# XIA: Efficient Support for Evolvable Internetworking

Dongsu Han            Ashok Anand              Fahad Dogar

Boyan Li          **Hyeontaek Lim**       Michel Machado

Arvind Mukundan              Wenfei Wu

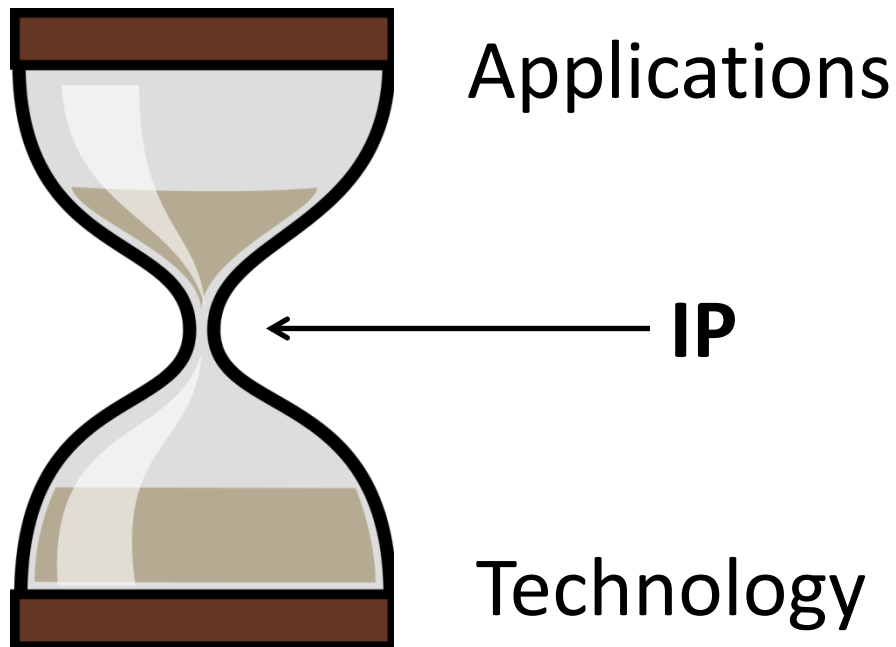Aditya Akella      David G. Andersen        John W. Byers

Srinivasan Seshan             Peter Steenkiste

**Carnegie Mellon**    **BOSTON UNIVERSITY**    **WISCONSIN** UNIVERSITY OF WISCONSIN–MADISON

# IP: Narrow Waist of the Internet

Applications

Innovation both above and below IP

**But what about IP?**

**IP**

Technology

# Proposed ☐-Centric Networking

- Service: Serval (This NSDI!)
- Content: Named Data Networking
- Mobility: MobilityFirst
- Cloud: Nebula

Problem: Focusing on one communication type may hinder using other communication types, as occurred to IP

Can we support **heterogeneous** communication types on a **single** Internet architecture?

# Future ☐-Centric Networking

- Service, content, mobility, and cloud did not receive much attention before as now

- Yet more networking styles may be useful in the future

  – E.g., DTN, wide-area multicast, …?

Problem: Introducing additional communication types to the existing network can be very challenging

Can we support **future** communication types **without redesigning** the Internet architecture?

# Legacy Router May Prevent Innovation

Internet
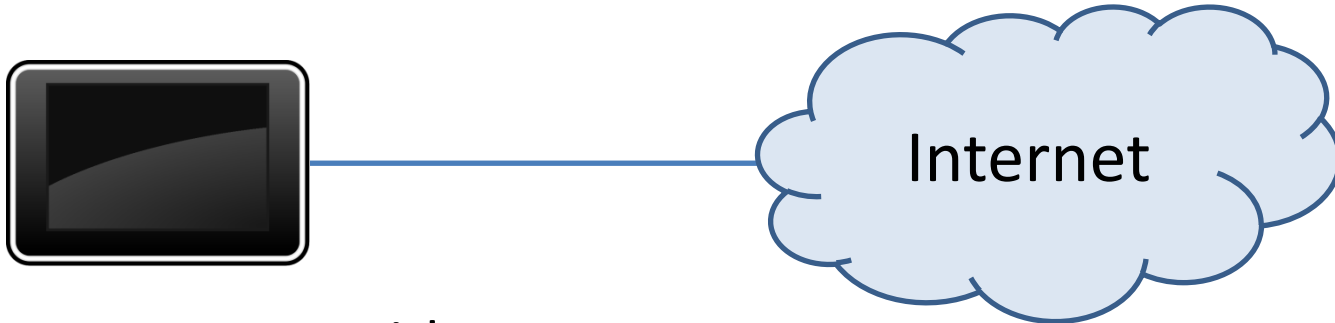
"I got a computer with Awesome-Networking announced in NSDI 2022! Can I use it right now?"

"Ouch, we just replaced all of our routers built in 2012. Can you wait for another 10 years for new routers?"

Problem: Using a new communication type may require every legacy router in the network to be upgraded

Can we allow using a **new** communication type even when the network is **yet to natively support** it?

