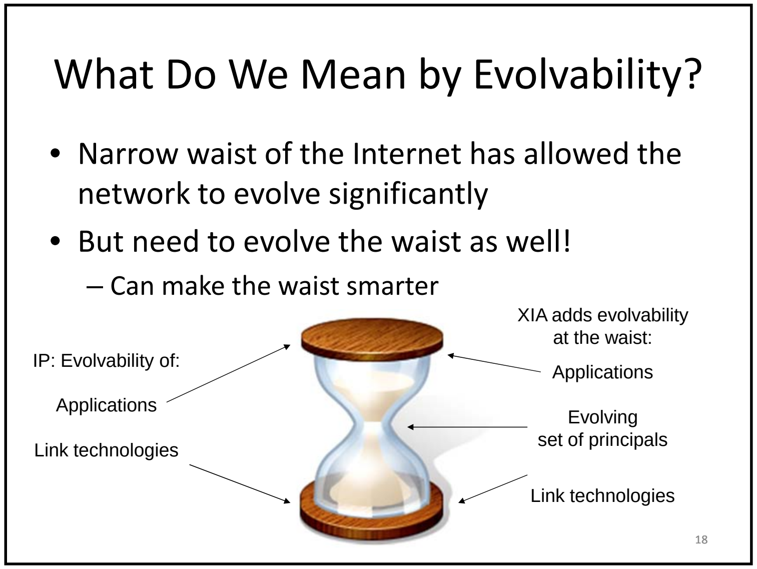
Developing XIA v0.1

- Principles do not make an architecture!
- Meet the core XIA team:

Five happy professors cheering:
John Byers, Aditya Akella, Dave Anderson,
Srini Seshan, Peter Steenkiste

- Next: quick look at multiple principals, fallbacks and DAGs, intrinsic security

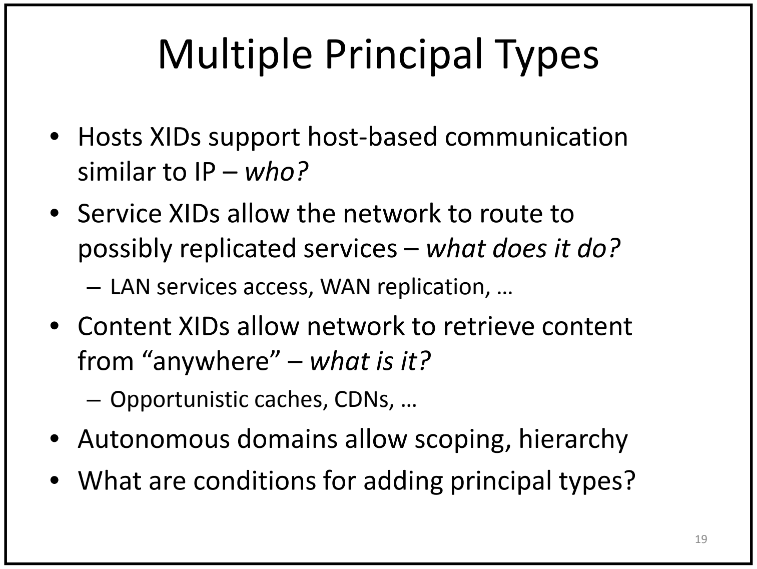
17



What Do We Mean by Evolvability?

- Narrow waist of the Internet has allowed the network to evolve significantly
- But need to evolve the waist as well!
  - Can make the waist smarter

