CAT: A Last Mile Protocol for Content-Centric Networks

Sasu Tarkoma, Dmitriy Kuptsov, and Petri Savolainen

Helsinki Institute for Information Technology

University of Helsinki and Aalto University

Pasi Sarolahti

Aalto University

FutureNet IV Fourth International Workshop on the Network of the Future, in conjunction with IEEE ICC

Contents

- Introduction
- Background
- CAT overview
- Analysis and discussion
- Conclusions

Introduction

- In recent years, content-centric networking has become an active area of research
- Proposals in this area often use custom protocols for the last mile communication between end users and content networks, which makes the technologies hard to be adopted
- In this paper we present a content-aware publish-subscribe protocol, called CAT, that can become a common solution for end users to access different content-centric networks
  - Leverage parallel access: performance, availability
  - Leverage caching and multiple sources
  - Offer a unifying interface and protocol to content

Content-centric Networking (CN)

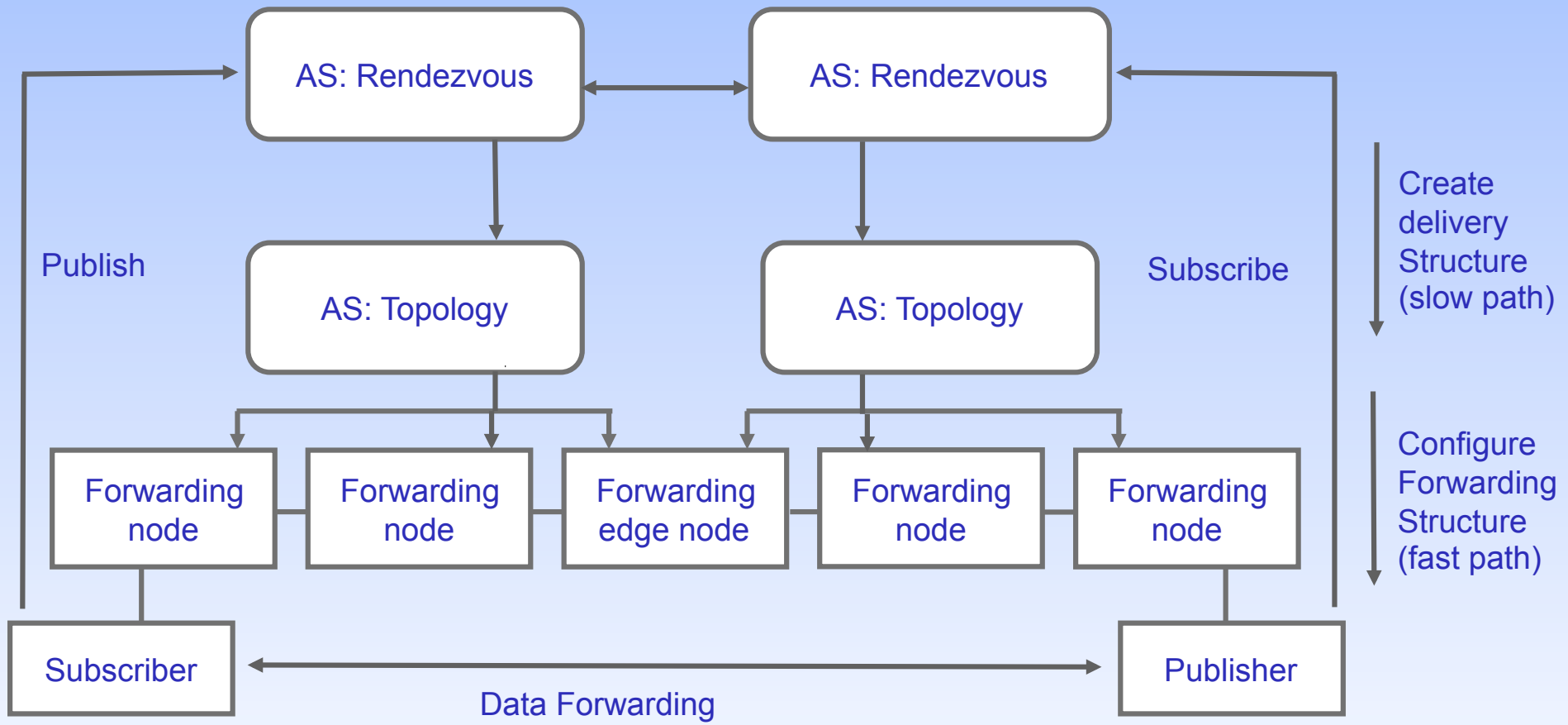
- Network endpoints are more and more interested in content
- Publish/subscribe is a candidate paradigm for content distribution
  - Many-to-many, control distributed between subscribers (receivers) and publishers (senders)
  - **Inherently receiver driven**
- Recent systems: CCN and PSIRP
  - CCN is based on hierarchical names and router caches, interest and data packets
  - PSIRP is based on rendezvous to labels that define forwarding paths (several mechanisms)