# XIA's Goals and Design Pillars

## "Principal types"

Support multiple communication types (heterogeneity)

Support future communication types (evolution)

## "Fallbacks"

Allow using new communication types at any point (incremental deployment)

# Principal Types

Define your own communication model

# Principals

**Current Internet**

**XIA**

| Principal type | Type-specific identifier |
|:---:|:---:|

| IP address |
|:---:|

| Host | 0xF63C7A4... |
|:---:|:---:|

Hash of host's public key

| 128.2.10.162 |
|:---:|

| Service | 0x8A37037... |
|:---:|:---:|

Hash of service's public key

| Content | 0x47BF217... |
|:---:|:---:|

Hash of content

| Future | ... |
|:---:|:---:|

# Principal Type-Specific Semantics

| Host | 0xF63C7A4… |
|------|------------|

Contact a host

| Service | 0x8A37037… |
|---------|------------|

Use a service

| Content | 0x47BF217… |
|---------|------------|
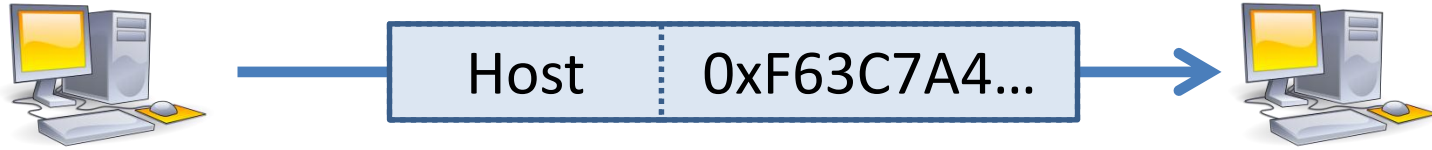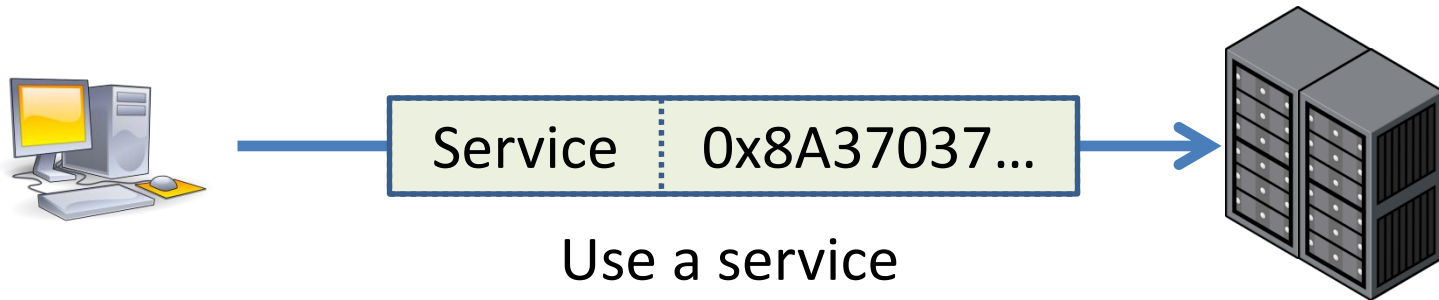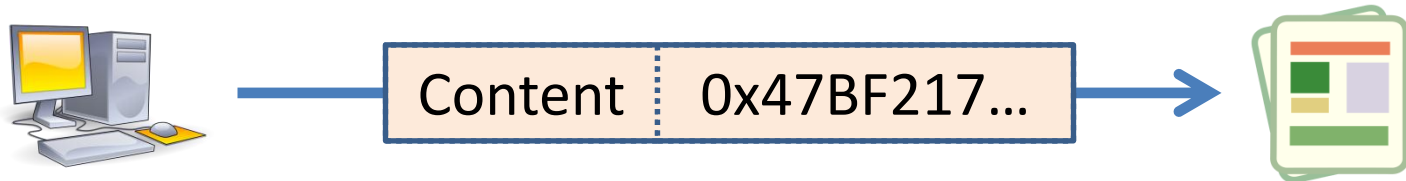
Retrieve content

# Principal Type-Specific Processing



Input → **Common processing** | Host-specific processing / Service-specific processing / Content-specific processing / … → Output