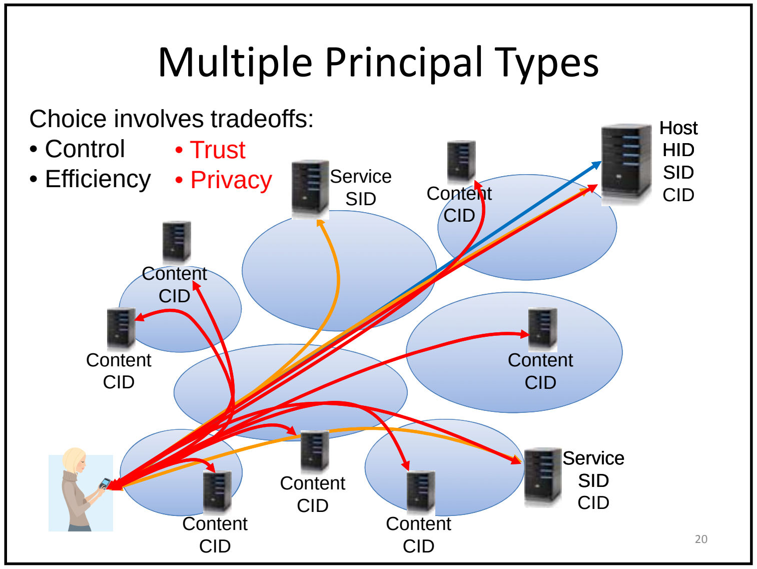
18



Multiple Principal Types

- Hosts XIDs support host-based communication similar to IP – *who?*
- Service XIDs allow the network to route to possibly replicated services – *what does it do?*
  - LAN services access, WAN replication, …
- Content XIDs allow network to retrieve content from “anywhere” – *what is it?*
  - Opportunistic caches, CDNs, …
- Autonomous domains allow scoping, hierarchy
- What are conditions for adding principal types?

19



Multiple Principal Types

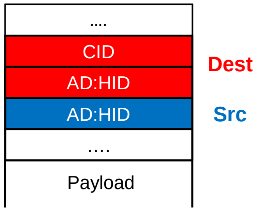
Choice involves tradeoffs:

- Control
- Efficiency
- Trust
- Privacy

20

Supporting Evolvability

- 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 is to provide an *intent* and *fallback* address
  - Intent address allows in-network optimizations based on user intent
  - Fallback address is guaranteed to be reachable



....

CID

AD:HID

AD:HID

....

Payload

Dest

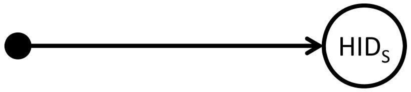
Src

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

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 HID_S



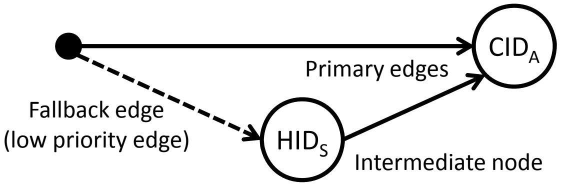
HID_S

Dummy source:
special node indicating
packet sender

Intent:
final destination of packet
with no outgoing edges

Support for Fallbacks with DAG

- A node can have multiple outgoing edges



CID_A

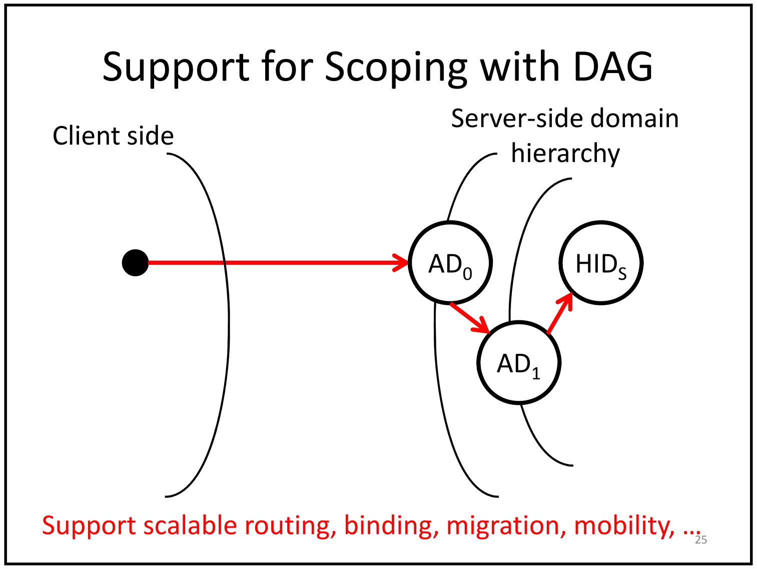
Primary edges

Fallback edge
(low priority edge)

HID_S

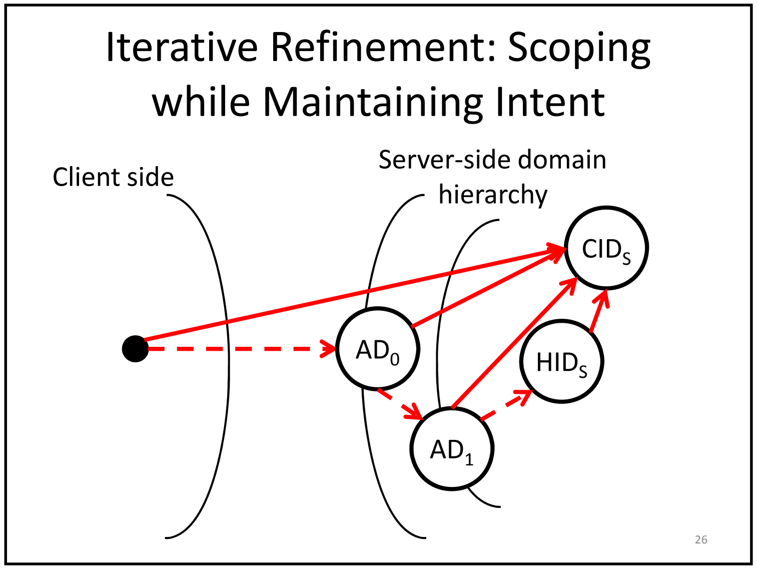
Intermediate node

- Outgoing edges have priority among them
  - Forwarding to HID_S is attempted if forwarding to CID_A is not possible – Realization of fallbacks