Background

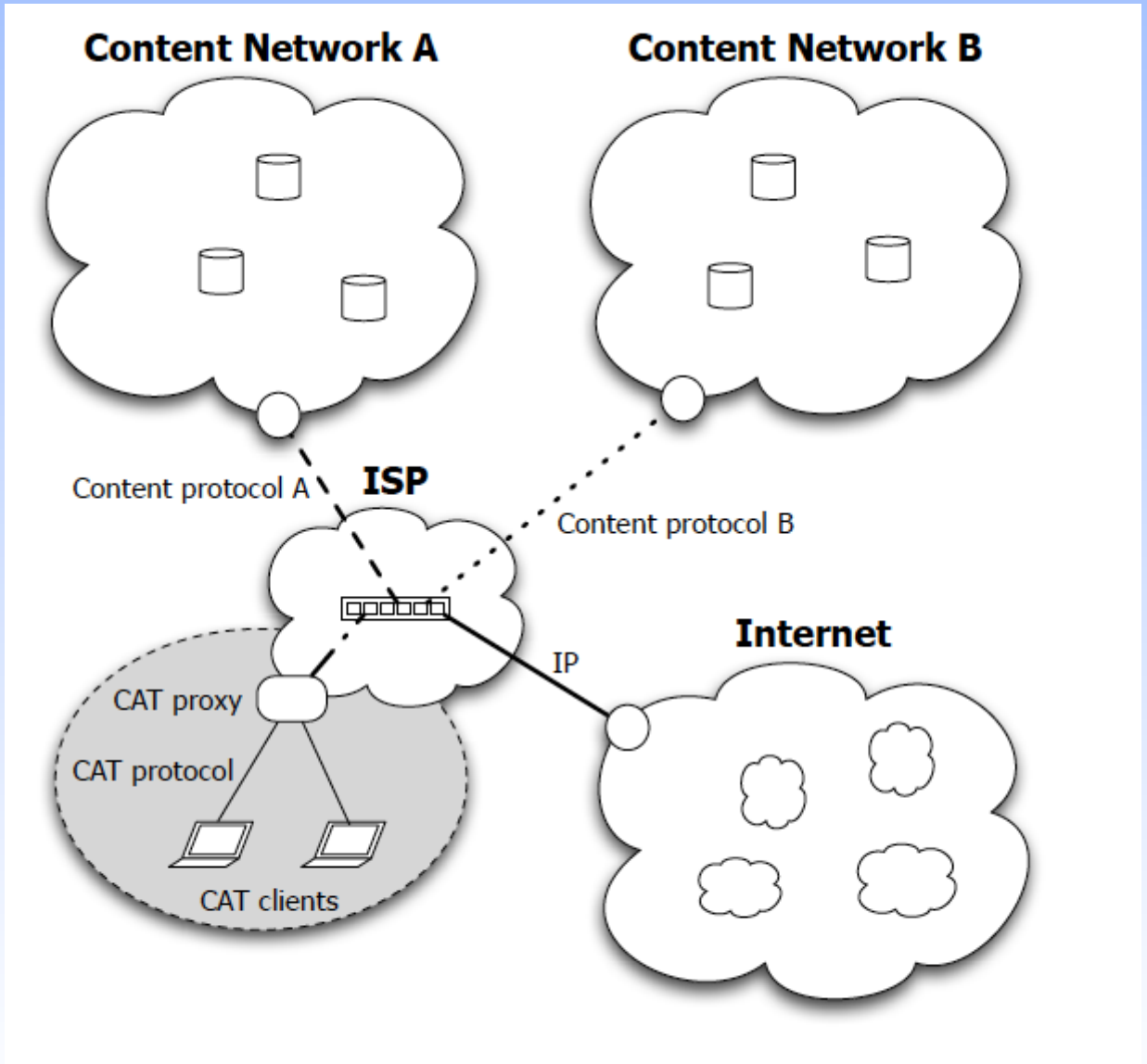
- Related systems include NIRA, DONA, ROFL, CCN, PSIRP, and PURSUIT
- PSIRP includes a 2-tier system where a hierarchical DHT based rendezvous interconnect network joins multiple rendezvous networks together for global reachability
  - Publications are identified by scopes and flat identifiers
  - Typically only scopes are advertised in the interconnect
  - Hierarchical structure guarantees locality for the communication
- Past work on SIP pub/sub gateway at HIIT