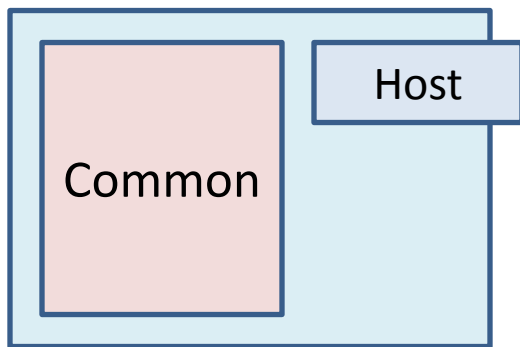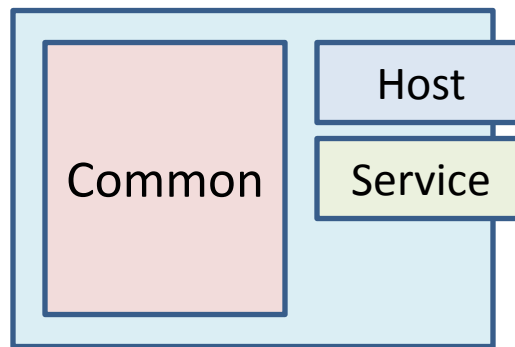XIA router
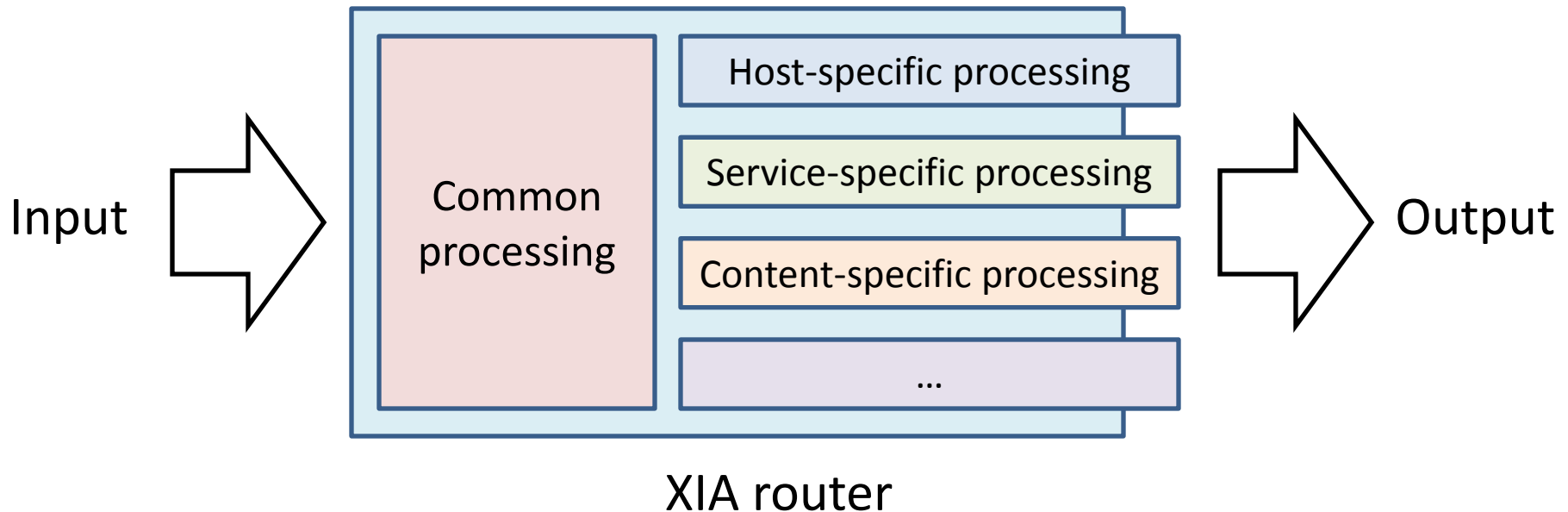
- Type-specific processing examples
  - Service: load balancing or service migration
  - Content: content caching
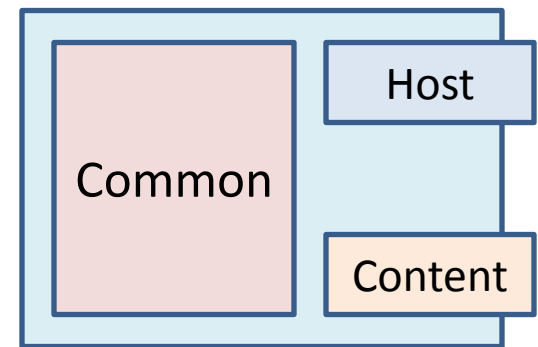
# Routers with Different Capabilities