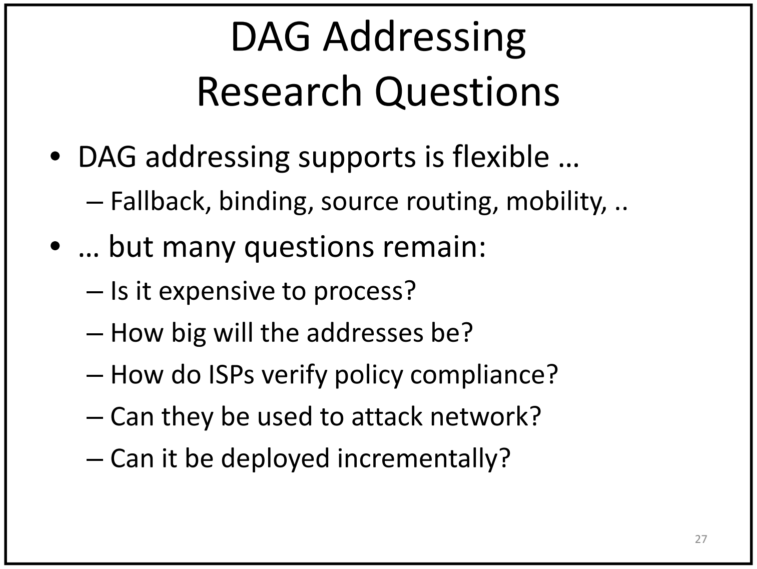


Support for Scoping with DAG

Support scalable routing, binding, migration, mobility, …

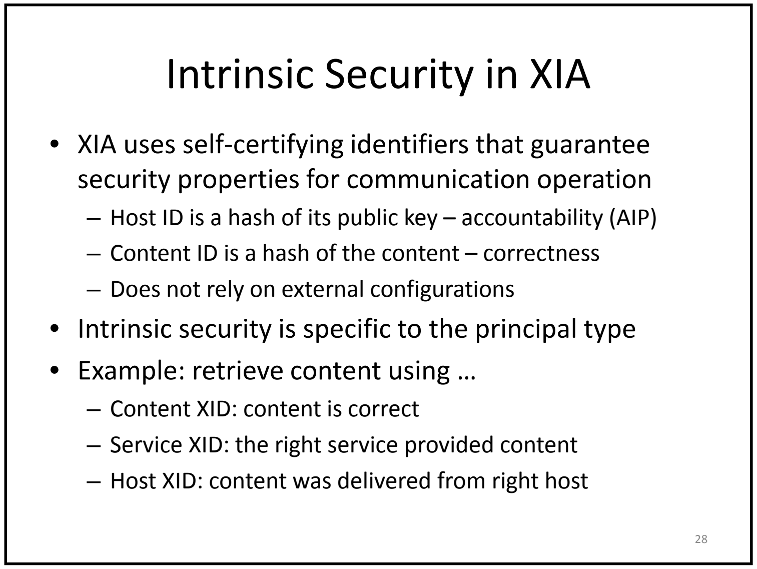


Iterative Refinement: Scoping while Maintaining Intent



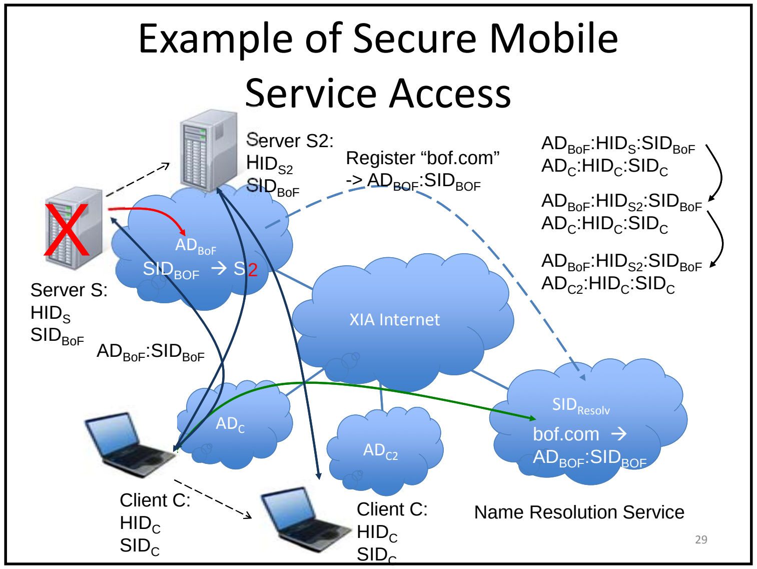
DAG Addressing Research Questions

- DAG addressing supports is flexible …
  - Fallback, binding, source routing, mobility, ..
- … but many questions remain:
  - Is it expensive to process?
  - How big will the addresses be?
  - How do ISPs verify policy compliance?
  - Can they be used to attack network?
  - Can it be deployed incrementally?



Intrinsic Security in XIA

- XIA uses self-certifying identifiers that guarantee security properties for communication operation
  - Host ID is a hash of its public key – accountability (AIP)
  - Content ID is a hash of the content – correctness
  - Does not rely on external configurations