	Layer	Underlying transport	API	Namespace	Self-certification	Receiver driven	Key application
DONA	L4-L7	TCP/IP	Anycast	Flat	Yes	Yes, fetch operation	Content discovery and delivery
CCN	L3	Unreliable data transport, flow control	Name-based (with leases)	Hierarchical	Yes	Yes	Content delivery, voice
PSIRP	L2-L7, layerless	Ethernet, TCP/IP and PLA	Pub/sub and metadata-based	Various, recursive	Yes	Yes	Content delivery
Internet Indirection Infrastructu-re (i3)	L7	TCP/IP	Trigger-based	Flat	Yes	Yes (also sender driven)	Various
Haggle	Layerless	Packet-based	Metadata-based	ADU, user level names	Possible	Receiver/ sender	Mobile environment
Siena	L7	TCP/IP	Pub/sub	Content-based	No	Yes	Content delivery

PSIRP Architecture



System model



The CAT System

- In order to be able to leverage content-centric networks immediately, we assume that CAT is implemented on top of HTTP in a REST-like manner

- The CAT protocol has two parts:
  - Control part: subscribe/publish metadata
  - Data part: receive/fetch and send actual data

Namespace mapping

- Differing namespaces: flat, hierarchical, content-based
  - URLs, topics, interests, filters/content
- A mapping function is needed to bridge naming domains
  - Name/Hierarchical → Flat: easy
  - Content-based → Flat: difficult, possible
  - Flat → hierarchical: requires resolver
  - Flat → content-based: requires resolver
- Include various mappings in the metadata or develop/ deploy resolver
  - Starting point: URL (Web/CCN) and flat labels

Key Phases I/II

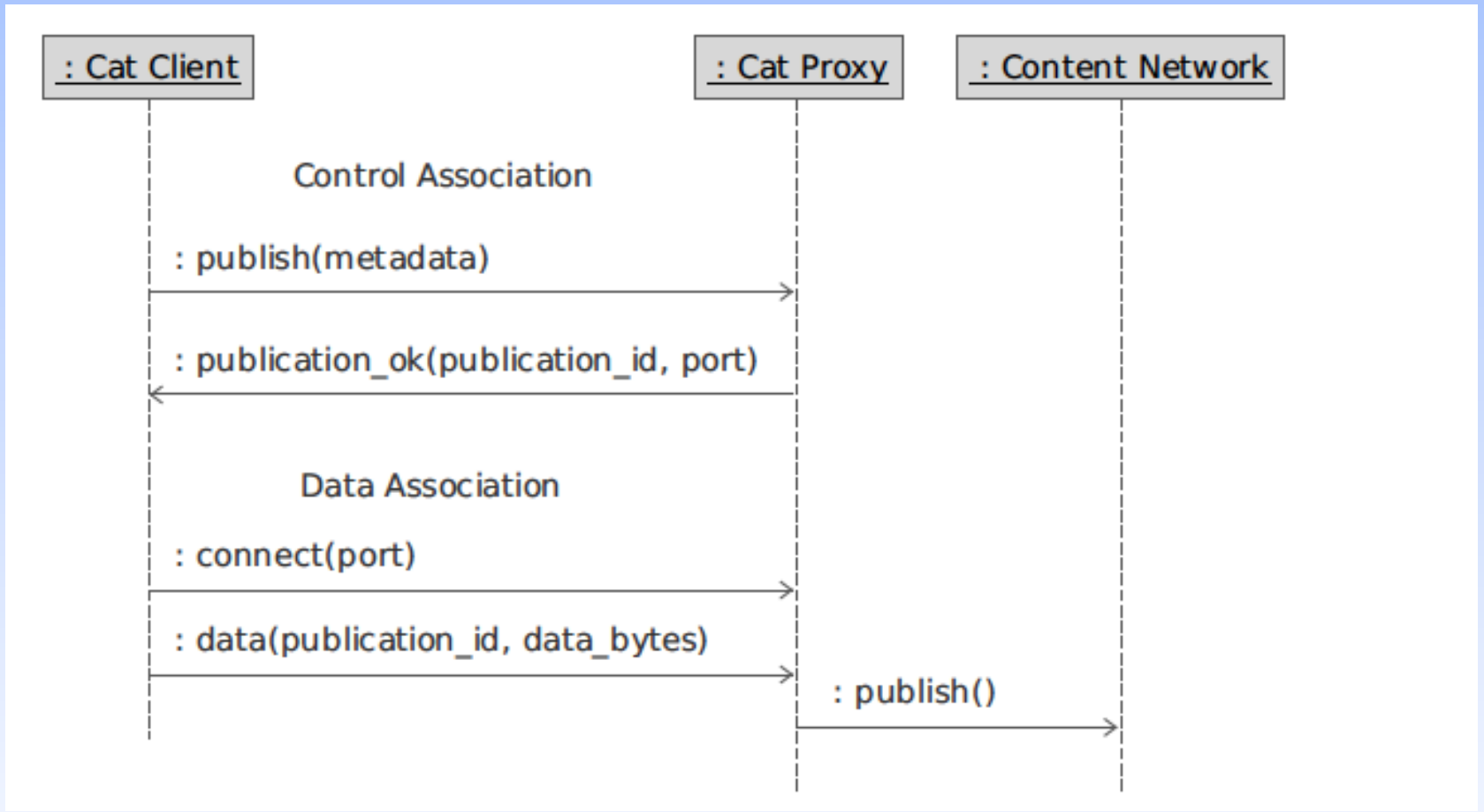
1. Local proxy discovery: **CAT proxy URL and DNS**, DHCP-like discovery, IPv6-like discovery

