Experience with the eXpressive Internet Architecture

Peter Steenkiste

Carnegie Mellon University

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

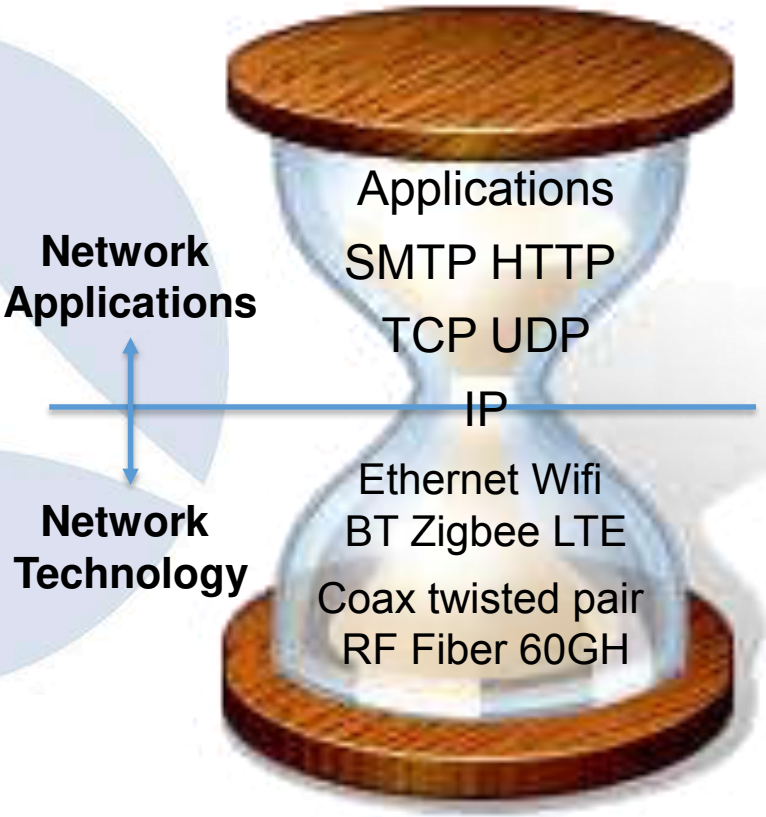
Aditya Akella, University of Wisconsin

John Byers, Boston University

Bruce Maggs, Duke

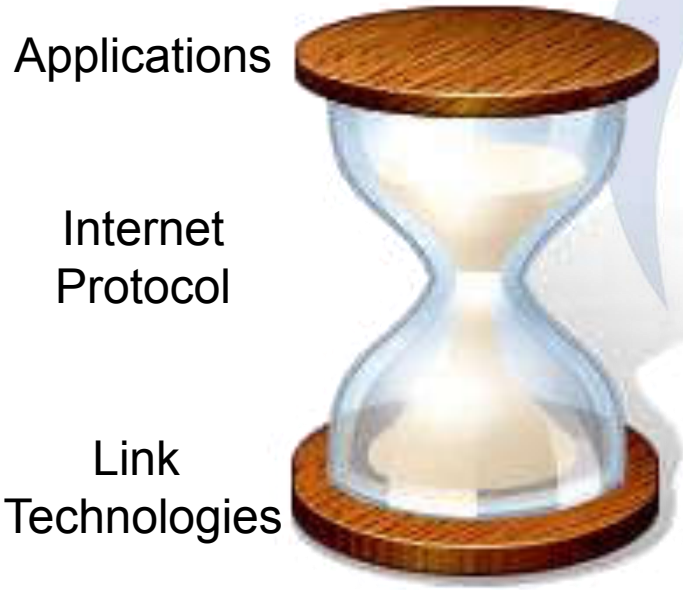
Role of the Internet Protocol (IP)

- IP is the shared language that is shared by all networks
  - IP is simple on purpose
- IP creates abstraction layer that hides underlying technology from network application software
  - Splits protocol stack
- Allows network technology and applications to evolve independently



“Narrow Waist” of the Internet Key to its Success

- Has allowed Internet to grow and evolve dramatically in the last 40 years
  - Adoption throughout society
    - E-commerce, social networks, cyber-physical, …
  - Transformation usage models
    - Host-based → content, services
  - Revolution in infrastructure
    - Kilobits/sec -> Terabits/sec
    - Copper -> fiber + wireless



But Narrow Waist Has Also Become an Obstacle

- Security is a huge problem – no support built into the network (IP)
  - DOS attacks, address spoofing, routing attacks, …

Evolvability

- New usage models add complexity, overhead
  - Content, service networking require a level of indirection
- Adding functionality in the network is difficult
  - IPv6, multicast, caching, "transparent" middleboxes, ..

Outline

- Overview of XIA
  - Motivation
  - Concepts
  - Implementations
- Some XIA use cases
  - What we got right
- Some lessons learned
  - What we got almost right

Funded by NSF through the FIA and FIA-NP programs

Three Simple Ideas

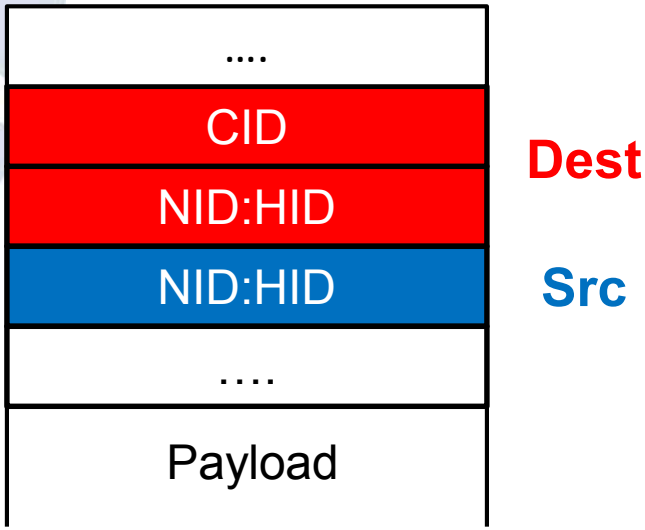
- Support multiple types of destinations
  - Not only hosts, but also content, services, etc.
  - Not having to force communication at a lower level (e.g., hosts) reduces complexity and overhead
- Flexible addressing gives network more options for successfully completing communication operations
  -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

Multiple Principal Types

- Associated with different *forwarding semantics*
  - Support heterogeneity in usage and deployment models
- **Hosts** XIDs support host-based communication – *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, …
- Set of principal types can evolve over time

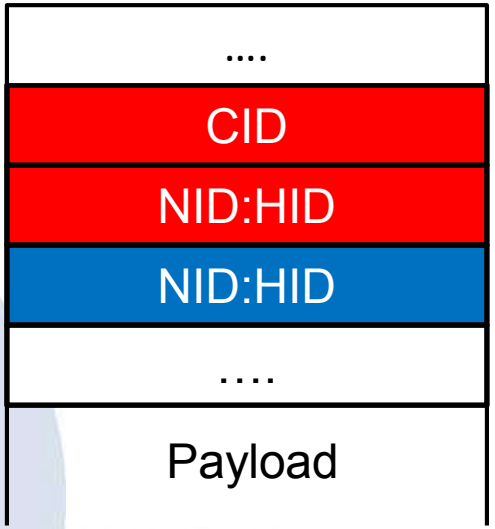
Supporting Evolvability

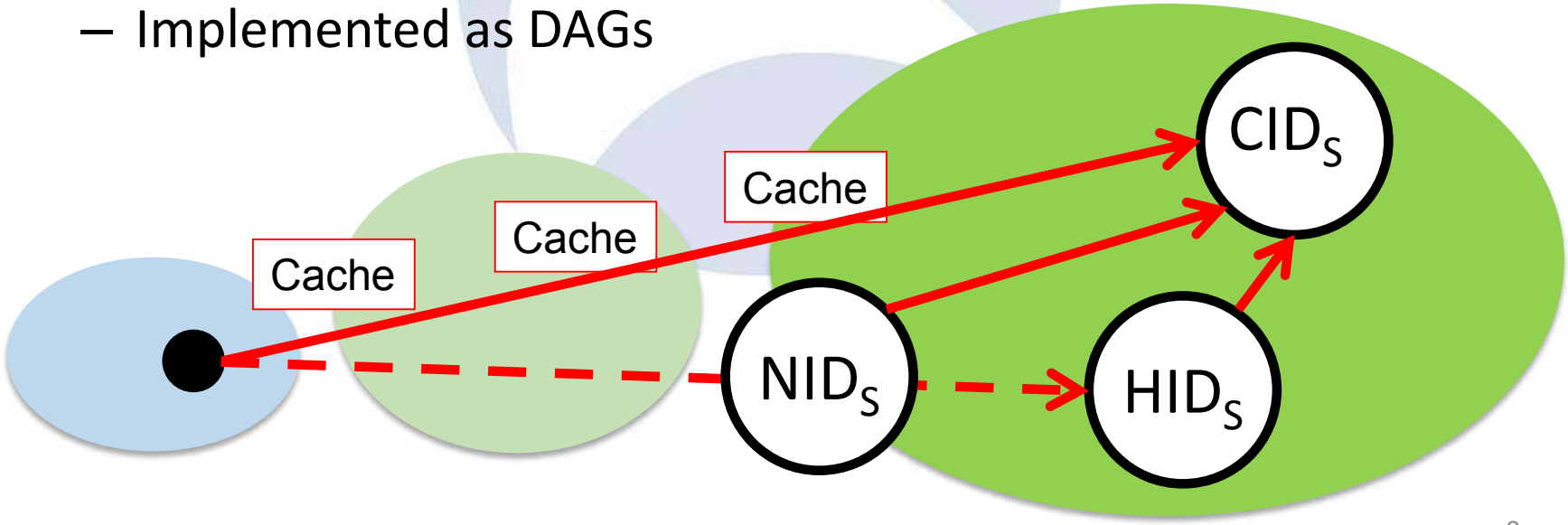
- Introduction of a new principal type will be incremental – no “flag day”!
  - Not all routers and ISPs will provide support from day one
- Creates chicken and egg problem - what comes first: network support or use in applications
- Solution is to provide an *intent* and *fallback* address
  - Intent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



Flexible Addressing: DAGs

- Combining intent and fallback address offers flexibility for network in completing request
  - Set of principal types can evolve
  - Also supports scoping
  - Implemented as DAGs