- Intrinsic security is specific to the principal type
- Example: retrieve content using …
  - Content XID: content is correct
  - Service XID: the right service provided content
  - Host XID: content was delivered from right host



Example of Secure Mobile Service Access

Server S2:
HID_{S2}
SID_{BoF}

Register "bof.com"
-> AD_{BOF}:SID_{BOF}

AD_{BoF}:HID_S:SID_{BoF}
AD_C:HID_C:SID_C

AD_{BoF}:HID_{S2}:SID_{BoF}
AD_C:HID_C:SID_C

AD_{BoF}:HID_{S2}:SID_{BoF}
AD_{C2}:HID_C:SID_C

AD_{BoF}
SID_{BOF} → S2

Server S:
HID_S
SID_{BoF}

AD_{BoF}:SID_{BoF}

XIA Internet

AD_C

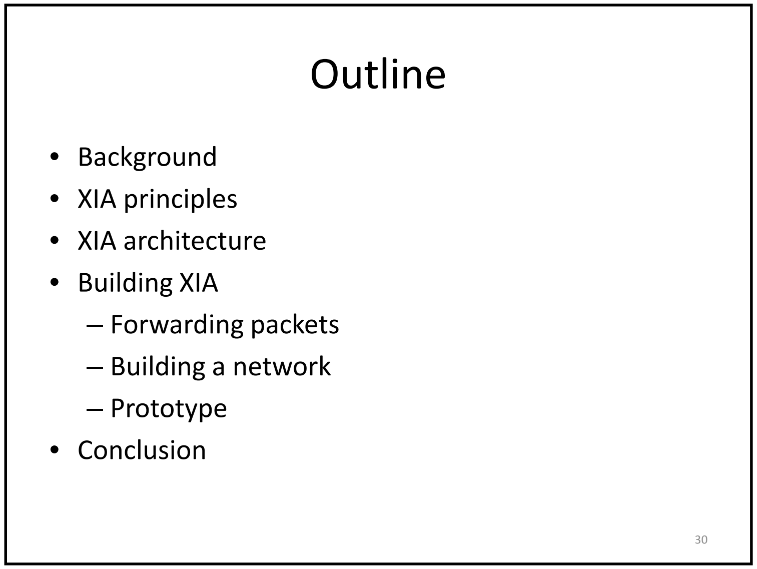
AD_{C2}

SID_{Resolv}
bof.com →
AD_{BOF}:SID_{BOF}

Client C:
HID_C
SID_C

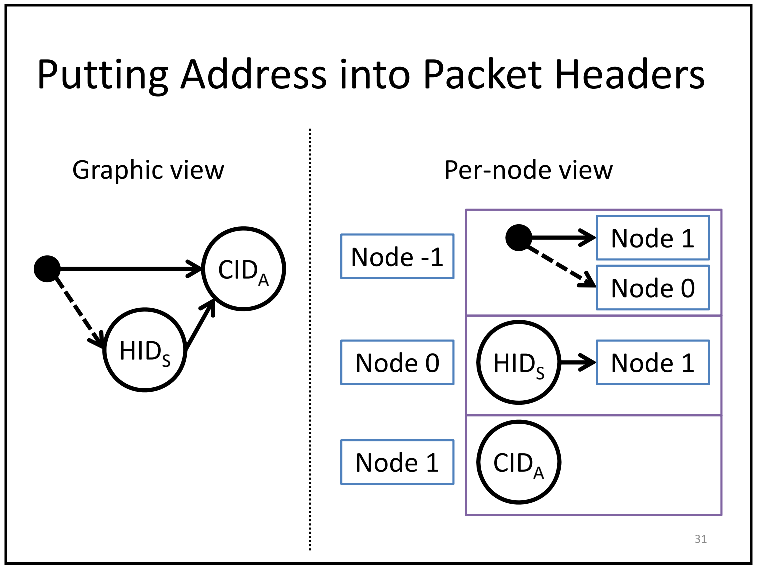
Client C:
HID_C
SID_C

Name Resolution Service



Outline

- Background
- XIA principles
- XIA architecture
- Building XIA
  - Forwarding packets
  - Building a network
  - Prototype
- Conclusion



