UNIVERSITY OF CAMBRIDGE

Node Design for an Information-Centric Network Architecture

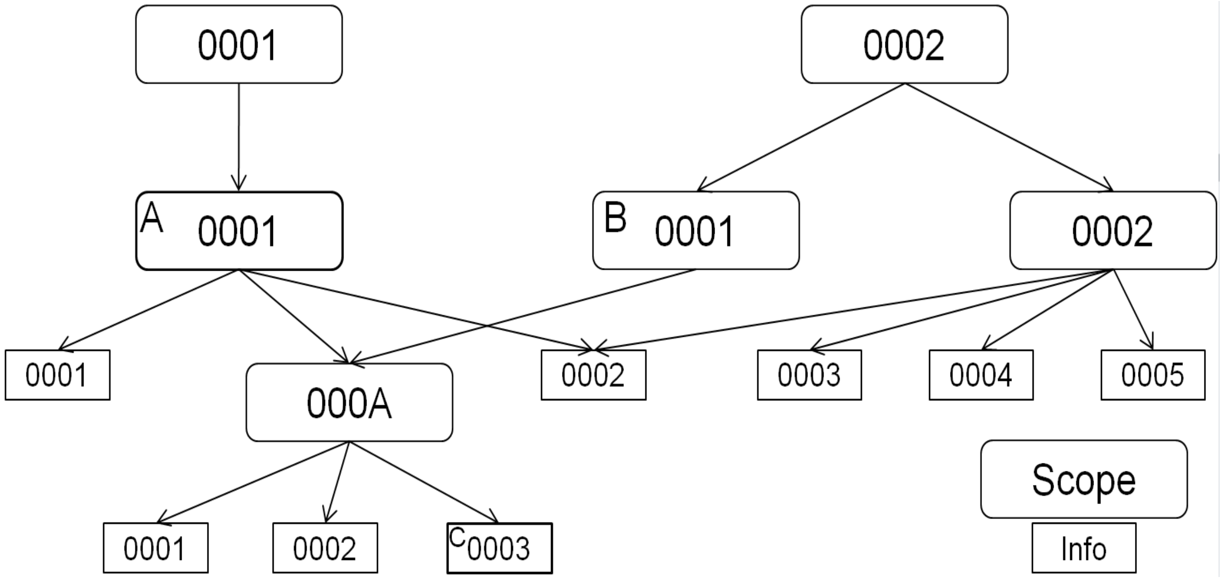
George Parisis, Computer Laboratory

Motivation and Contribution

- Information-Centric Networking (ICN) is increasingly attracting attention
  - PURSUIT (PSIRP), CCN (NDN), DONA, SAIL(netinf)
- How do network nodes look like in such an environment?
  - evolve the current IP-based design
  - clean-slate design
- Design and implementation of an information-centric network stack
  - nothing untouched with respect to the IP legacy
  - runs in parallel with TCP/IP
  - can be deployed in relatively large networks