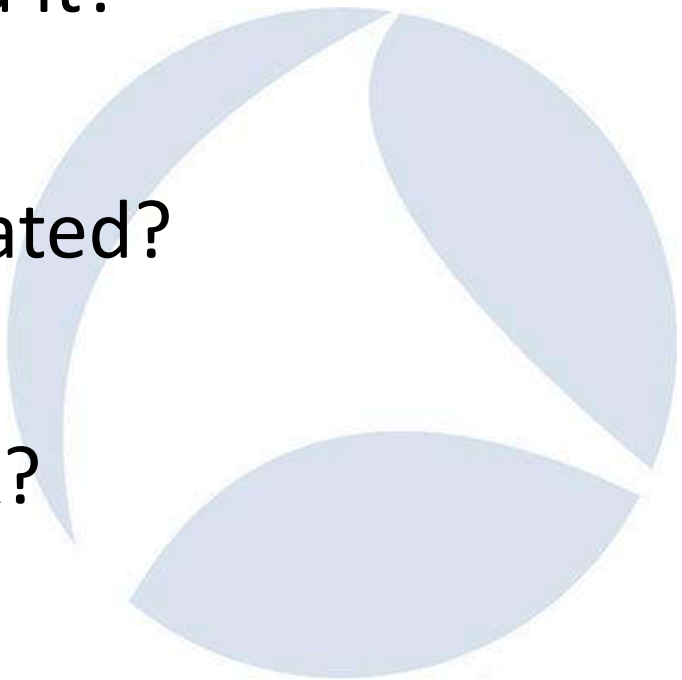


Intrinsic Security in XIA

- 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
- Useful for bootstrapping e-e security solutions
- Intrinsic security is specific to the principal type:
  - Content XID: content is correct
  - Service XID: the right service provided content
  - Host XID: content was delivered from right host

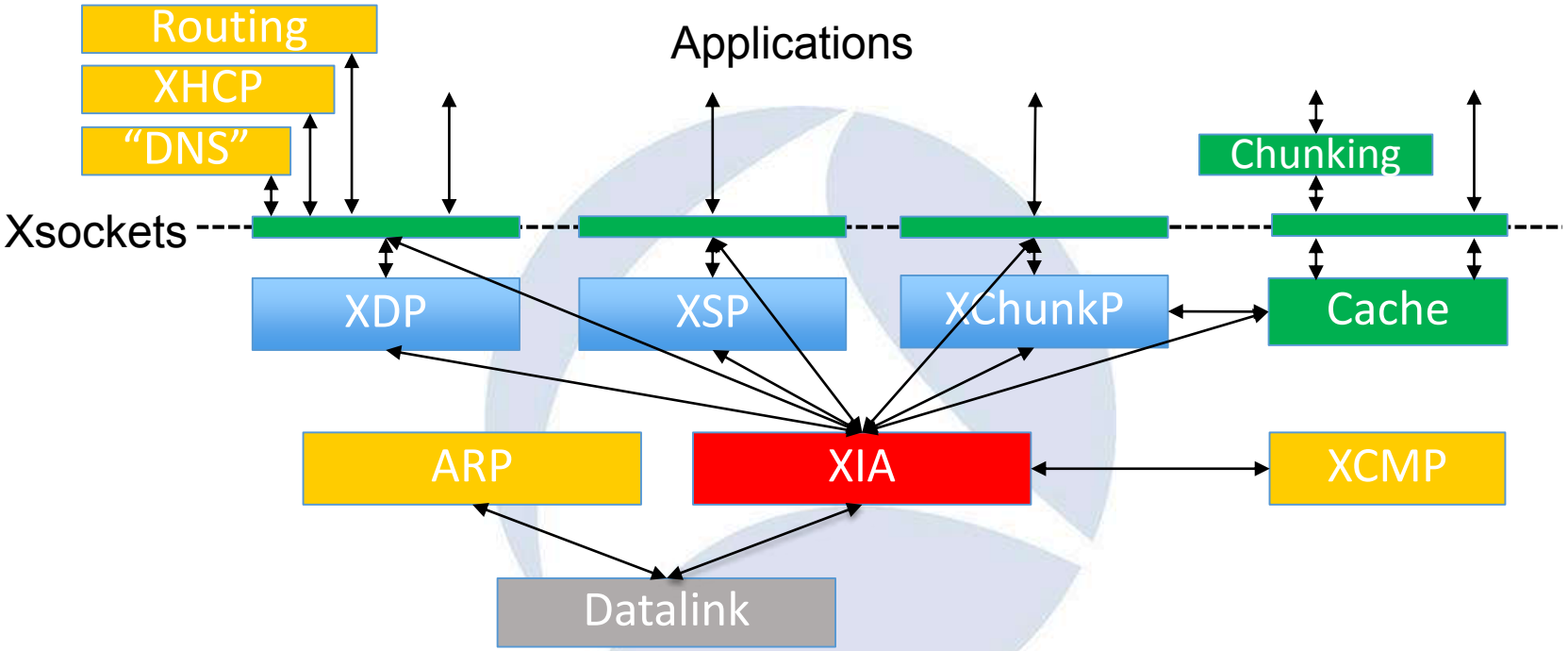
Nice, but …

- Can we build it?
- Is it complicated?
- Does it work?
- Is it a real network?



XIA Protocol Stack

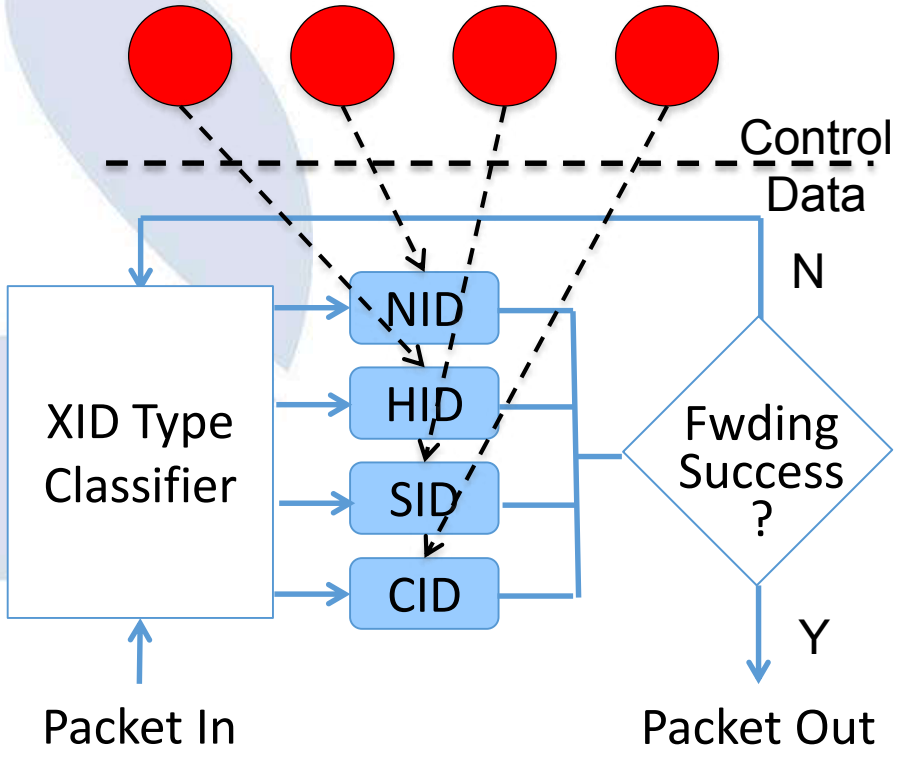
https://github.com/xia-project/



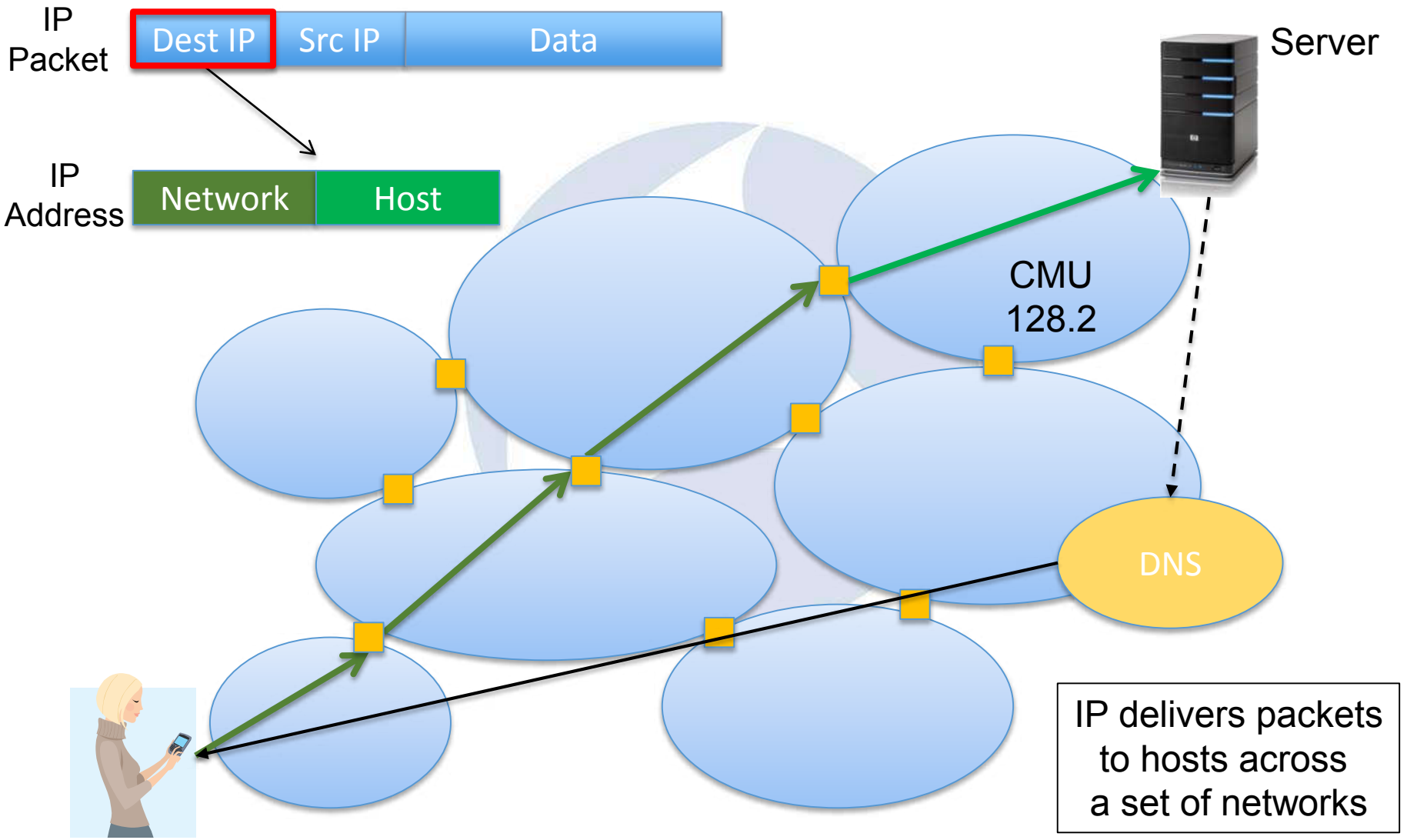
- First XIA Prototype released in May 2012
- Includes full XIA protocol stack, SID/CID support, utilities
  - But not quite perfect … more on this later
- Available as open source on github

But the Network Gets More Complicated!