Putting Address into Packet Headers

Graphic view

Per-node view

CID_A

HID_S

Node -1

Node 1

Node 0

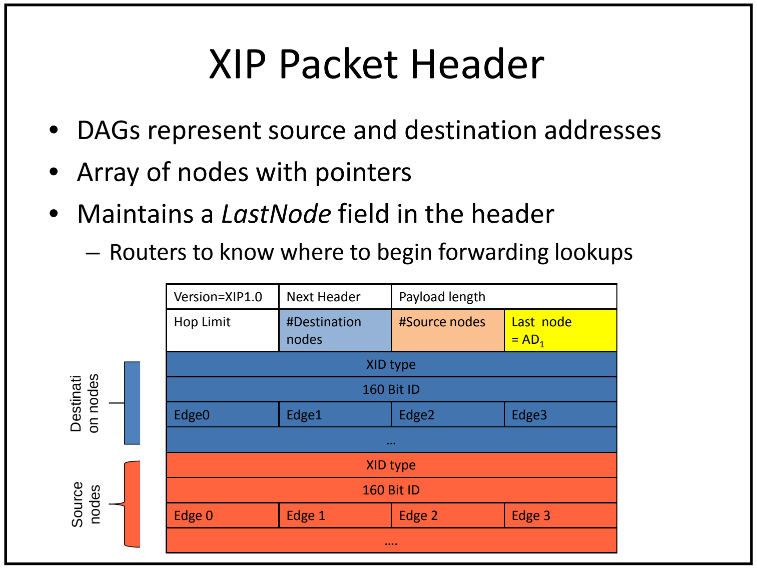
Node 0

HID_S

Node 1

Node 1

CID_A

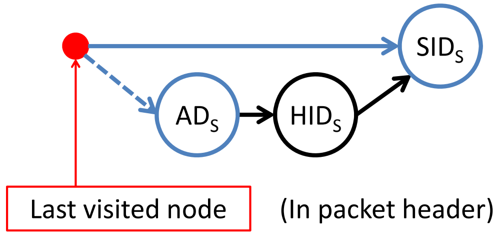


XIP Packet Header

- DAGs represent source and destination addresses
- Array of nodes with pointers
- Maintains a *LastNode* field in the header
  - Routers to know where to begin forwarding lookups

	Version=XIP1.0	Next Header	Payload length	
	Hop Limit	#Destination nodes	#Source nodes	Last node = AD_1
Destination nodes	XID type			
	160 Bit ID			
	Edge0	Edge1	Edge2	Edge3
	…			
Source nodes	XID type			
	160 Bit ID			
	Edge 0	Edge 1	Edge 2	Edge 3
	….			

Router's View on Packet Forwarding



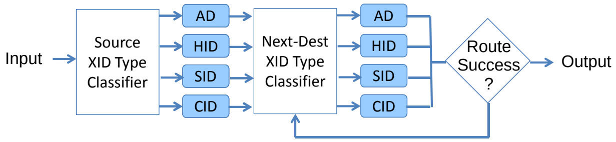
1. Forward to SID$_S$ if possible
2. Otherwise, forward to AD$_S$
   - If router is AD$_S$ itself,
     update last visited node to AD$_S$

XIP Forwarding

1. Find a routable edge from last visited node and select one with highest priority
2. If the next node refers to itself
   1. Advance last visited node pointer
   2. If there is no outgoing edge, the packet has arrived at the destination; send it to upper layer (transport)
   3. Otherwise, go to step 1
3. Otherwise, try to forward to next hop
   1. Forward if forwarding information is available
   2. Otherwise move to next node in the list

- No per-packet state in routers – high-speed forwarding

Packet Processing Pipeline



- Principle-independent processing defines how to interpret the DAG
  - The core XIA architecture
- Principle-dependent processing realizes forwarding semantics for each XID type
- Optimizations possible: fast path processing, packet level and intra-packet parallelism

Evaluation Setup

- Router
- Packet generator

Software:

