Abstracting IP over ICN

I-CAN Workshop
June 2, 2015

George Xylomenos
AUEB-RC
Outline

- Motivation
- POINT Architecture & Implementation
- Why abstractions?
- Basic abstractions: IP & HTTP
- Possible abstractions: TCP, CoAP, MPEG DASH
- Conclusions
Motivation

• POINT plants ICN below IP
  – Way to sneak ICN to unsuspecting Internet!
  – Must offer better performance than IP
    • Where “better” may mean various things
  – Must offer compatibility to IP services

• Mapping needed from IP to ICN concepts
  – Possibly from TCP, CoAP, HTTP, MPEG-DASH
    • If the extra complexity makes sense...
The realization of the ICNx interfaces varies depending on the dissemination strategy.

The IP interface represents the various supported abstractions, such as HTTP, CoAP, TCP or IP.
Why Abstractions?

• IP/TCP/HTTP and ICN offer different concepts
  – IP: endpoint address, packet
  – TCP: connection, byte stream
  – HTTP: URL, GET, PUT
  – ICN: publisher, subscriber, publication, SID/RID

• Abstractions map IP/TCP/HTTP to ICN
  – Semantic mapping depends on protocol
  – Higher layers offer more meaningful semantics
IP Abstraction (1 of 3)

- IP addresses are mapped to ICN names
  - Hierarchical namespace for IP hierarchy
  - Separate namespace for internal/external IPs
  - Subscribe to name to receive IP packets
  - Publish to name to send IP packets
  - NAP represents “internal” UEs
  - GW represents rest of Internet
  - Some changes need to BlackAdder rendezvous
IP Abstraction (2 of 3)

Namespace

- Root identifier
- Scope identifier for inside ICN communication
- Scope identifier for outside ICN communication

- Class A
- Class B
- Class C
- Class D

IP addresses inside ICN network
IP addresses outside ICN network
Design for the IP-Level Abstraction Handler

Remember that in relation to ICN core node, this is an application!
TCP Abstraction

• Similar starting point with IP
  – Mapping of IP address + Port pairs to names
  – Reliable transport required for packets

• Not extremely interesting
  – Have explored it in PSIRP/PURSUIT
  – Most applications of interest use HTTP
  – TCP requires considerable complexity over IP
  – TCP semantics are not very useful for ICN
HTTP Abstraction (1 of 2)

- **GET request mapped to server FQDN**
  - Clients publish requests to FQDN
    - Includes full URL in data
  - Server subscribes to its FQDN only
- **Response mapped to full URL**
  - Server publishes response to full URL
  - Client subscribes to full URL
  - May serve multiple clients at once!
Is the abstraction worthwhile?

- Why not do it over IP? TCP?
- HTTP has a specific structure
  - Request – response are (clearly) paired
  - Much better match with ICN
  - Many opportunities for optimizations

- IP and TCP are not important for HTTP
  - Any reliable transport will do
  - TCP/IP can be terminated at the NAP/GW
CoAP Abstraction

- CoAP is very similar to HTTP
  - Also based on URLs
  - Request/response protocol
  - Can work over UDP instead of TCP
  - Similar approach as for HTTP
    - GET to FQDN, response to URL
  - No need to terminate TCP connections
  - Are there enough CoAP apps to justify it?
MPEG DASH Abstraction

• MPEG DASH is a use case of HTTP
  – First GET an MPD describing a set of streams
  – Then GET the media segments
    • Segments identified via URLs

• Does an MPEG DASH abstraction make sense?
  – For: media segments are correlated
    • Opportunities for multicast and pre-fetching
  – Against: MPD file needs to be parsed
    • State must be maintained depending on MPD
Conclusions

• POINT is based on abstractions
  – Mapping of IP concepts to ICN ones
  – May operate at different semantic levels

• Basic abstractions
  – HTTP: captures most applications at a high level
  – IP: offers low level compatibility for others

• Additional abstractions?
  – TCP, CoAP, MPEG DASH, ...
References

Thank you

xgeorge@aueb.gr

POINT

i-CAN