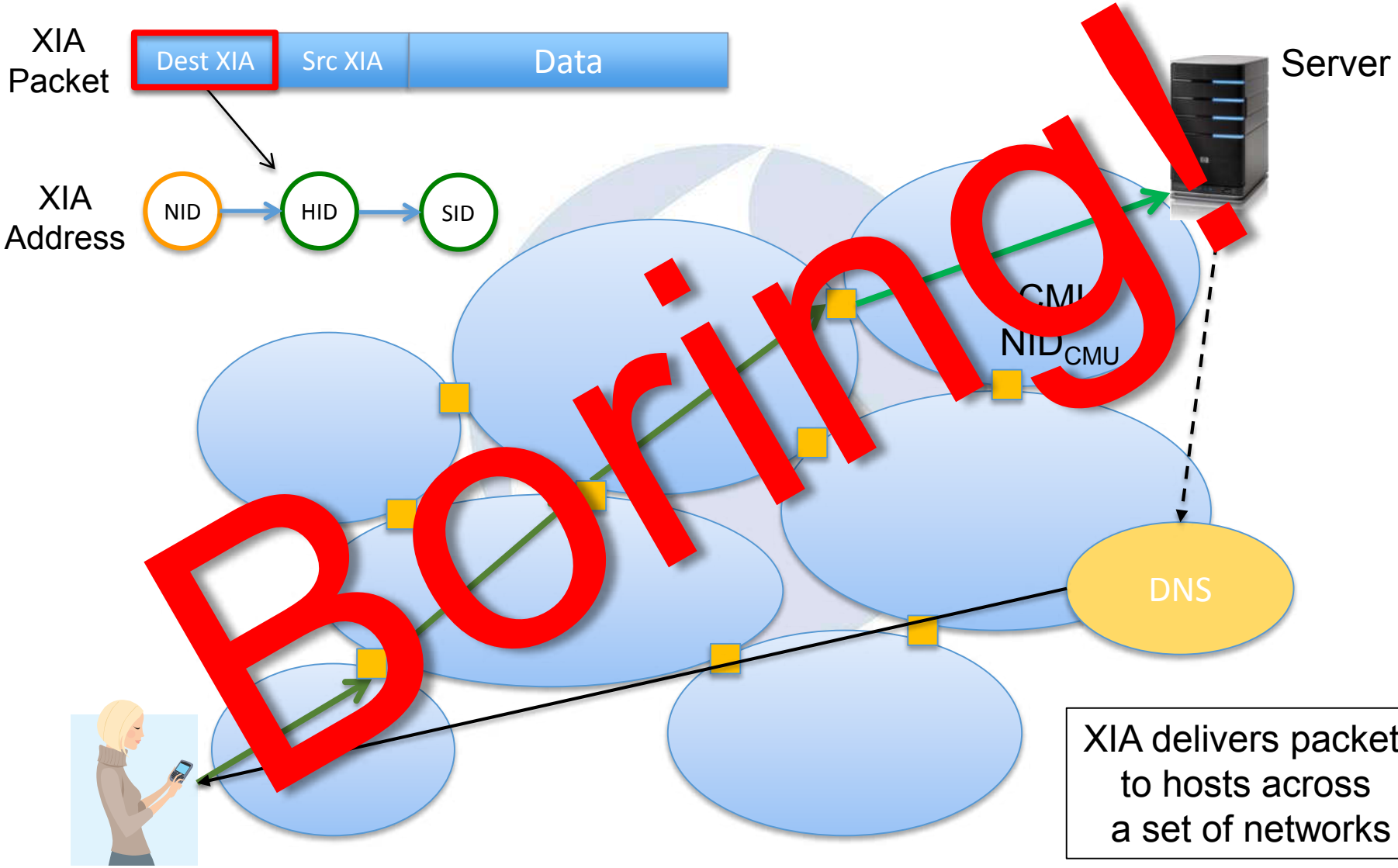
- Not really: per-XID additions are relatively small
- Forwarding engines are simple: often based on exact match
- Intrinsic security is not used during forwarding
- Biggest difference is in routing: that is where the “smarts” are!
- Good synergy with SDN!



Internet 101



Expressive Internet 101



XIA delivers packets to hosts across a set of networks

It is not a Real Network without Wireshark!

- Originally developed by Michel Machado (BU)
  - As part of his XIA implementation in Linux kernel
- Extended and adapted for the CMU XIA implementation
- Supports roughly the (new versions of) the protocols shown on previous slide

Capturing from ens33 ((ether proto 0xc0de) && !(ether dst ff:ff:ff:ff:ff:ff)) [Wireshark 2.3.0 (v2.3.0rc0-1112-gae987d5 from xia)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xdgram	222	XIA XDatagram Packet
2	0.002077692	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xdgram	349	XIA XDatagram Packet
3	0.082906751	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xstream	226	[SYN] Win=65535 Seq=1052094146 Ack=0 MSS=1500 WS=16 T
4	0.083572355	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xstream	226	[SYN, ACK] Win=65535 Seq=3821598699 Ack=1052094147 MS
5	0.083778297	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xstream	218	[ACK] Win=65535 Seq=1052094147 Ack=3821598700 TSval=1
6	0.084182056	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xstream	730	[ACK] Win=65535 Seq=1052094147 Ack=3821598700 TSval=1
7	0.084220002	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xstream	218	[ACK] Win=65535 Seq=3821598700 Ack=1052094147 TSval=4
8	0.086083283	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xstream	730	[ACK] Win=65535 Seq=3821598700 Ack=1052094659 TSval=4
9	0.087116288	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xstream	218	[FIN, ACK] Win=65535 Seq=1052094659 Ack=3821599212 TS
10	0.088146787	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xstream	218	[ACK] Win=65535 Seq=3821599212 Ack=1052094660 TSval=4
11	0.088379926	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xstream	218	[FIN, ACK] Win=65535 Seq=3821599212 Ack=1052094660 TS
12	0.088465681	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xstream	218	[ACK] Win=65535 Seq=1052094660 Ack=3821599213 TSval=1

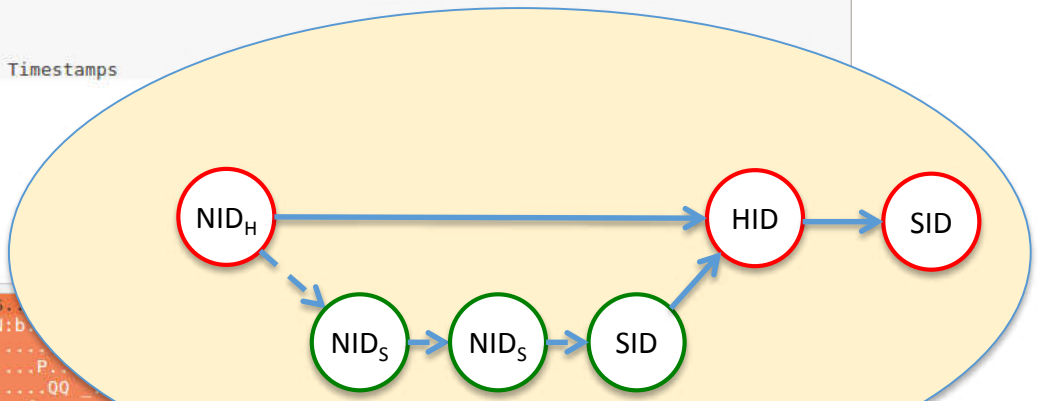
```
▸ Frame 3: 226 bytes on wire (1808 bits), 226 bytes captured (1808 bits) on interface 0
▾ Ethernet II, Src: 00:0c:29:79:bf:3c (00:0c:29:79:bf:3c), Dst: 00:0c:29:61:04:f1 (00:0c:29:61:04:f1)
  ▸ Destination: 00:0c:29:61:04:f1 (00:0c:29:61:04:f1)
  ▸ Source: 00:0c:29:79:bf:3c (00:0c:29:79:bf:3c)
    Type: eXpressive Internet Protocol (0xc0de)
▾ eXpressive Internet Protocol
    Version: 1
    Next Header: Xstream (0x03)
    Payload Length: 36 bytes
    Hop Limit: 249
    Number of Destination Nodes: 3
    Number of Source Nodes: 3
    Last Node: 0
  ▾ Destination DAG
      !ad-ecf128e16a87298ee7dcd5e4029f70be28de41c2-1***:
      hid-84edbe0dd1e5cfca2ee19d4146a2ed1ee64362c6-2***:
      sid-073bf1c6352f0a1cbece1bea144fc72a0777c158->0***
  ▾ Source DAG
      ad-ecf128e16a87298ee7dcd5e4029f70be28de41c2-1:
      hid-f6d79b8d32c40501fab6e0f8a3eee7a4984694e5-2:
      sid-e23aa78ddf520d328d72e3c6a295d4bc6d6a930e-0
▾ XIA XStream
    Next Header: Data (0x00)
    Header Length: 36 bytes (9)
  · Flags: 0x0002 (SYN)
    Sequence #: 1052094146
    Ack #: 0
    Window: 65535
  ▾ Options: (20 bytes), Maximum segment size, Window scale, Timestamps
    ▾ Maximum segment size: 1500 bytes
        Kind: Maximum Segment Size (2)
        Length: 4 bytes (1)
        MSS Value: 1500
    ▸ Window scale: 4 (multiply by 16)
    ▸ Timestamps: TSval 1060, TSecr 0
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xdgram	222	XIA XDatagram Packet
2	0.002955524	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xdgram	500	XIA XDatagram Packet
3	0.077297511	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xstream	310	[SYN] Win=65535 Seq=2840934383 Ack=0 MSS=1500 WS=16 T
4	0.077964214	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xstream	310	[SYN, ACK] Win=65535 Seq=730808597 Ack=2840934384 MSS
5	0.078115037	00:0c:29:79:bf:3c	00:0c:29:61:04:f1	Xstream	302	[ACK] Win=65535 Seq=2840934384 Ack=730808598 TSval=36
6	0.079087849	00:0c:29:61:04:f1	00:0c:29:79:bf:3c	Xstream	302	[ACK] Win=65535 Seq=730808598 Ack=2840934384 TSval=36
7	0.079504643	00:0c:29:79:bf:				
8	0.080802357	00:0c:29:61:04:				
9	0.081153154	00:0c:29:61:04:				
10	0.082269613	00:0c:29:79:bf:				
11	0.083113686	00:0c:29:61:04:				
12	0.083538865	00:0c:29:61:04:				
13	0.083628958	00:0c:29:79:bf:				

AD:d79a0f34002bf44e3a62ac06d1ffda509ce95b6f 4 1 -
AD:d79a0f34002bf44e3a62ac06d1ffda509ce95b6f 2 -
HID:01c9e61351515fc9aa53a1c093f73ce705037b1e 3 -
SID:8114ae4343b20a6fac42db3d0abf9da8d09d33d9 4 -
HID:01c9e61351515fc9aa53a1c093f73ce705037b1e 5 -
SID:374784ea077f5806cdae115997ca84e8f276be25