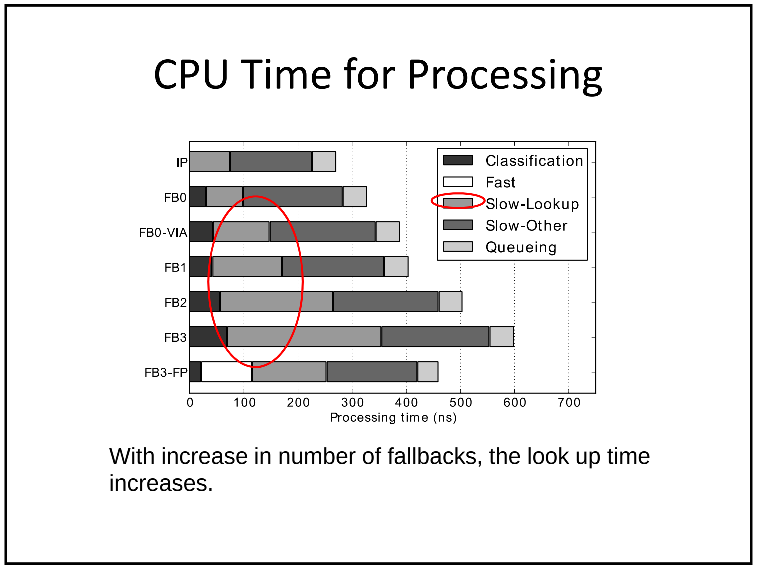
PacketShader I/O Engine

Click modular router – multithreaded(12 threads)

Hardware:

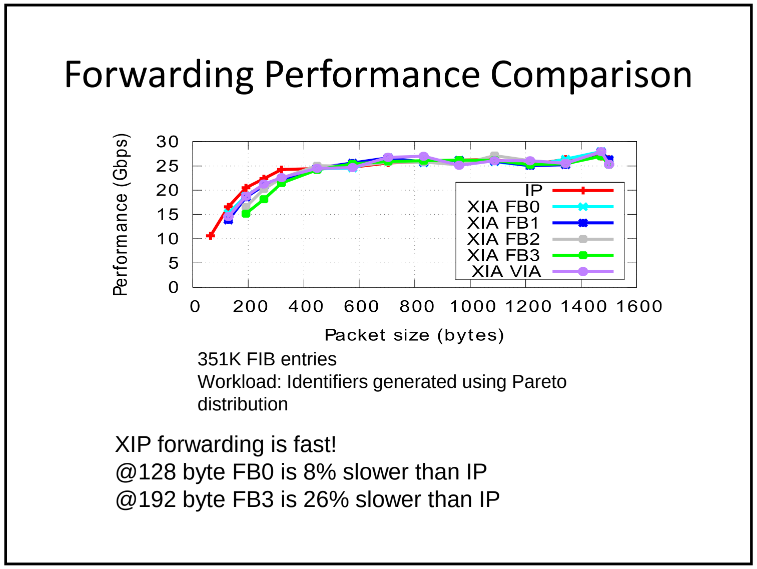
10Gbit NIC : 4 ports (multi-queue support)

2x 6 Core Intel Xeon @ 2.26GHz



CPU Time for Processing

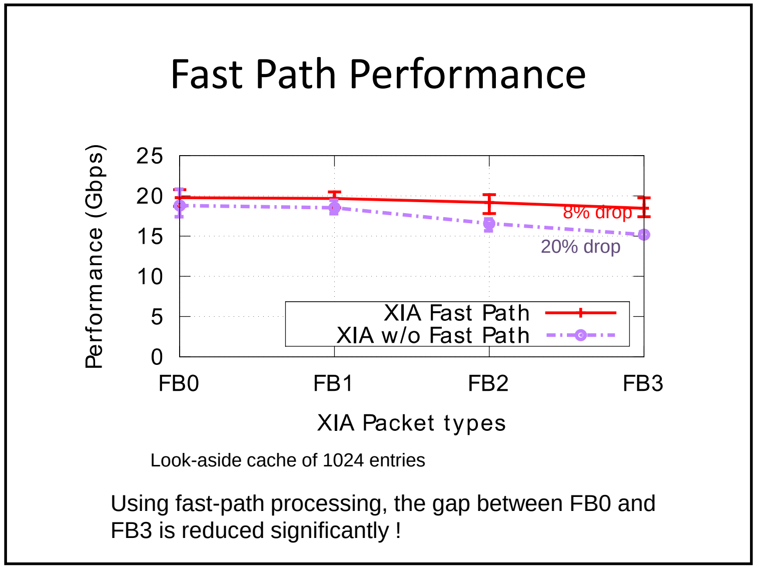
With increase in number of fallbacks, the look up time increases.