Architectural context: information and scoping

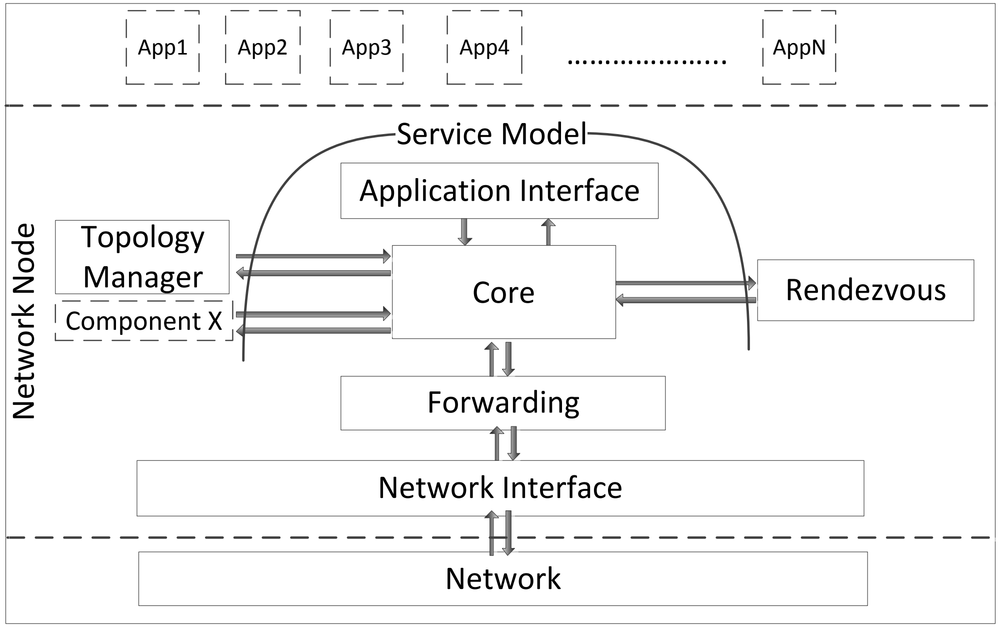
- Information labelling in contrast to end-point addressing
- Information aggregation through scoping



Architectural context

- Spatial and temporal decoupling of communicating parties
  - Native publish/subscribe network access
- Separation of core network functions
  - **Rendezvous**: matches demand for and supply of information
  - ***Topology management and formation:*** determines a suitable delivery relationship for the transfer of the information
  - **Forwarding:** executes information transfer
- Flexible information dissemination:
  - information scoping and well-defined dissemination strategies

Node Design

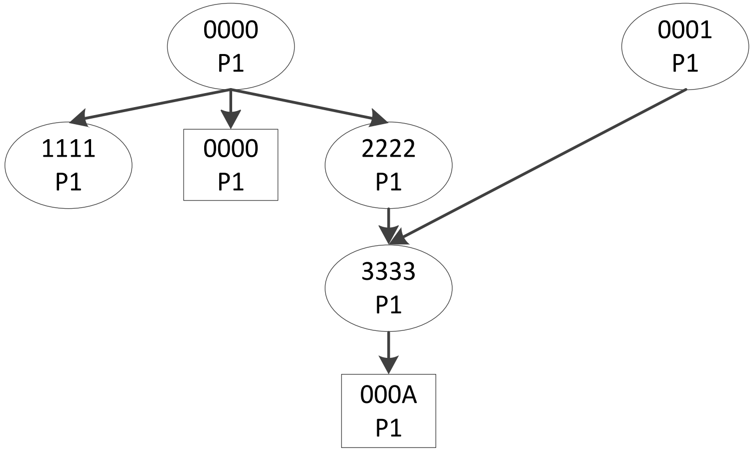


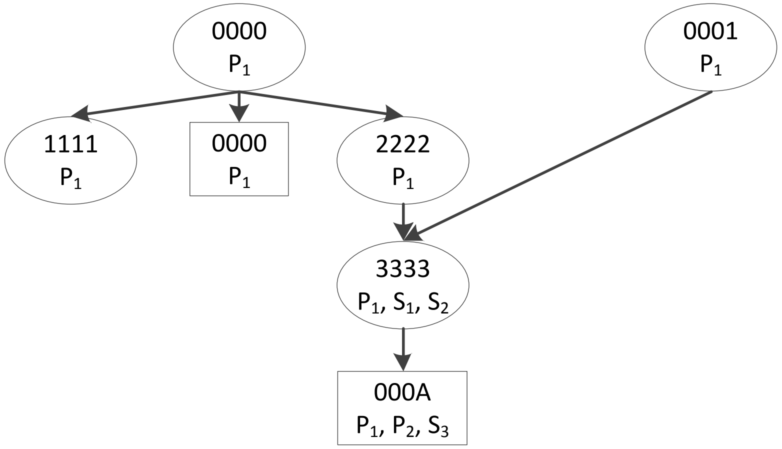
- Modularity of core functions
- A node can assume any role in the network
  - Forwarding, Forwarding+, RV, TM, end-nodes