```
▸ Frame 3: 310 bytes on wire (2480 b
▸ Ethernet II, Src: 00:0c:29:79:bf:3
▾ eXpressive Internet Protocol
    Version: 1
    Next Header: Xstream (0x03)
    Payload Length: 36 bytes
    Hop Limit: 249
    Number of Destination Nodes: 6
    Number of Source Nodes: 3
    Last Node: 0
  ▾ Destination DAG
      [truncated]!ad-d79a0f34002bf44e3a62ac06d1ffda509ce95b6f-41**:
     ad-d79a0f34002bf44e3a62ac06d1ffda509ce95b6f-2***:
     hid-01c9e61351515fc9aa53a1c093f73ce705037b1e-3***:
     sid-8114ae4343b20a6fac42db3d0abf9da8d09d33d9-4***:
     hid-01c9e61351515fc9aa53
  ▸ Source DAG
▾ XIA XStream
    Next Header: Data (0x00)
    Header Length: 36 bytes (9)
  ▸ Flags: 0x0002 (SYN)
    Sequence #: 2840934383
    Ack #: 0
    Window: 65535
  ▸ Options: (20 bytes), Maximum segment size, Window scale, Timestamps
```

```
0010  00 24 f9 06 03 00 00 00  00 10 d7 9a 0f 34 00 2b   .$.
0020  f4 4e 3a 62 ac 06 d1 ff  da 50 9c e9 5b 6f 04 01   .N:b
0030  7f 7f 00 00 00 10 d7 9a  0f 34 00 2b f4 4e 3a 62   .....
0040  ac 06 d1 ff da 50 9c e9  5b 6f 02 7f 7f 7f 00 00   .....P..
0050  00 11 01 c9 e6 13 51 51  5f c9 aa 53 a1 c0 93 f7   ......QQ _..S....
0060  3c e7 05 03 7b 1e 03 7f  7f 7f 00 00 00 13 81 14   <...{... ........
0070  ae 43 43 b2 0a 6f ac 42  db 3d 0a bf 9d a8 d0 9d   .CC..o.B .=......
0080  33 d9 04 7f 7f 7f 00 00  00 11 01 c9 e6 13 51 51   3....... ......QQ
0090  5f c9 aa 53 a1 c0 93 f7  3c e7 05 03 7b 1e 05 7f   _..S.... <...{...
00a0  7f 7f 00 00 00 13 37 47  84 ea 07 7f 58 06 cd ae   ......7G ....X...
00b0  11 59 97 ca 84 e8 f2 76  be 25 80 7f 7f 7f 00 00   .Y.....v .%......
00c0  00 10 d7 9a 0f 34 00 2b  f4 4e 3a 62 ac 06 d1 ff   .....4.+ .N:b....
00d0  da 50 9c e9 5b 6f 01 7f  7f 7f 00 00 00 11 f6 d7   .P..[o.. ........