Forwarding Performance Comparison

351K FIB entries
Workload: Identifiers generated using Pareto distribution

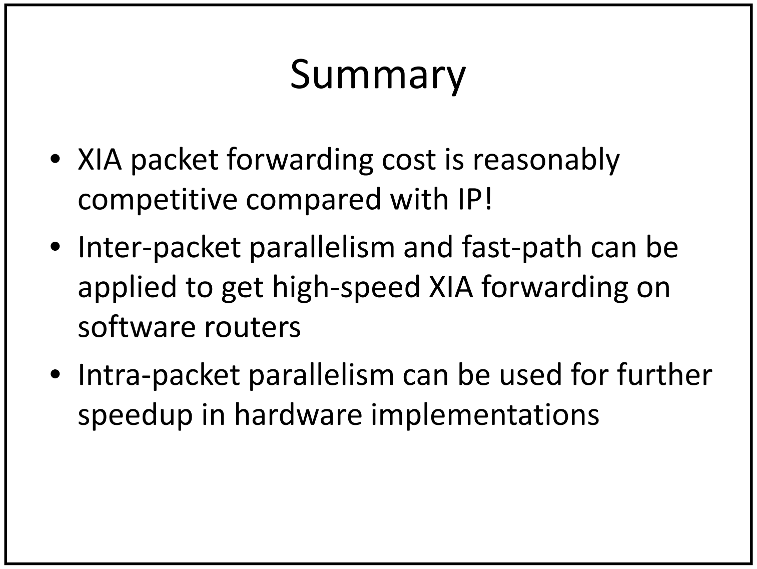
XIP forwarding is fast!
@128 byte FB0 is 8% slower than IP
@192 byte FB3 is 26% slower than IP



Fast Path Performance

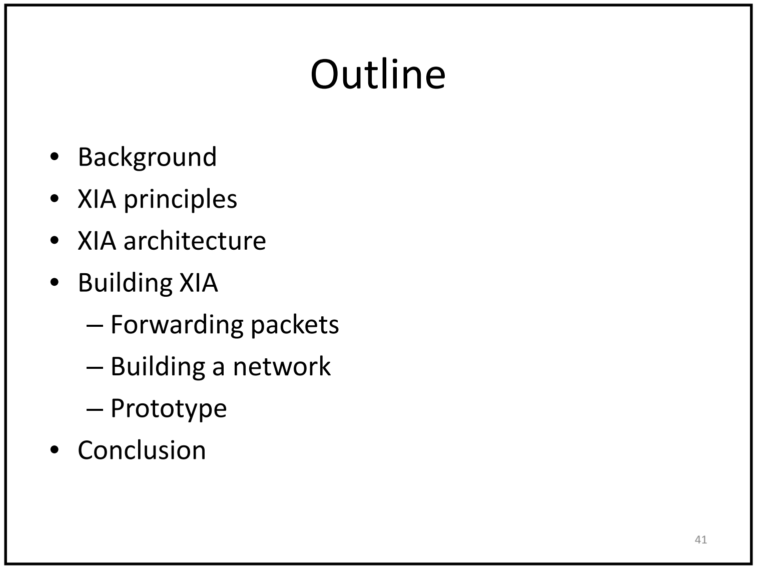
Look-aside cache of 1024 entries

Using fast-path processing, the gap between FB0 and FB3 is reduced significantly !



Summary

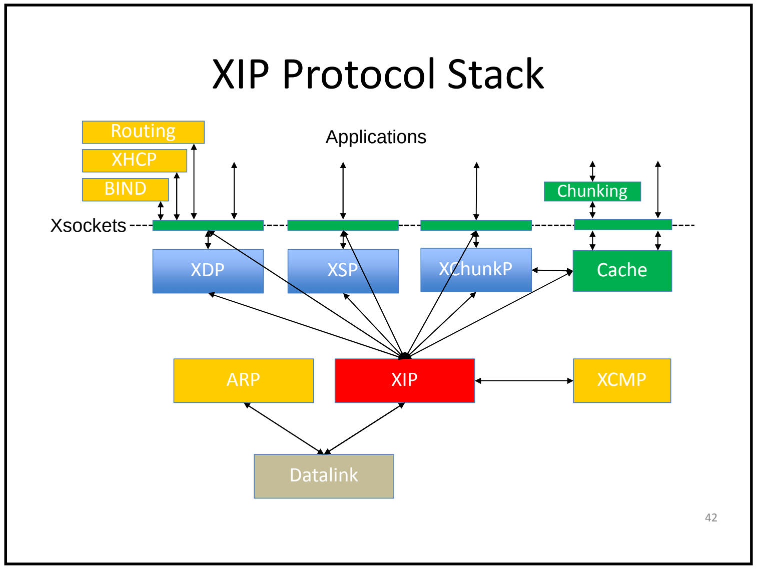
- XIA packet forwarding cost is reasonably competitive compared with IP!
- Inter-packet parallelism and fast-path can be applied to get high-speed XIA forwarding on software routers
- Intra-packet parallelism can be used for further speedup in hardware implementations