Information Semantics

- Node design is agnostic
- Immutable data
  - Statistically unique labels from content
- Mutable Data
  - A single ID for many items (e.g. live video)
- Hybrid approach (channel-like with immutable objects)
  - Algorithmic identification

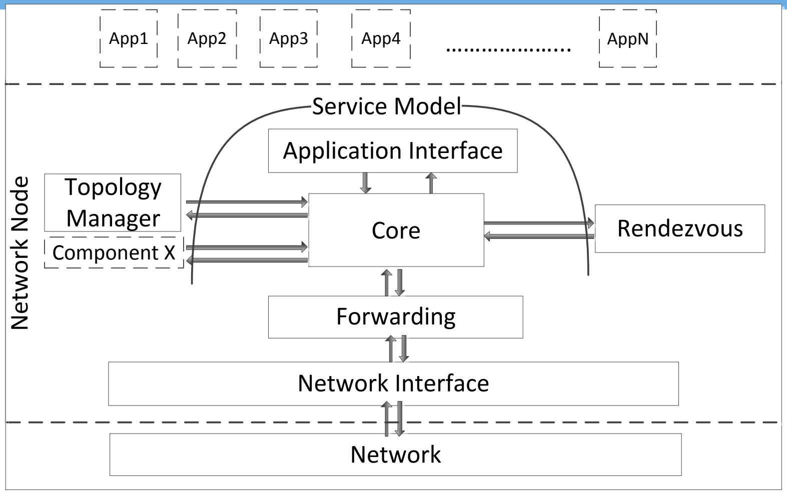
Service Model





- publish_scope, unpublish_scope, publish_info, unpublish_info
- subscribe_scope, subscribe_info, unsubscribe_scope, unsubscribe_info
- publish_data
- New Scope, Deleted Scope
- Start Publish, Stop Publish
- Published Data

Node Implementation

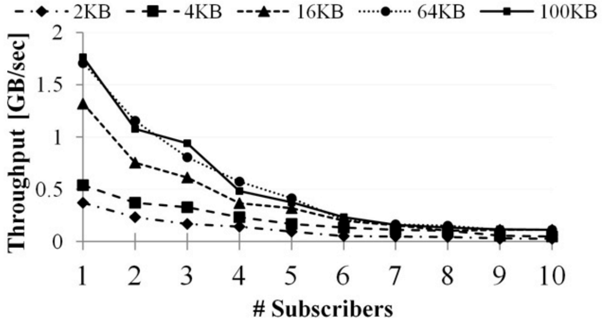


- Click Modular Router (User/kernel space, OpenWrt, Android, NS3)
- Components are implemented as Click Elements
- “System Calls”-like interface based on Netlink Sockets
- Network Interface implemented using Click Communication Elements
  - FromDevice, ToDevice, RawSocket
  - FromSimDevice, ToSimDevice

Implemented Dissemination Strategies

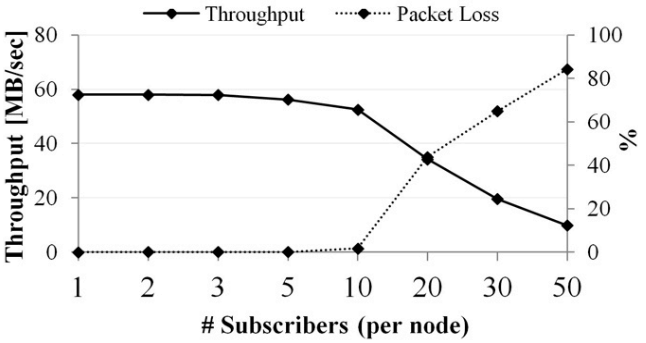
- Node-local
  - Dissemination within a single node – private information space
- Link-local and broadcast strategy
  - Dissemination across physical links
- Domain-local
  - Dedicated TM(s)
  - Dedicated RV(s) manages information space visible to domain
  - Core does housekeeping (pubs, subs, dispatches events, moves data to FW)
  - Forwarding based on LIPSIN identifiers
- Implicit Rendezvous