00e0  9b 8d 32 c4 05 01 fa b6  e0 f8 a3 ee e7 a4 98 46   ..2..... .......F
00f0  94 e5 02 7f 7f 7f 00 00  00 13 2b 6c f9 a1 11 69   ........ ..+l...i
0100  15 0d 72 5b 54 7b 9c ac  58 49 64 39 ae 1d 00 7f   ..r[T{.. XId9....
```



Outline

- Overview of XIA
- Some XIA use cases
  - Caching
  - Mobility
  - Incremental deployment
  - Network diversity
- Some lessons learned

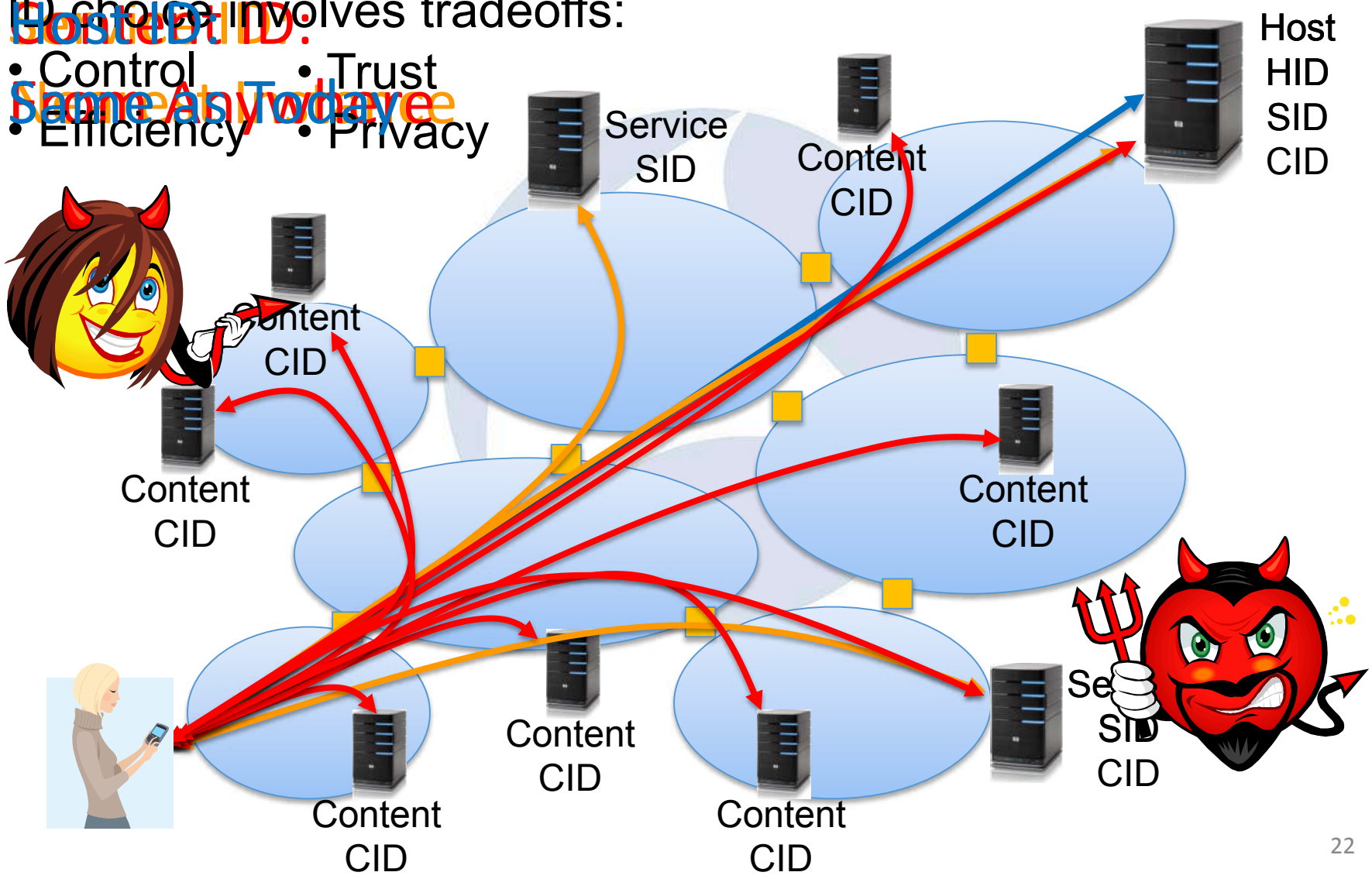
More Interesting Uses of XIA

	Destination Types	Flexible Addressing	Intrinsic Security
Evolvability: CID, SID, …	✓	✓	✓
Network level Caching	✓	✓	✓
Incremental Deployment	✓	✓	
Mobility	✓	✓	✓
Service Anycast	✓	✓	✓
Extreme Evolvability	✓	✓	✓
Path-based Fwding: Scion	✓	✓	✓

XIA Content Retrieval

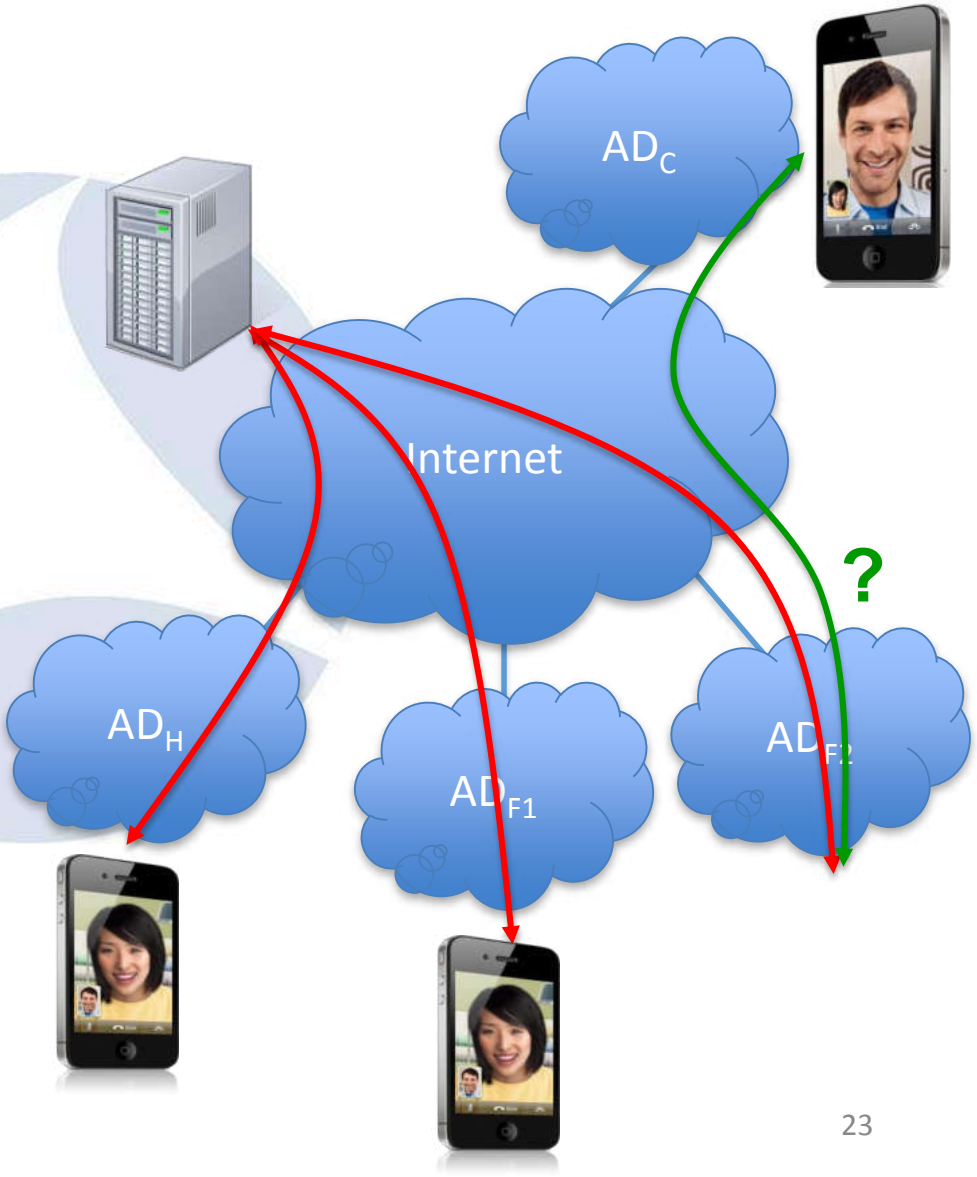
ID choice involves tradeoffs:

- Control
- Trust
- Efficiency
- Privacy

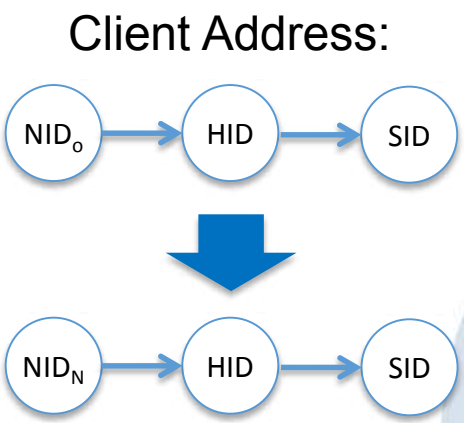


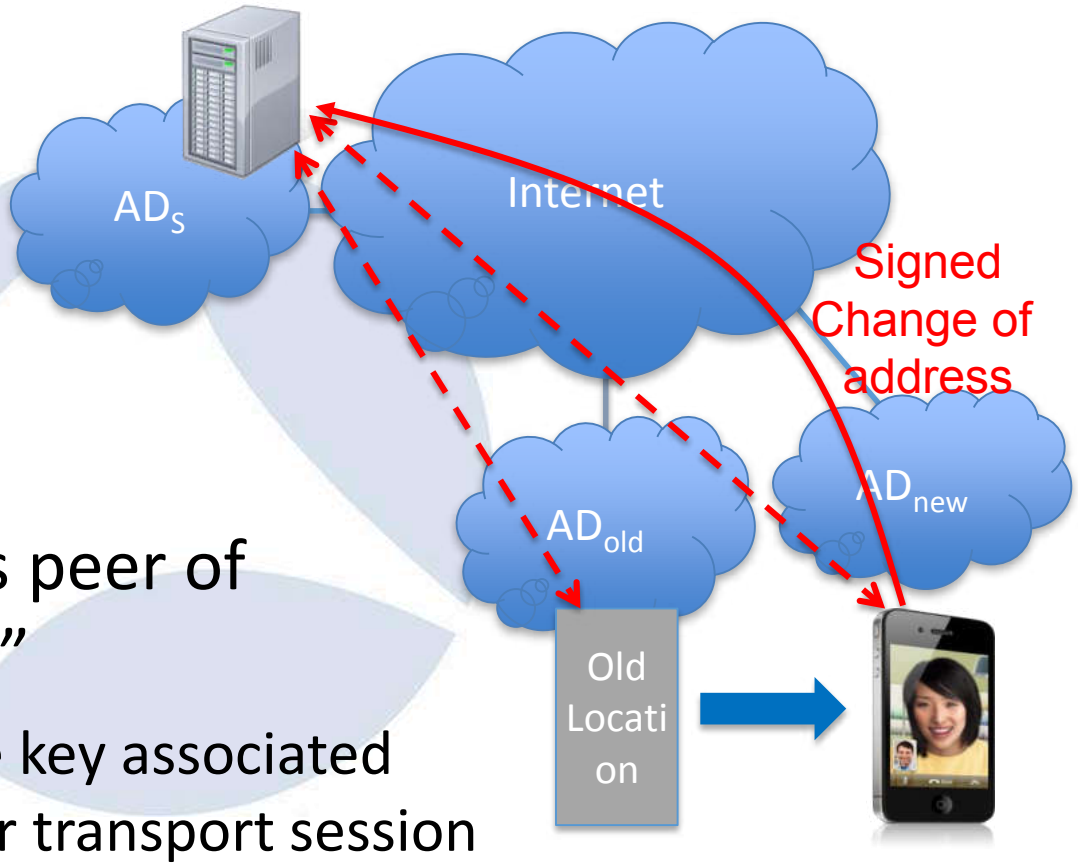
Mobility is a Key Requirement

- Inter-domain mobility remains a challenge in today's Internet
  - Active sessions
  - New sessions
- Requires separation of identifier and locator – for XIA:
  - Identifier = HID
  - Locator = DAG



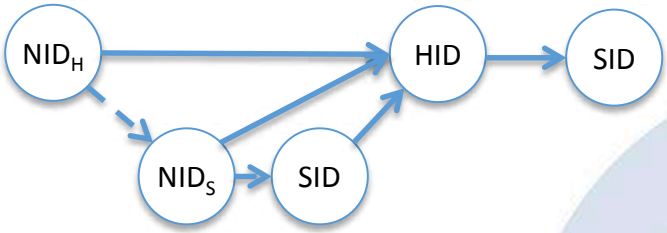
Maintaining an Active Session

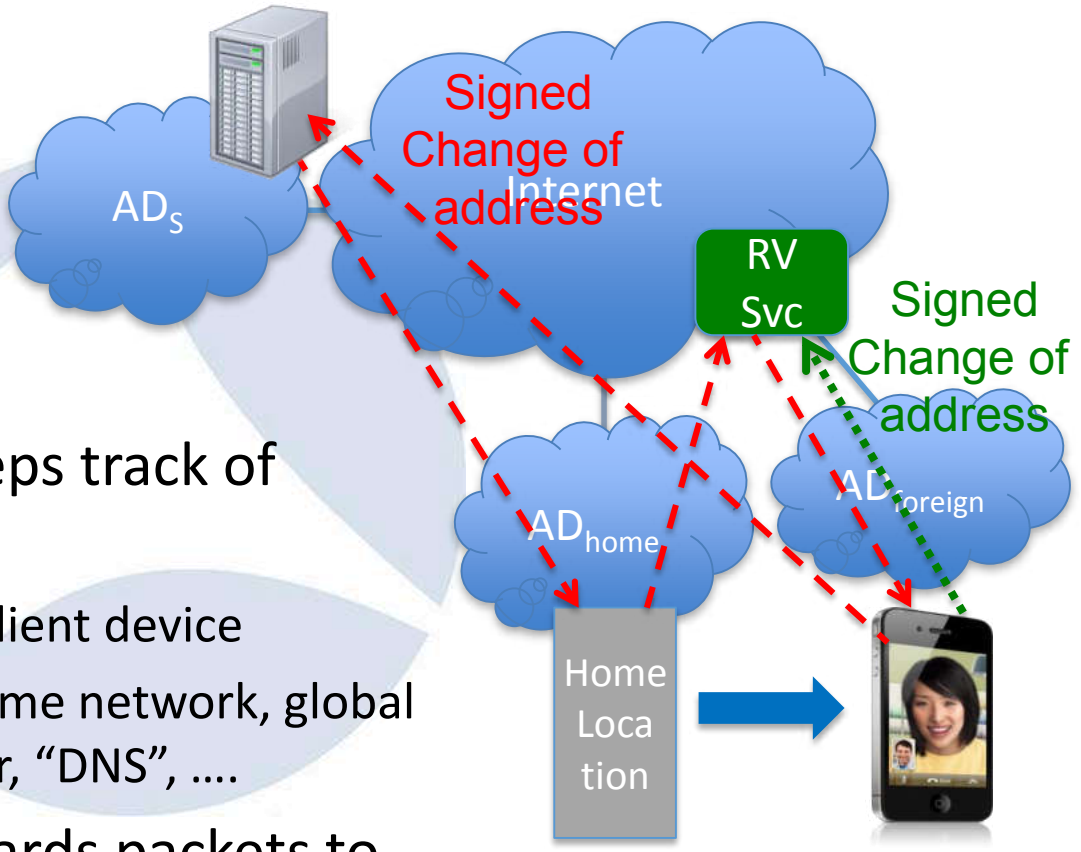




- Mobile host informs peer of "change of address"
  - Signed using private key associated with HID; sent over transport session
- Example of "rewriting" the DAG
  - Also: binding to service instance, in-path services, …

Finding a Mobile Device

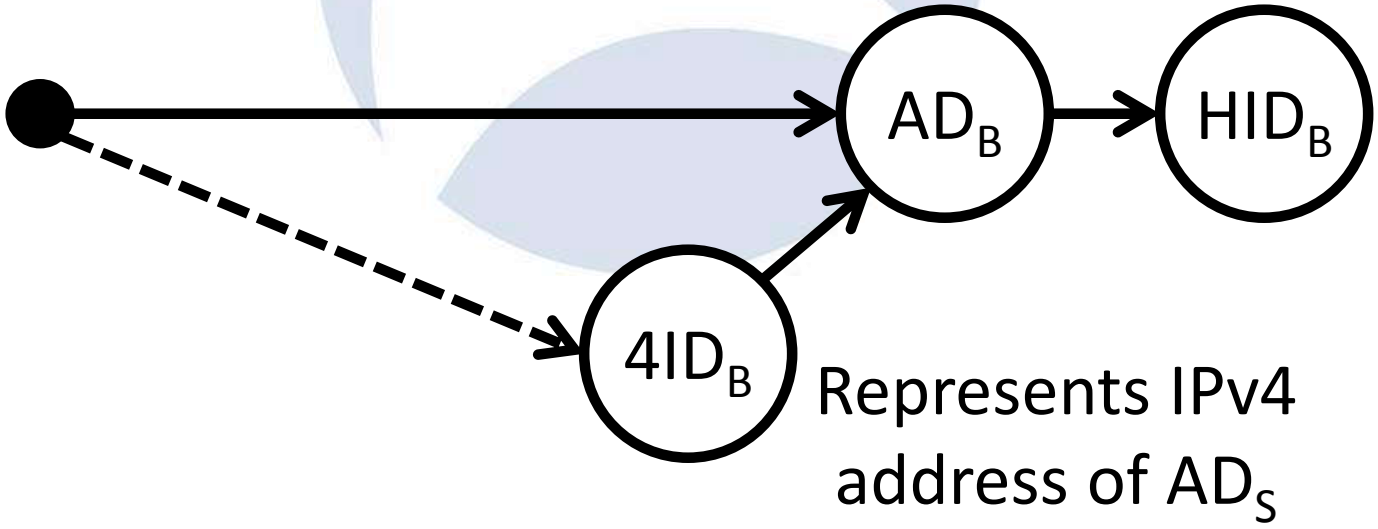




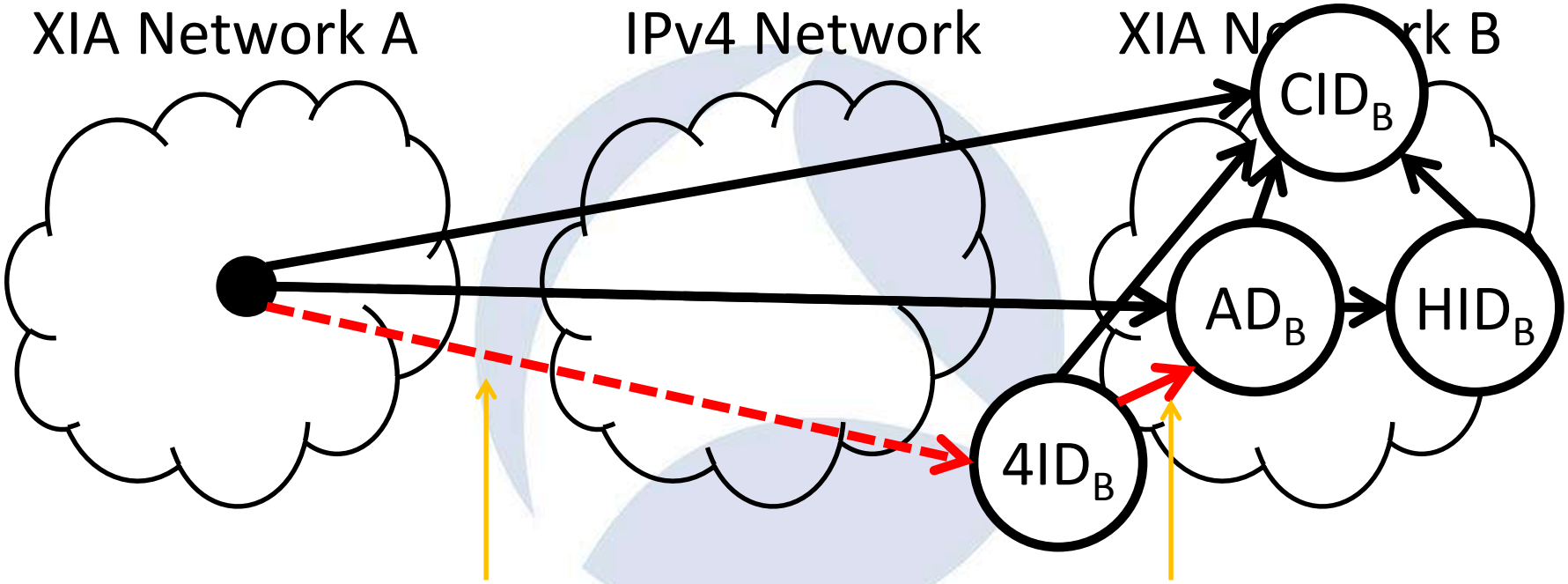
- Rendez-vous service keeps track of location of its users
  - Based on updates from client device
  - Can take many forms: home network, global service, hot spot provider, "DNS", ….
- Rendez-vous point forwards packets to mobile device, e.g., SYN