- Routers are **not** required to support every principal type
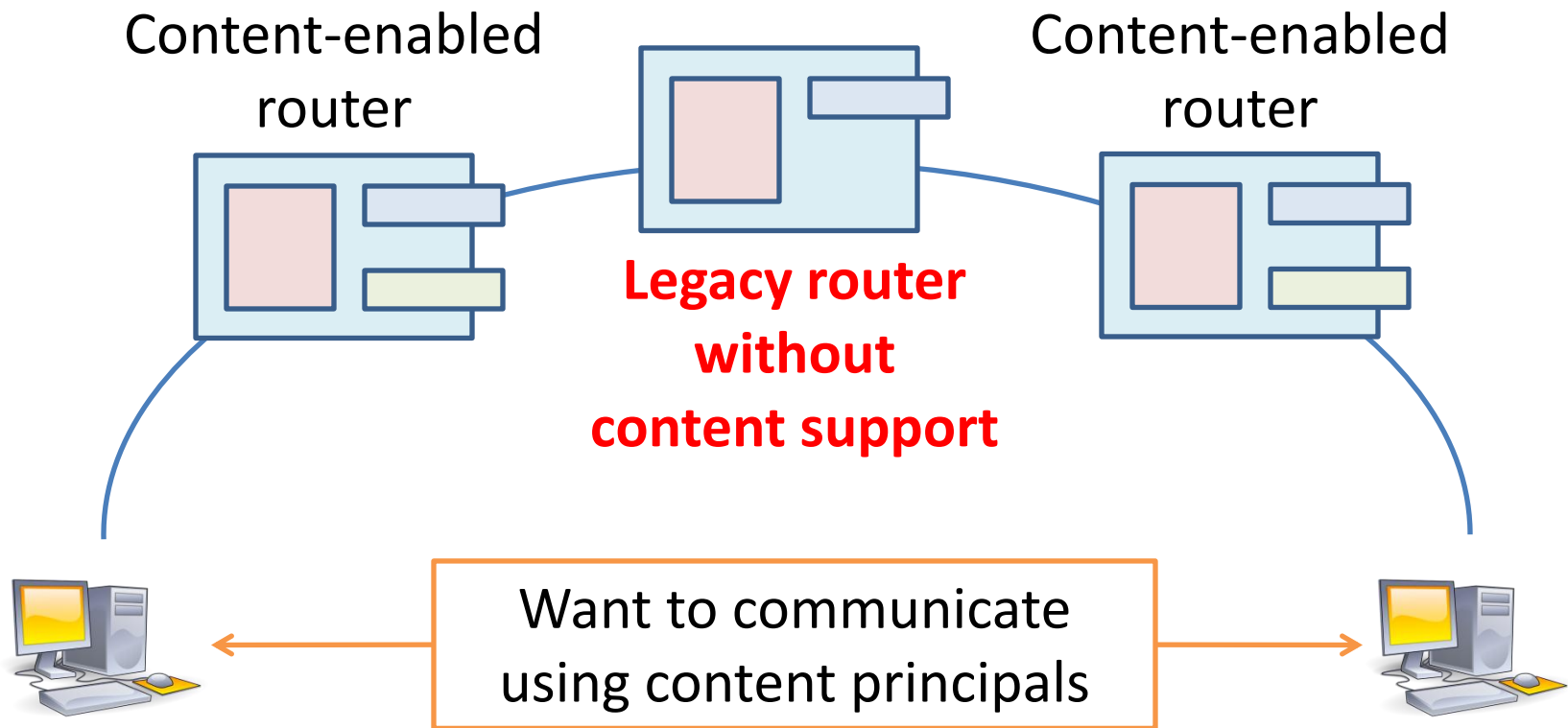    - The only requirement: Host-based communication



Host-only router

Service-enabled router

Content-enabled router

# Using Principal Types that are Not Understood by Legacy Routers?

Content-enabled router

Content-enabled router

**Legacy router without content support**

Want to communicate using content principals

# Fallbacks

Tomorrow's communication types… today!