2. Connection establishment: Before publishing or subscribing content, the CAT client opens a single reliable transport connection (usually TCP or SCTP) that we refer to as **control association**, to a CAT proxy.

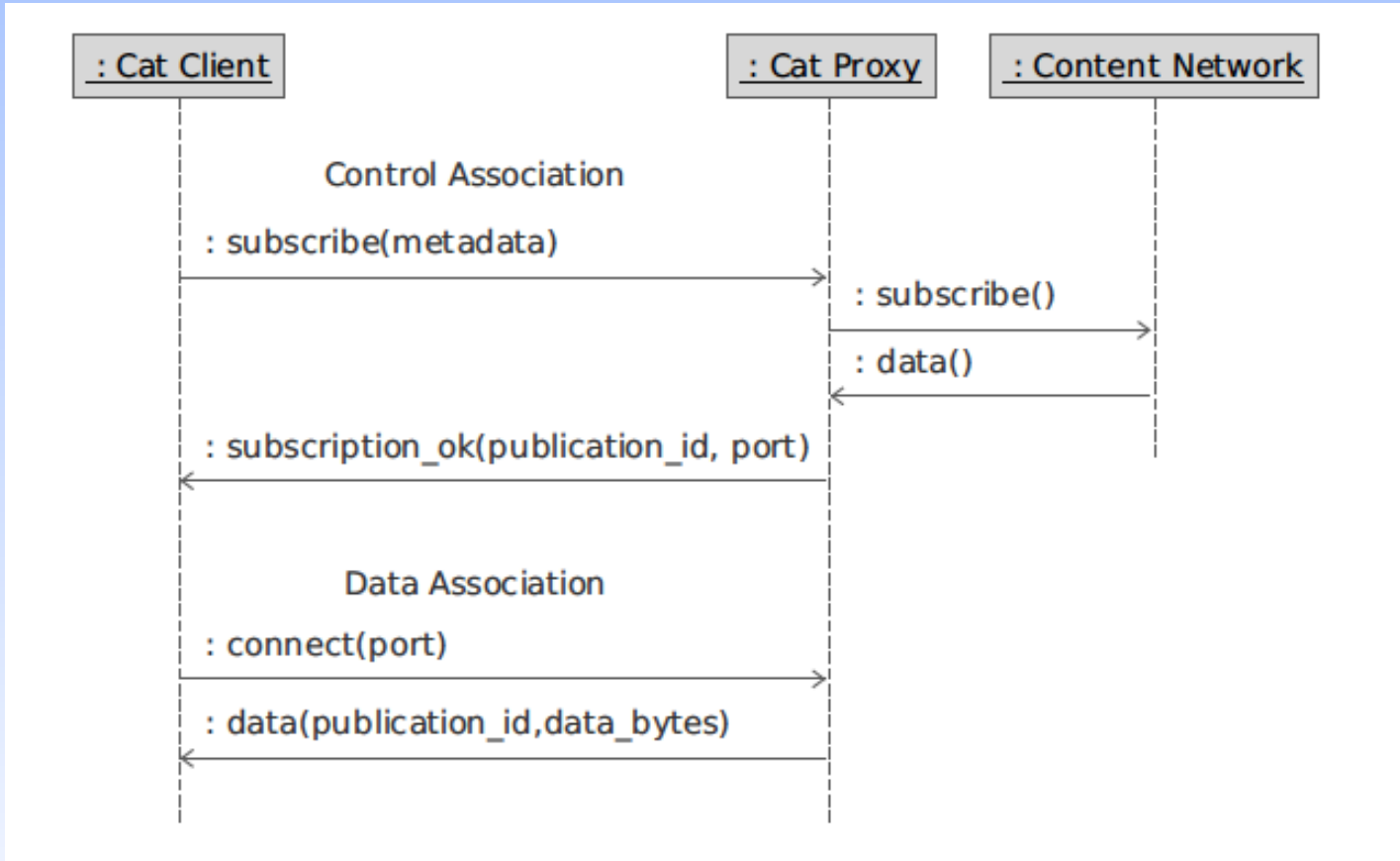
Key Phases II

3. Publishing: In order to **publish** a content item, the CAT client first sends metadata about the content item to be published to the CAT proxy using a HTTP POST request. Client can then publish the item through the CAT proxy.
4. Subscribing: In a similar fashion to the publication procedure, a client that wishes to **subscribe** specific content contacts CAT proxy and submits metadata about the data item to be subscribed to the proxy. Client will receive an update when the data is ready to be retrieved.
5. Continuous updates are notified through the control plane

Publishing


: Cat Client
: Cat Proxy
: Content Network
Control Association
: publish(metadata)
: publication_ok(publication_id, port)
Data Association
: connect(port)
: data(publication_id, data_bytes)
: publish()

Subscribing



Analysis

- One of the most important notions of content distribution networks is availability, responsiveness and resulting throughput of the system.

- The system aims to provide high availability for content by caching content locally, and utilizing a number of CN networks.

Parameters

- The key parameters for the proxy are the following:
  - The request rate l that determines the overall load of the CAT proxy, and also the load on the CNs
  - The cache hit probability p, which determines the request forwarding rate to the CNs by a proxy
  - The number of CN systems k
  - The fan-out parameter f of the requests, which determines the level of parallel processing in the network, and the overall load on the CNs

Simple Load Model

- Load of a CN system (k = 1)
  - (1-p)*l + A, where A is the request rate from other proxies to the CN (req/s)
- Load of a CN system (k >1)
  - (f(1-p)*l+A)/k (req/s)
- Goal: minimize client latency and load of CNs by caching at the proxy and utilizing parallel operations toward CNs

Content Updates

- The updates for the same content/topic/interest may be available from different CNs, which results in unnecessary delivery overhead unless content version checking is done.
- This can be done by first signaling about new content that matches subscriptions, and only after that transferring the content.
  - Taken into account in the control plane of CAT