- Mobile device rewrite DAG

Incremental Deployment of XIA

- 4ID: IPv4 address as an XID
  - IPv4 encapsulation between XIA network islands
  - Leverages fallback for legacy networks
- No need for statically configured tunnels!



4ID in Action: Partially Deployed XIA Networks



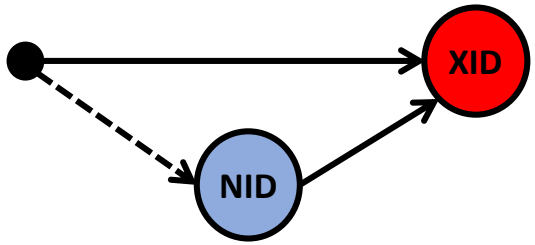
Entering IPv4 network: Encapsulate XIA packet with IP header

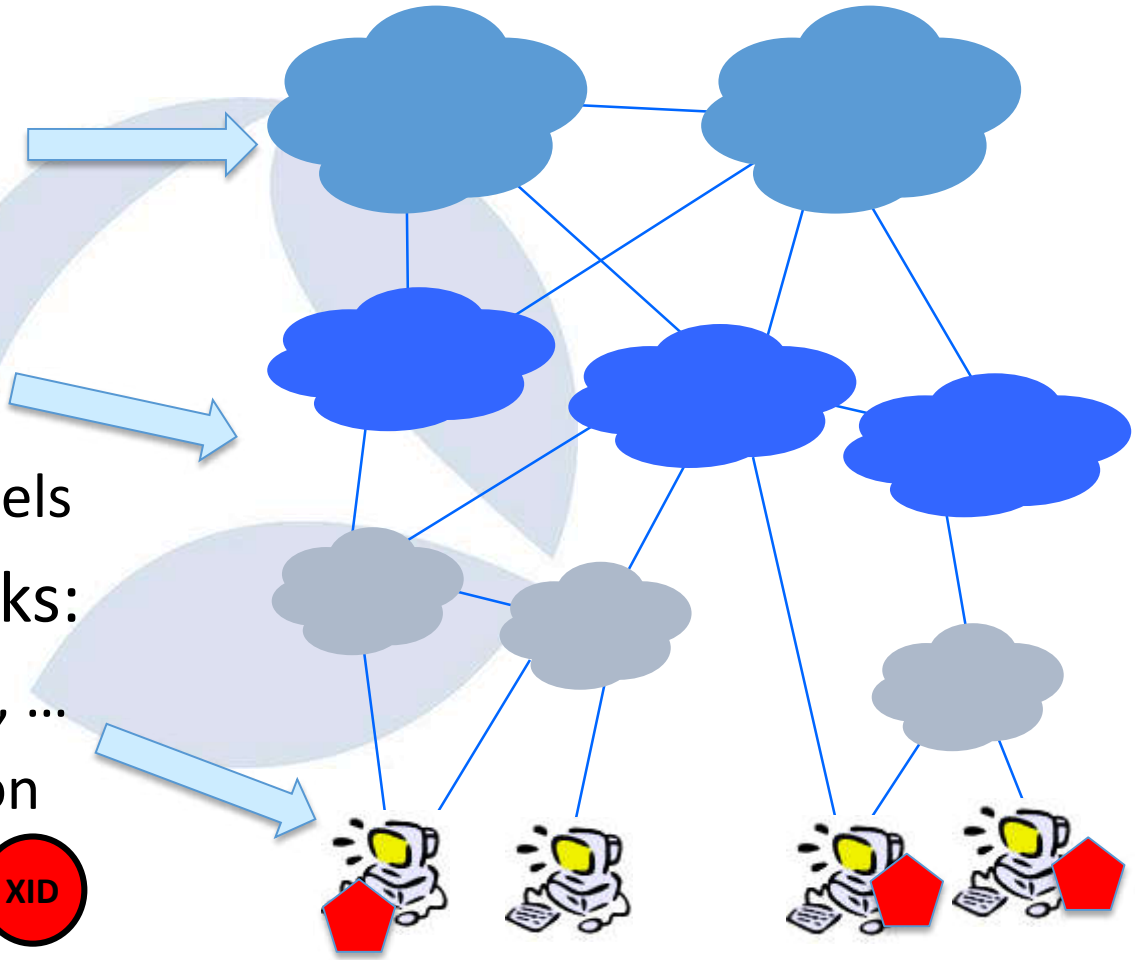
Entering XIA network: Remove IP header for native XIA packet processing

Dynamic encapsulation: no static tunnels

XIA Supports Network Diversity

- Core networks:
  - NID only
  - Same as today
- Eyeball networks
  - NID, CID, SID, …
  - New service models
- Customer networks:
  - NID, HIS, SID, CID, …
  - Also customization





Leveraging XIA Features

- Evolvability really is
  - Incremental deployment of new features
  - Dealing with legacy infrastructure
  - Diversity: “local” custom features in edge, core
- Built-in support for content and service centric networking
  - SID: routing, security, …
  - CID: file systems, video …
  - Anycast for availability, performance, …
- Richer functionality in the network
  - Multicast, pub-sub, DTN, content variants, …
  - In-network processing using SIDs in the DAG (e.g., mobility)
  - Platform for FIA networking research

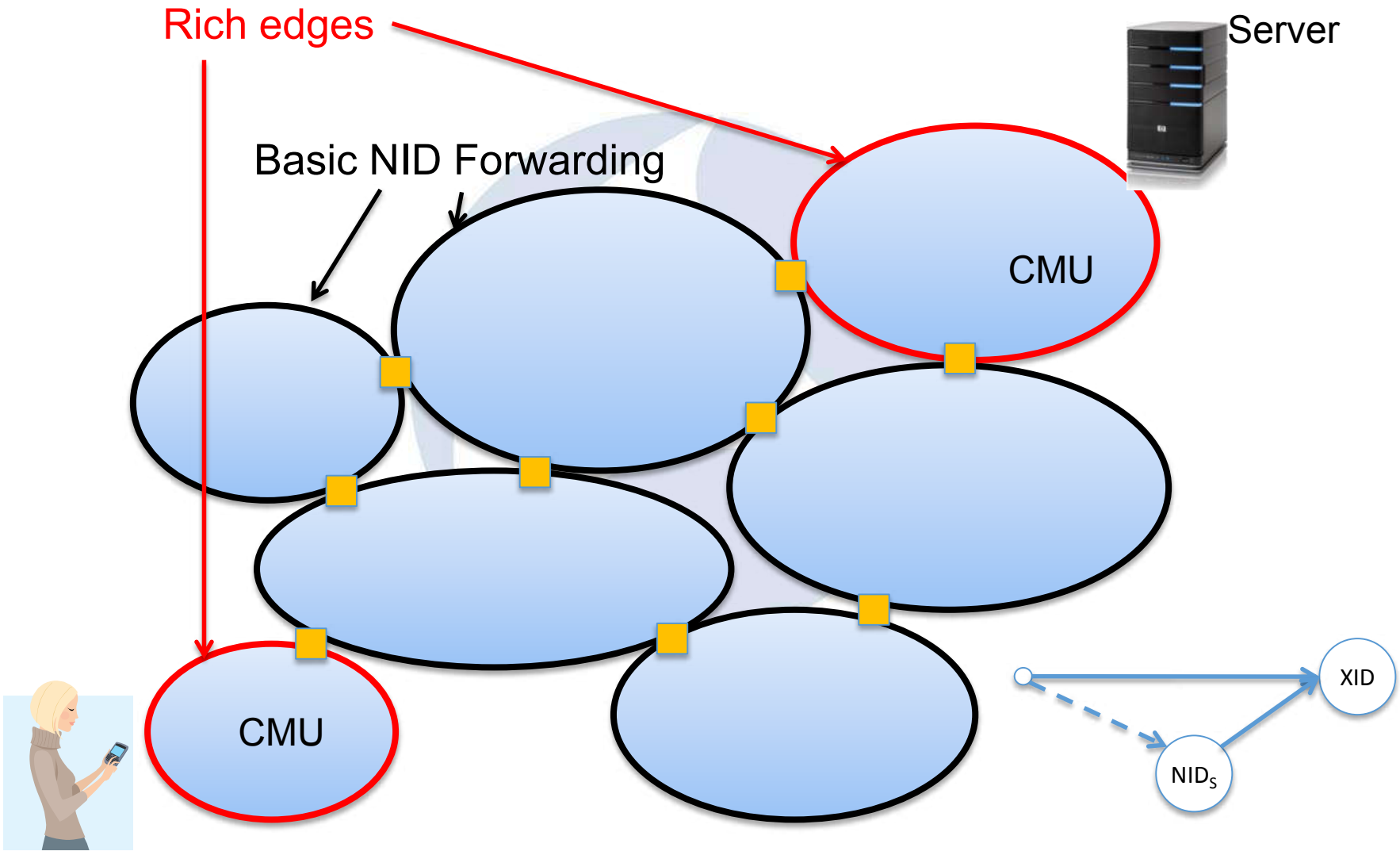
Outline

- Overview of XIA
- Some XIA use cases
- Some lessons learned
  - XIDs have different roles
  - Engineering a protocol stack

Overview of XIA IDs

- The set of XIDs can evolve over time and domains do not have to support all XID types
  - Expectation: rich network edge with a simple core
- Minimum requirement for interoperability is that all transit domains must support NIDs
  - Same as today: universal reachability of prefixes