# Fallbacks: Alternative Ways for Routers to Fulfill Intent of Packet

Intent: Retrieve Content

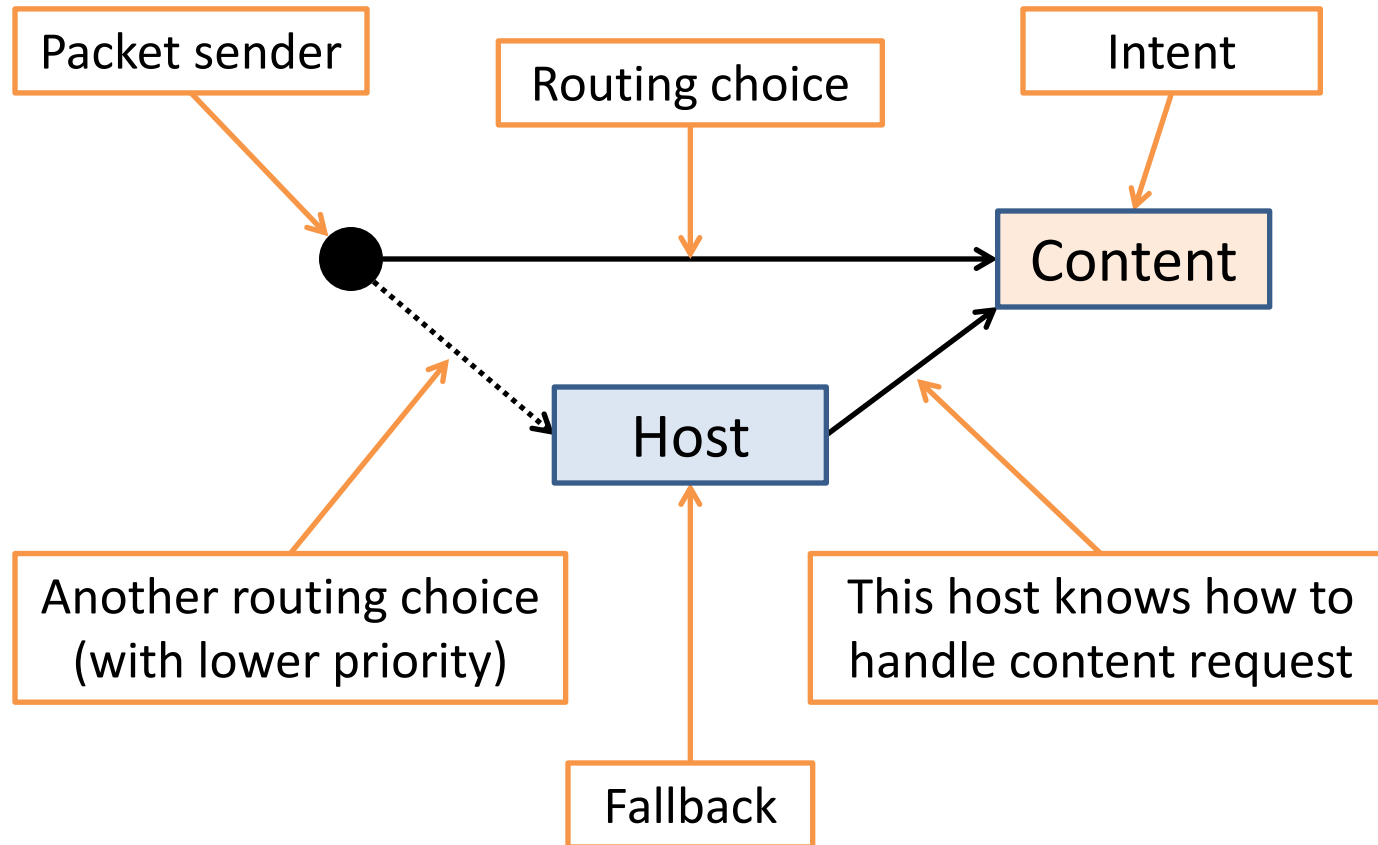**Fallback**: Contact Host ,

who understands Content request

What the network does:

- With content-enabled routers, use Content for routing

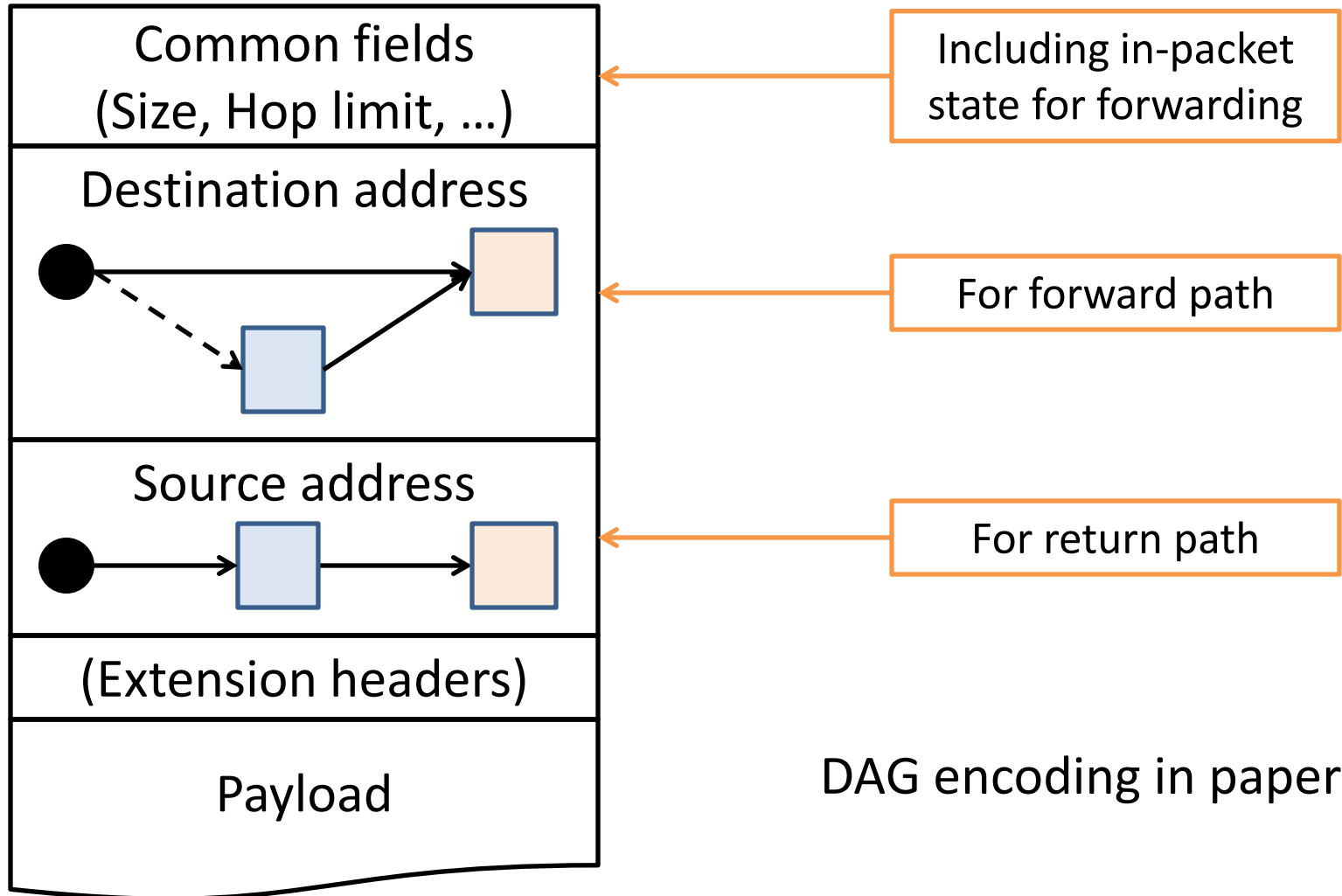- Otherwise, use Host for routing (always succeeds)