  - Possible if content/data available signal
- Namely, a subscribe request is sent to f CNs. This requires that there is a mechanism for preventing redundant content updates from the CNs towards the CAT proxy.
  - CN specific implementation at CAT proxy

Selector Function

- CAT operations, namely subscribe and publish, can be implemented in different ways from the viewpoint of CNs
- Goal is to hide most of the details, some parameters are needed
- An optional **selector** function
  - Used by clients to define which CN networks are used for subscriptions and publications
  - Selector and mapping/resolver are necessary for bridging multiple CN technologies

Observations

- **Parallel access** to multiple CNs improves system performance and content availability; however, the proxy needs to coordinate updates to subscribed content.
- The proxy thus provides a **multi-source delivery** service for subscribers.
- The CAT proxy offers significant improvements when content can be cached. Recent experimental results indicate that a large portion of Internet content can be cached.
- A **selector function** enables clients and applications to specify which CNs should be used for which content thus providing a universal CN service. The selector function should specify CN policies and delivery options.

Conclusions

- A common last-mile transport protocol is crucial for the adoption of future content centric networks
- We have outlined the concept of content-aware public-subscribe protocol for content-centric networks
- The CAT proxy design supports various transport protocols; however, HTTP is the main focus of our work due to its interoperable nature
- The CAT proxy needs to support content selection and delivery across various content-centric networks, which means that the proxy is metadata-aware and able to support multi-source data delivery to subscribers.
- The key features are thus universal data access, content locality, and parallel access to content-centric networks

Notes

- CAT is a bridge/policy enforcement point towards heterogeneous CN networks.
- Implementation layer for CN is an open issue, CAN uses HTTP for interoperability. Different systems operate on differing layers.
- CAT does not solve mobility/multi-homing alone. Mobility problems are alleviated by the CN communication paradigms (pub/sub). Of course the last hop IP address will change.
- Observation: IP address is not used end-to-end, only at the last hop (similar way to a link layer address).