- Each edge domain should support at least one "end-point" XID type

XID Support



Initial XID Types

- Network identifier identifies a network (NID)
  - Used for scoping, global connectivity
- Host identifier identifies a host (HID)
  - NID → HID equivalent to IP address
- Service identifier identifies a "socket" (SID)
  - In addition to or instead of an HID
  - Can be ephemeral or well-known SID
- Content identifier identifies static content (CID)
  - CID is hash of content

CIDs are Special

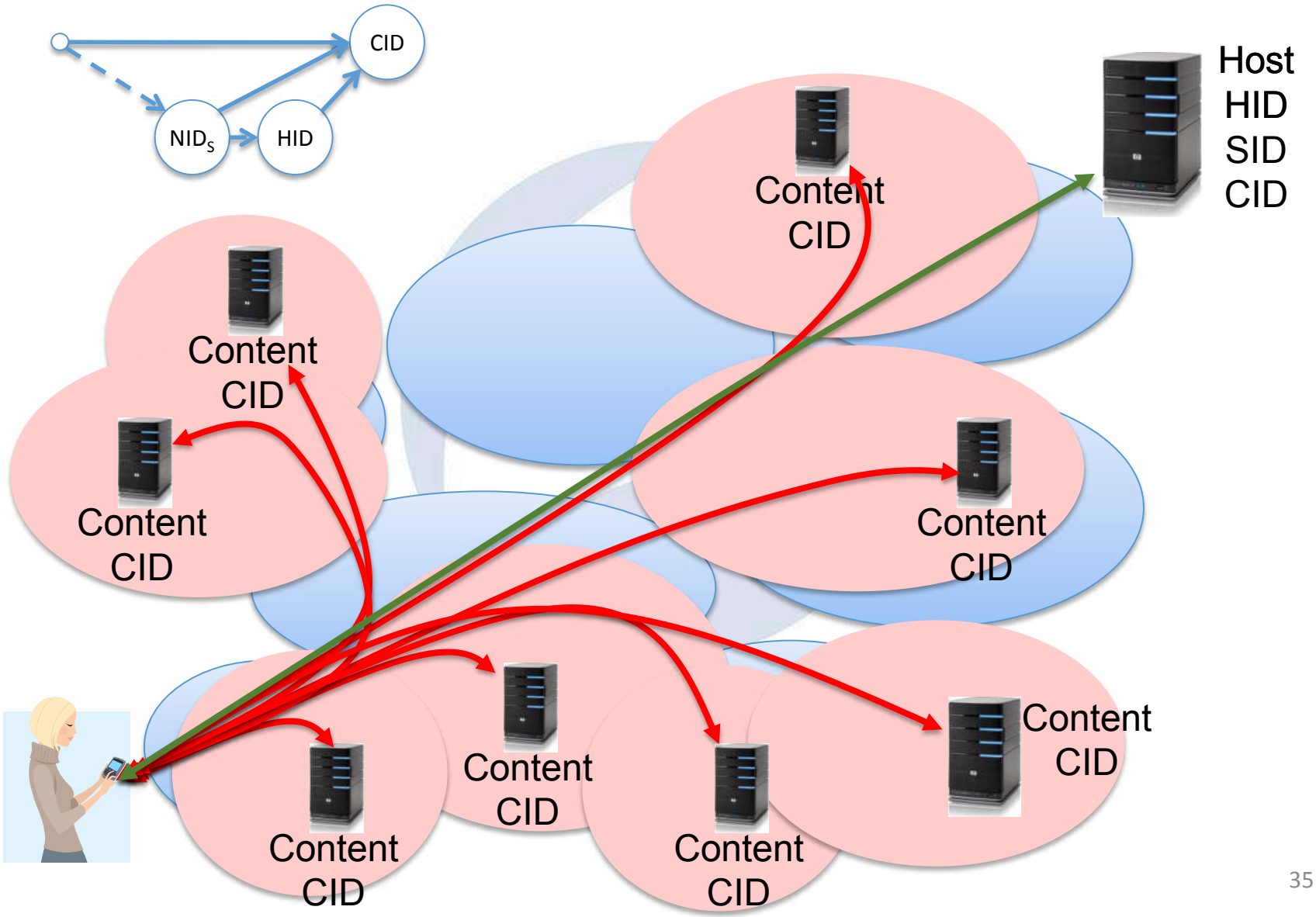
- CID lookup will often fail
  - Many domains/routers will not support CIDs
  - A lot of content may not appear in any cache
  - Many caches are too far “off path”

→ Opportunistic

- CIDs are only used for the request packet
  - Data delivery uses address provided by client
  - Typically an NID → HID → SID address

→ Discovery packets

Content Retrieval based on CIDs



New Discovery XID Types

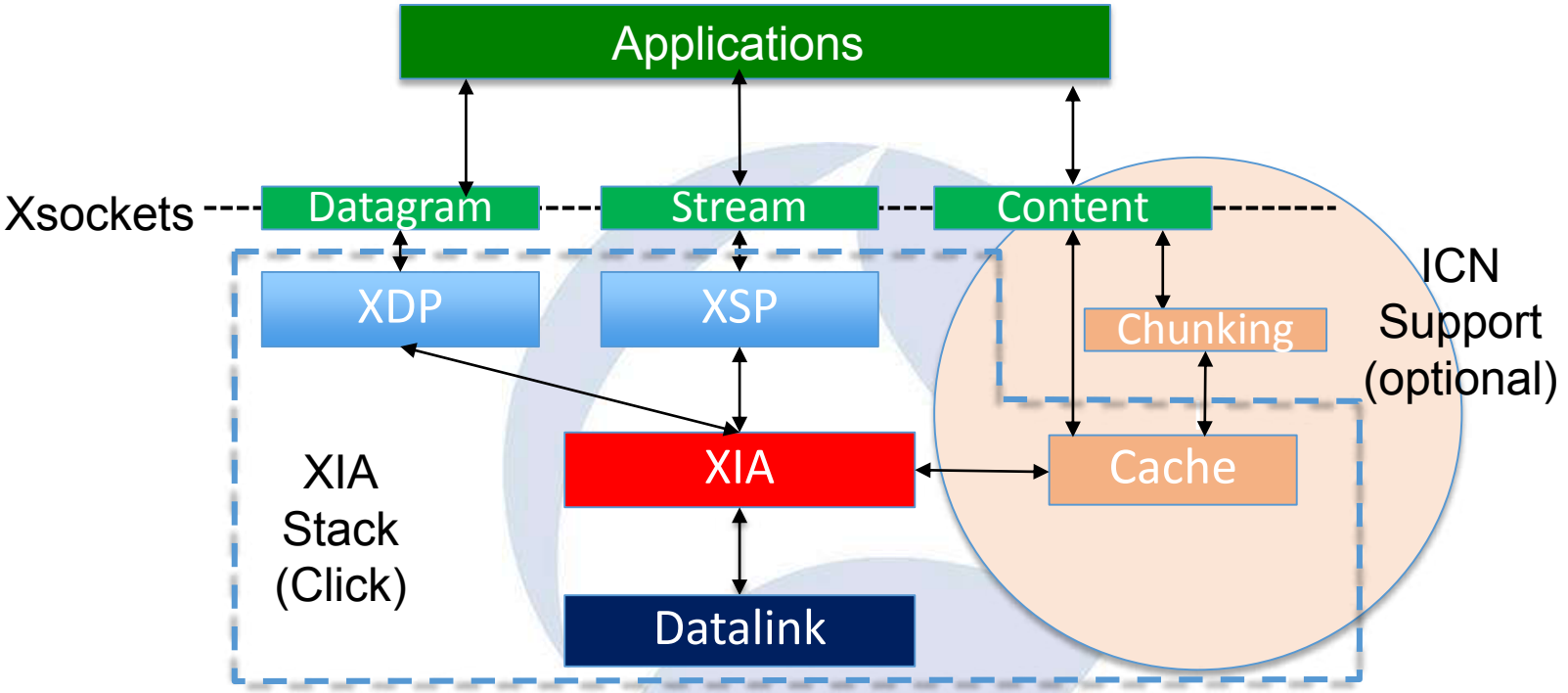
XID type depends on the "search criteria"

- "Named" CID: the nCID is derived from the name (not the content)
  - Can be used for certain types of dynamic content
  - Intrinsic security based on signing of content
- Anycast SID: access replicated services
  - Forwarding tables controlled by service provider based on its internal criteria
- (Pub-sub CID based on bloomfilters)

Building an Information-centric XIA-based Network

- Is XIA an ICN architecture - No, sorry
  - Networks must do more than retrieving content
  - There are many different types of content
  - Information retrieval is complex
- But XIA-based networks can be “information-centric”!
  - How do we engineer such a system?
  - Does “content” replace “host” as destination, or we add it as a first class citizen?