# DAG-Based Address

Your address is more than a number

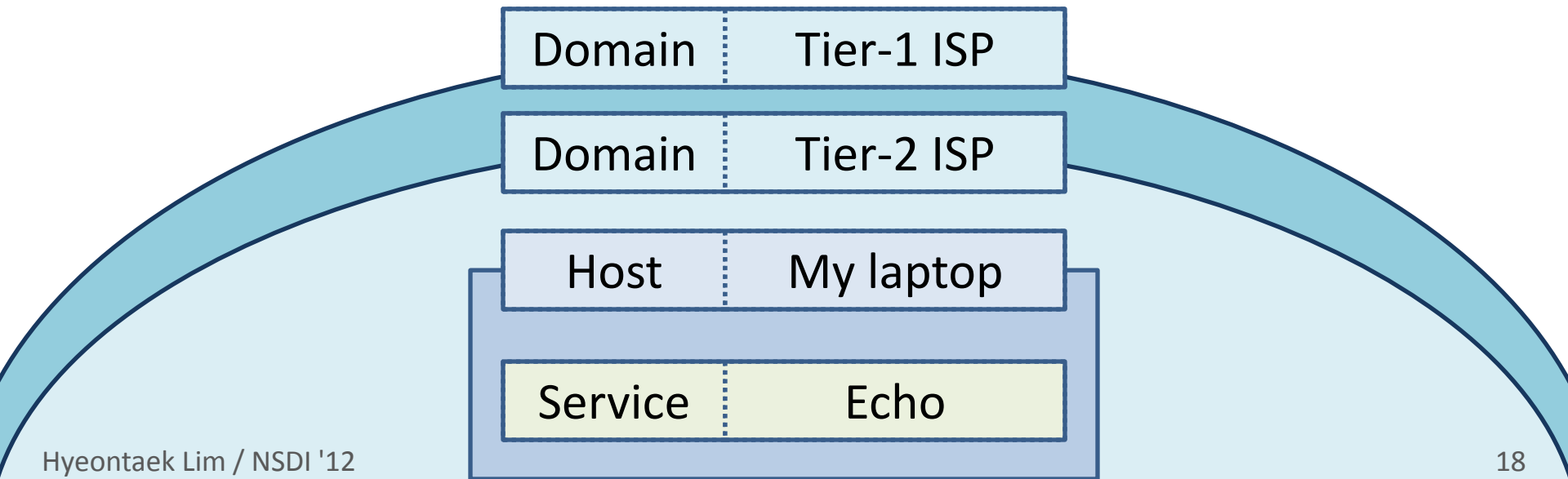# DAG (Direct Acyclic Graph)-Based Addressing Enables Fallbacks
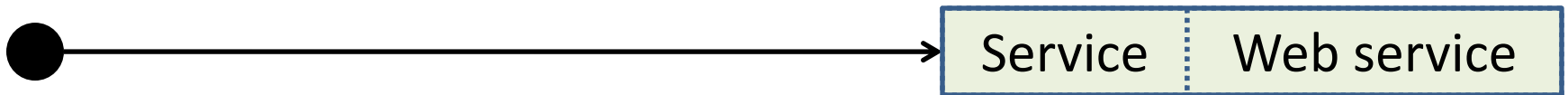


Packet sender

Routing choice

Intent

Content

Host

Another routing choice (with lower priority)

This host knows how to handle content request

Fallback

# DAG Addresses in Packet Header

Common fields
(Size, Hop limit, …)

Including in-packet
state for forwarding

Destination address

For forward path

Source address

For return path

(Extension headers)

Payload

DAG encoding in paper

# Scoping Using DAG

Tier-1 ISP → Tier-2 ISP → My laptop → Echo

More specific intent & Better scalability

| Domain | Tier-1 ISP |
|--------|------------|
| Domain | Tier-2 ISP |
| Host | My laptop |
| Service | Echo |