Outline

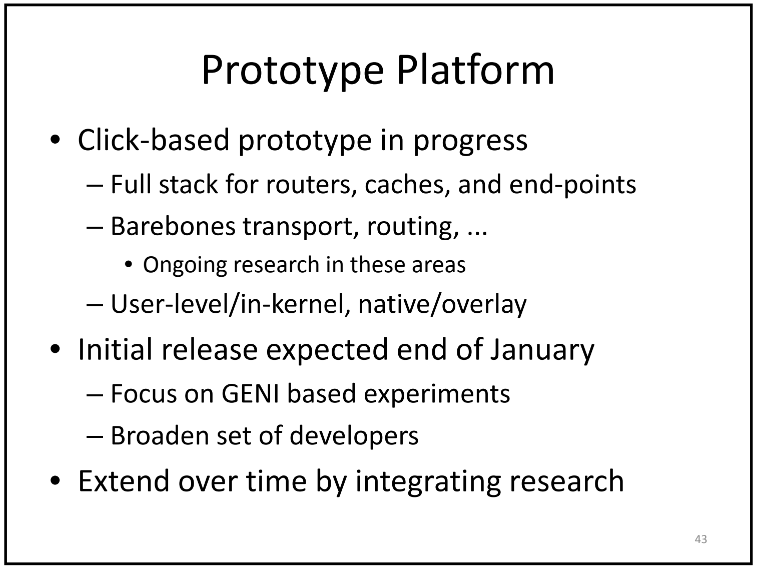
- Background
- XIA principles
- XIA architecture
- Building XIA
  - Forwarding packets
  - Building a network
  - Prototype
- Conclusion

41



XIP Protocol Stack

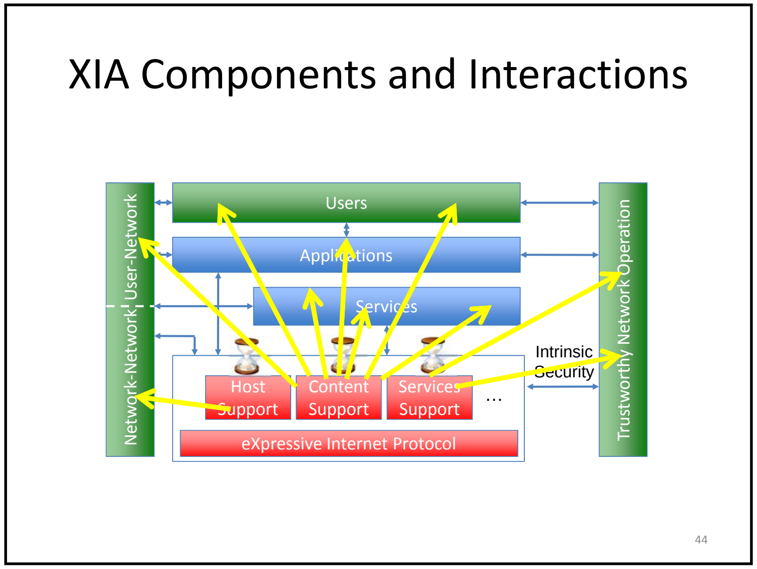
42



Prototype Platform

- Click-based prototype in progress
  - Full stack for routers, caches, and end-points
  - Barebones transport, routing, …
    - Ongoing research in these areas
  - User-level/in-kernel, native/overlay
- Initial release expected end of January
  - Focus on GENI based experiments
  - Broaden set of developers
- Extend over time by integrating research

43



XIA Components and Interactions

44

Some Ongoing Research Activities

- Transport protocols
  - Multiple principals, content, congestion control, ..
- Deployment of services and applications
  - Use of principal types, intrinsic security, …
- Network operations
  - Alternative routing strategies, e.g. Scion
  - Definition of DAGs, naming, diagnostics, …
- Evaluation and refinement of architecture
- User studies, policy/economics collaboration

Conclusion

- XIA supports evolution, expressiveness, and trustworthy operation.
  - Multiple principal types, flexible addressing, and intrinsic security
- But research has just started!
  - Transport protocols, applications, services, …
  - Trustworthy protocols that fully utilizes intrinsic security of XIA
- More information on

  http://www.cs.cmu.edu/~xia