Original XIA Implementation

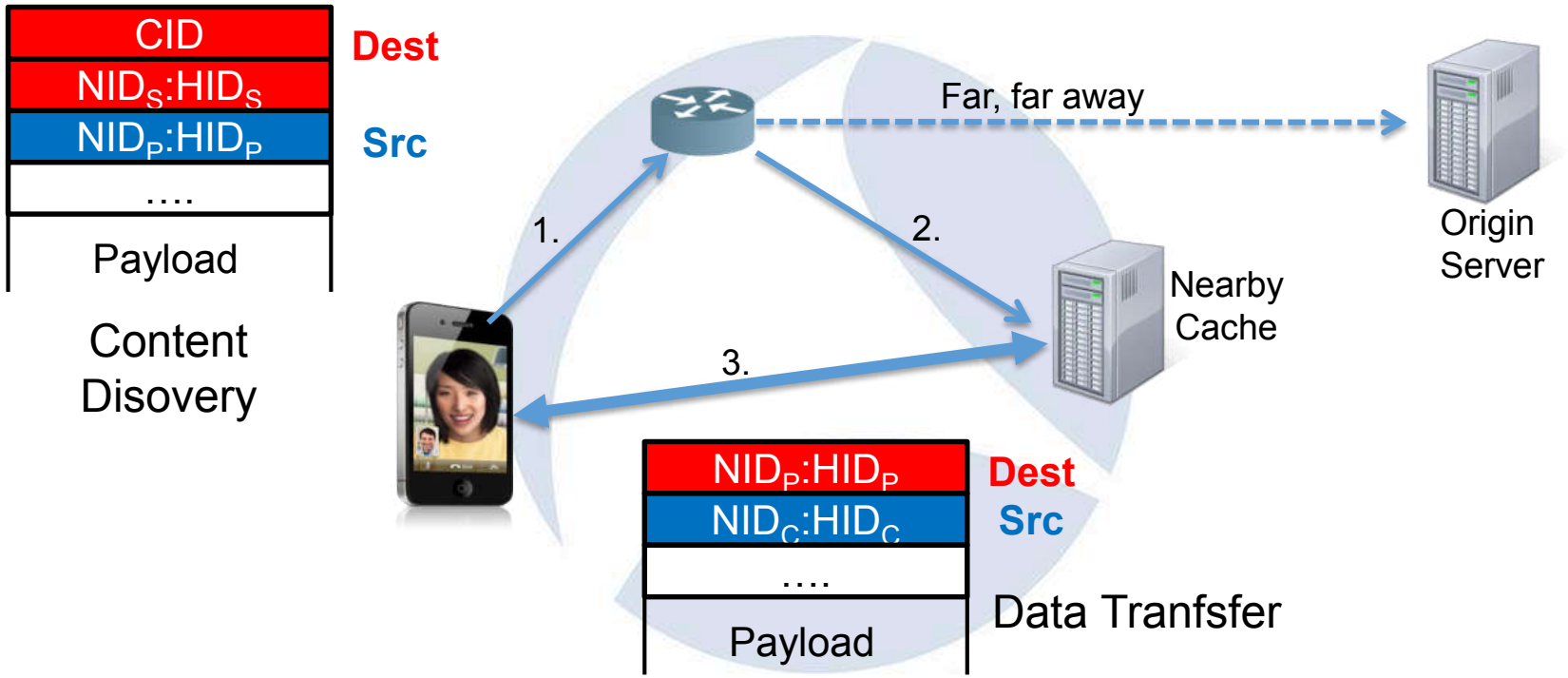


- CID-based ICN support tightly integrated into protocol stack
- Semantics: packet sent to CID requests content
  - Any node with the content replies with the content

Design Creates Many Problems

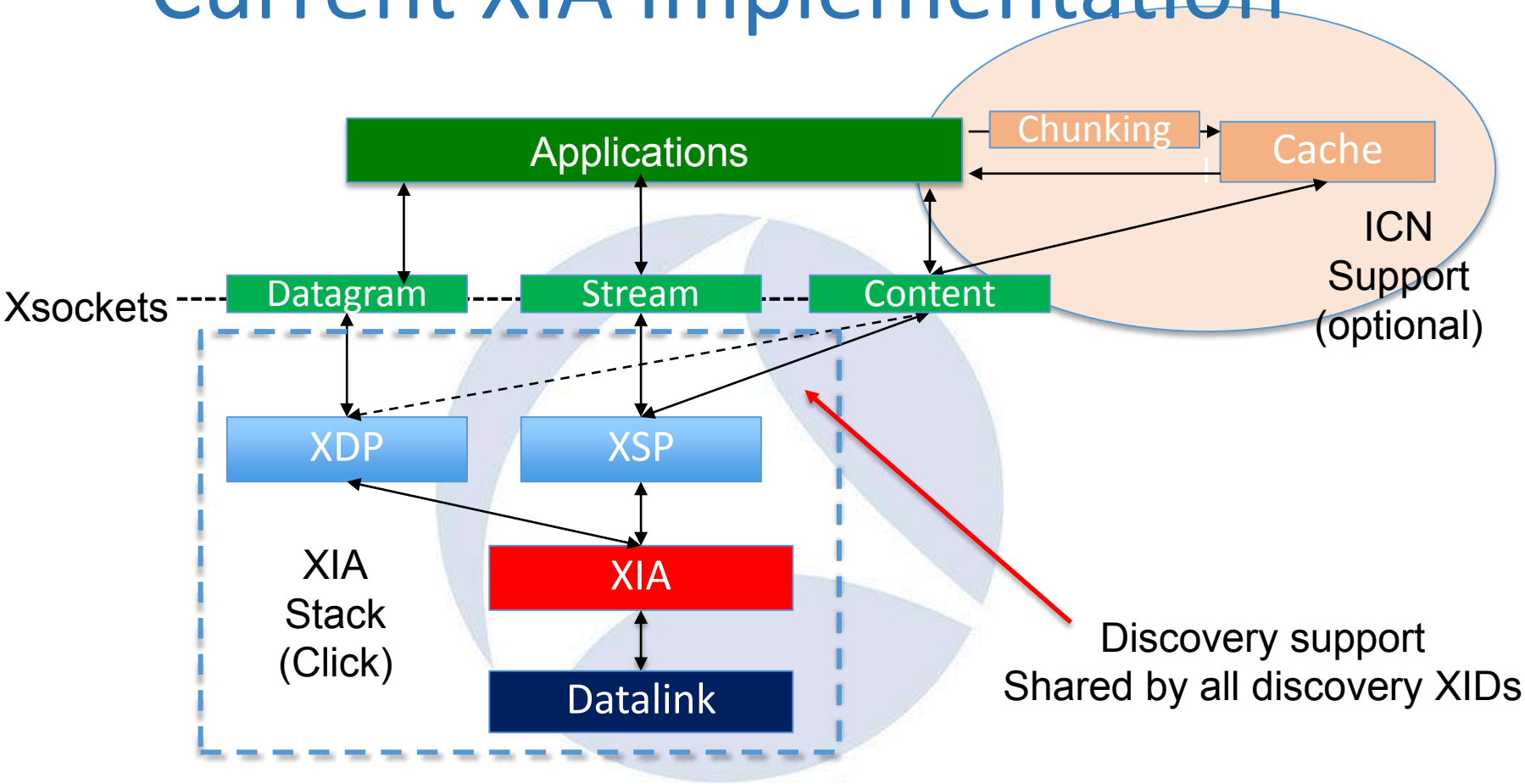
- Protocol stack was large and complex
  - Caches are large, and involve a lot of policy
  - CID support is mainly relevant at network edge!
- Performance was poor
  - Datalink packets are small!
  - Downloading a video requires a huge number of request, each of which is unreliable
- Why should “content” get preferential treatment?

How CIDs work today: They Discover Content



- CID request “discovers” nearest copy of content (cache, origin)
- Cache delivers content over reliable stream connection
- Client checks content using CID intrinsic security

Current XIA Implementation



- Separating unicast data transfer from discovery makes architecture simpler and more generic
- Transport choice is orthogonal decision

Conclusion

- Experience with the XIA prototype and experiments provided many insights
  - Multiple XID types offers a principled way of adding functionality to the network
  - Fallbacks allow not only evolvability but also opportunistic and custom XID types
  - Intrinsic security checks authenticity endpoints
- XIA's flexibility supports ICN network design but not at the detriment of other functions

https://github.com/xia-project/

- Core stack: XIA, datagram, streaming, sockets
- Support for NID, HID, SID, CID, mobility, FID, 4ID, nCID, aSID
- Netjoin protocol: host/router joining network
- Bootstrapping an SDN-controlled domain
- Routing, naming, XARP, XCMP, ..
  - Relatively basic implementations
- Sample applications, scripts, utilities, …