# Service Binding with DAG

Initial contact to a service

● ——————————————————————→ | Service ┊ Web service |

When a particular host should serve subsequent service requests
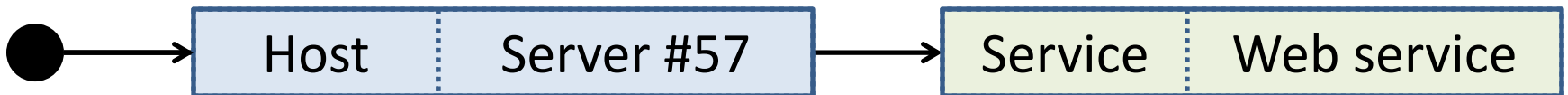
● —→ | Host ┊ Server #57 | —→ | Service ┊ Web service |

| "Late binding" |

# DAG Allows Nested Fallbacks
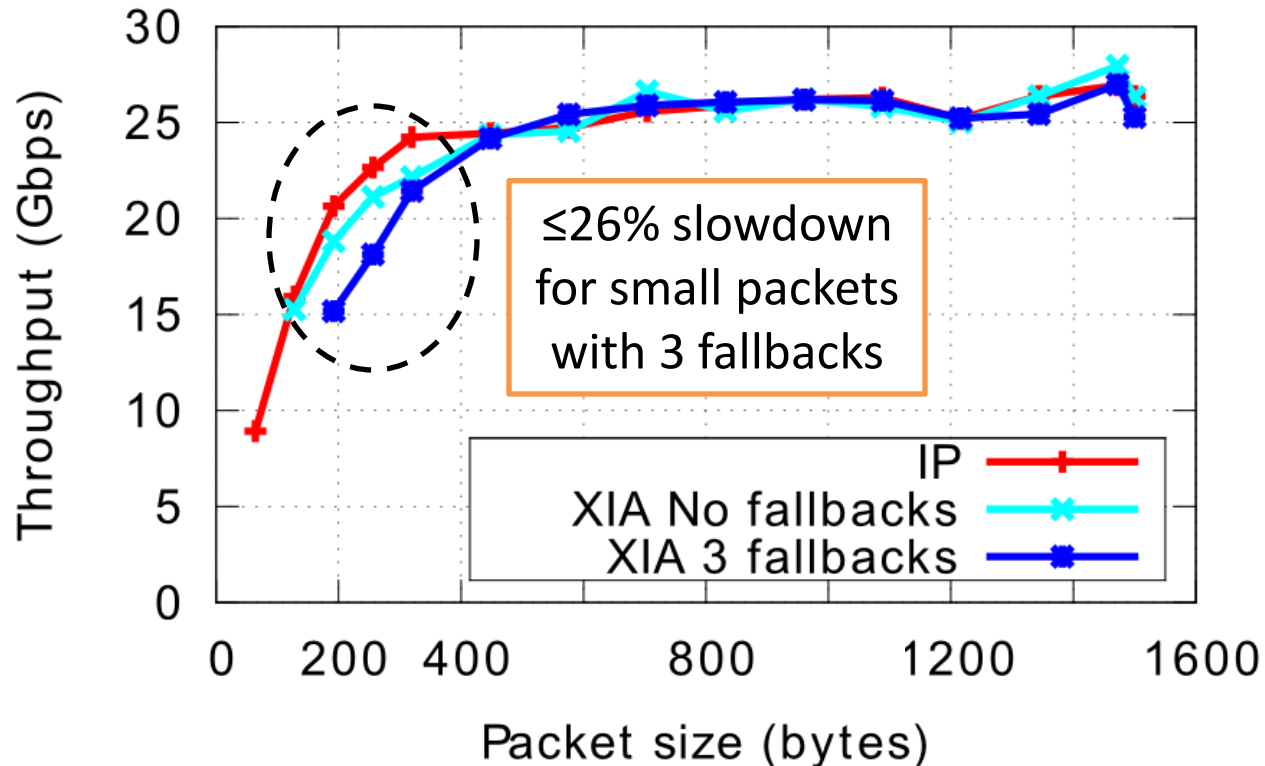


Domain → Host → Service → Content
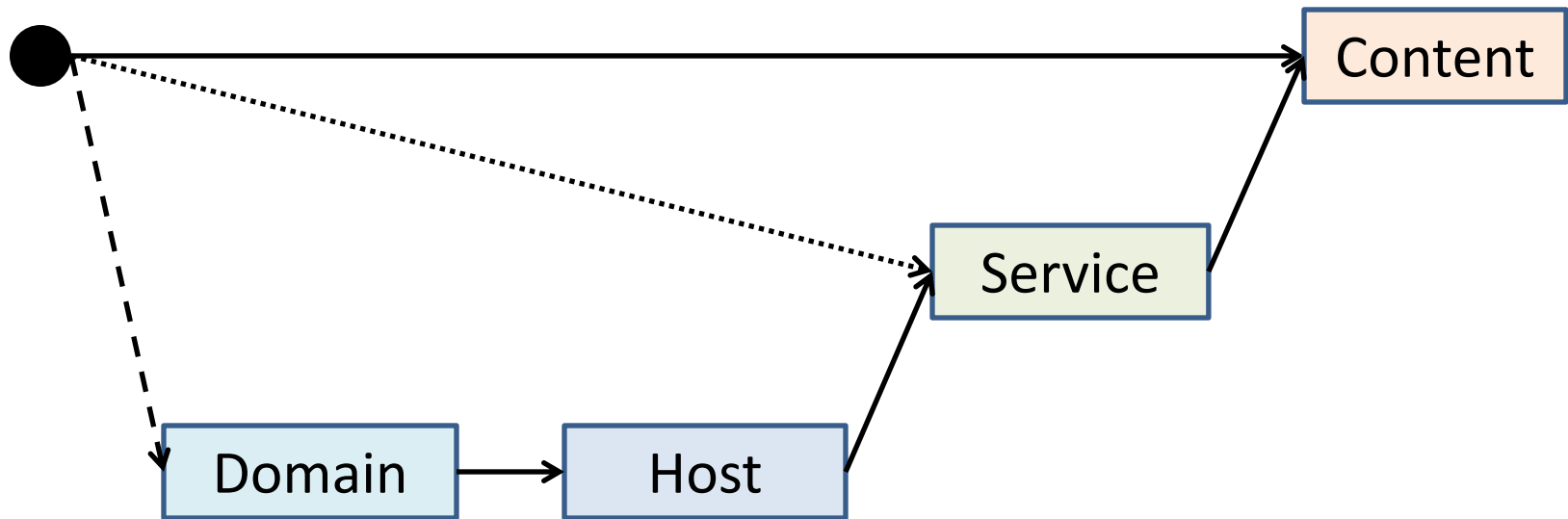
Strong support for evolvable internetworking

# Can We Forward DAGs Rapidly?

Expressive ≠ Expensive

# XIA Software Router's
# High Forwarding Throughput



≤26% slowdown for small packets with 3 fallbacks