Experimental Evaluation



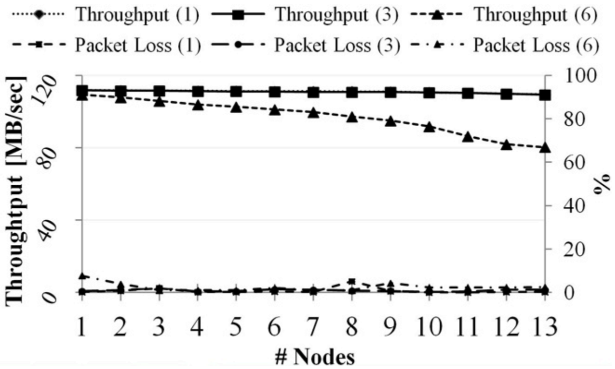
Inter-process Communication

- Several payload sizes (up to socket limit)
- Comparable to optimized TCP/IP (not shown)



Fast-path performance

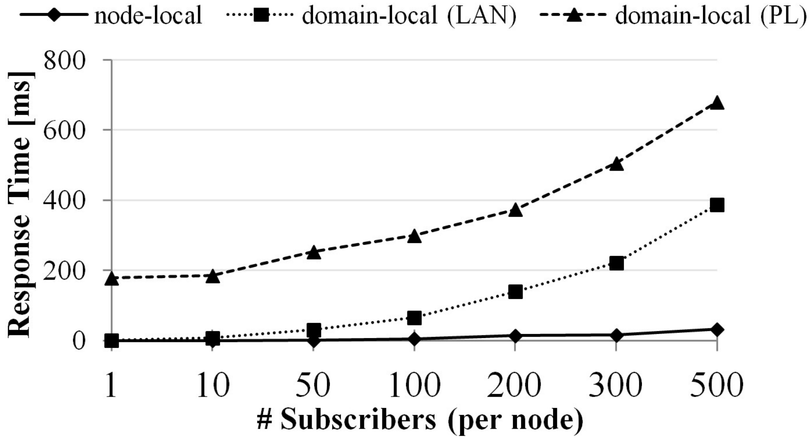
- 4 nodes star topology in a Gbit LAN
- No multicasting (~60MB/sec)
- >10 nodes - forwarding collapses



Forwarding efficiency

- 15 nodes in a chain
- Multicasting (when node is sub)
- ~line speed even when 3 subs per node for 13 nodes

Experimental Evaluation – Slow Path



100.000 advertisements under a single scope – Subscribers subscribe to a random item, wait until receive it and reiterate (500 times)

- Node-local
  - No net delays
  - No TM
  - 20ms for 500 processes

- Domain-local (Gbit LAN)
  - TM
  - ~400 ms for 500 processes per node (7000 subscribers)

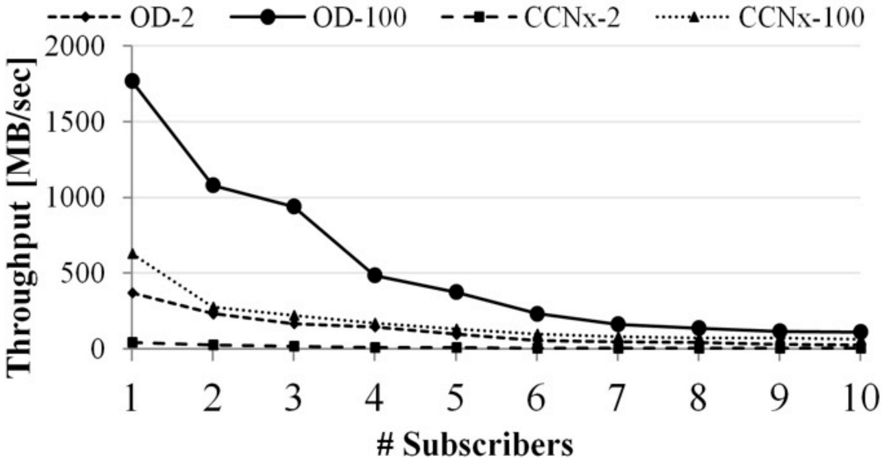
- Domain-local (PlanetLab)
  - Large delays
  - ~200ms for 1 sub per node (73 in total)
  - ~680ms for 36,500 subscribers