Click-based implementation on commodity hardware
351 K table entries based on a Route Views snapshot

# XIA: eXpressive Internet Architecture

- Support for evolvable internetworking
  - Heterogeneous communication types
  - Future communication types
  - Incremental deployment of new communication types


- Principal types & fallbacks


- DAG-based addressing

# XIA: Enabling Evolution by eXpression
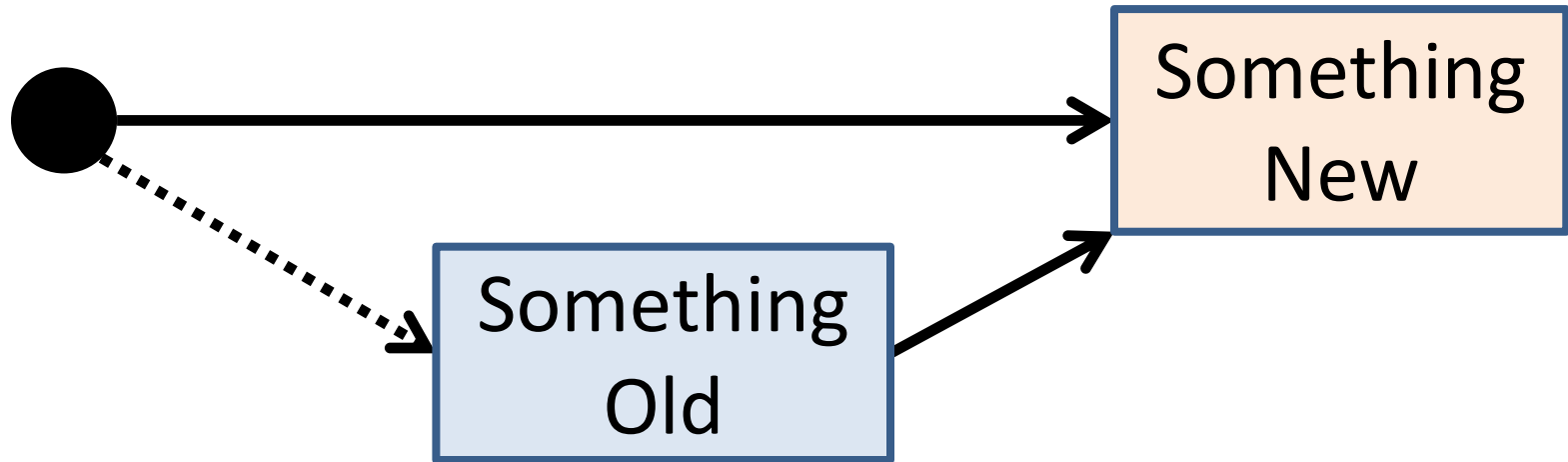


- Prototype: github.com/XIA-Project/xia-core
  - Router, socket, content cache, etc.
  - Supports LAN, XIA-over-IP, GENI