UNIVERSITY OF
CAMBRIDGE

Comparison with CCN(x)

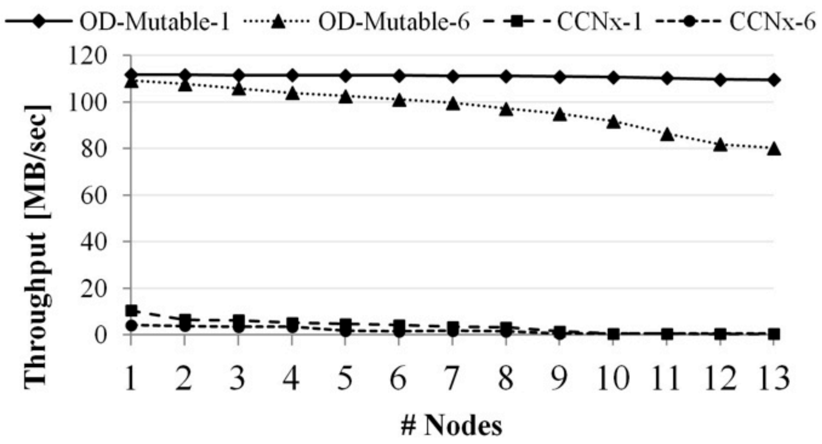
- Modularity of main functions
  - routing, forwarding, caching in the network – routing not specified yet
- Content security
  - packet signing is not an option in CCN
- Role of caching
  - CCN caches everything – many useless packets
- Supported information semantics
  - immutable data only
- Platform modularity and Deployment

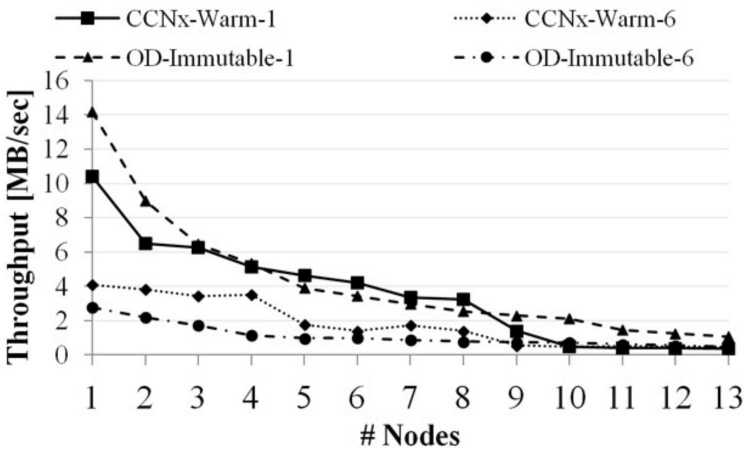
Comparison with CCN(x)



- **Node-local** (payload size: 2KB and 100KB)
    - CCNx application expresses interest for 10000 items (/content/segmentNumber)
    - CCNx replays all data from the local cache
    - Signing overhead gives performance of ~200KB/sec if each packet is signed on the fly

Comparison with CCN(x)





- Mutable data – 1Gbps LAN – 14 nodes chain
- CCNx is warm
- window of interests
- For a single node and a single publisher: 10.4MB/s (as in CCN paper)
- It degrades fast (0.38MB/s)

- Immutable data – 1Gbps LAN – 14 nodes chain
- window of interests/subscriptions for CCNx and our design
- RV for each subscription
- Similarly bad performance
- Extreme scenario for our design

References

- D. Trossen and G. Parisis. Designing and realizing an information-centric internet. Communications Magazine, IEEE, 50(7):60-67, July 2012.
- https://github.com/fp7-pursuit/blackadder

Thanks!