TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITY OF HELSINKI

Enhancing Compact Routing in CCN with Prefix Embedding and Topology-Aware Hashing

Stefanie Roos[1], Liang Wang[2], Thorsten Strufe[1], Jussi Kangasharju[2]

[1]TU Dresden, firstname.lastname@tu-dresden.de

[2]University of Helsinki, firstname.lastname@helsinki.fi

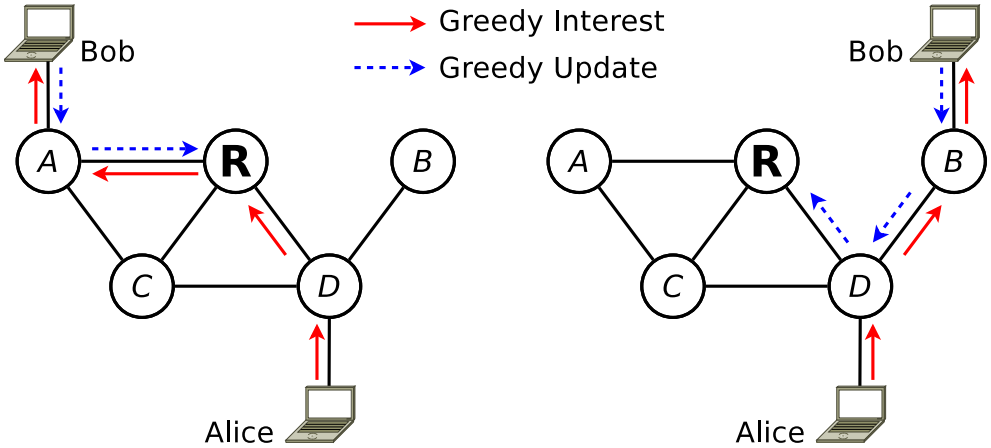
DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Motivation

- xCNs: new Internet architecture based on addressing content rather than locations (hosts)
- Goal: Improve content delievery
- Various challenges with regard to routing and content addressing
- Source mobility: One particular hard challenge

Greedy Embeddings

- Assign coordinates to fixed topology such that greedy routing works
- Handling mobility in CCNx [1]
  - Source s registers at closest host h
  - h forwards packets to s
  - If s moves, only h updates its information
- Benefits
  - Small routing table
  - Capability of hand-ling simultaneous handoff
  - Improved handoff delay and latency



[1] Wang, L., Waltari, O., & Kangasharju, J. (2013). Mobiccn: Mobility support with greedy routing

Prior Work: Limitations

- Limitations:
  - Embedding Hyperbolic space, combined with an naive content addressing algorithm (SHA1)
  - Traffic and storage load is highly imbalanced
  - Severe scalability issue when network becomes bigger

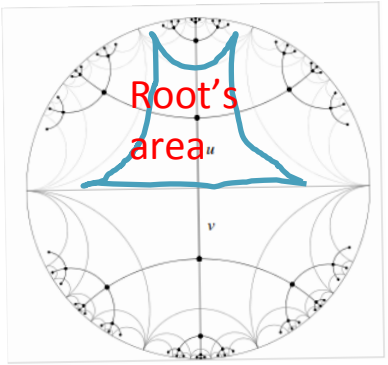


Image from R. Kleinberg: Geographic Routing using Hyperbolic Space, Infocom 2007

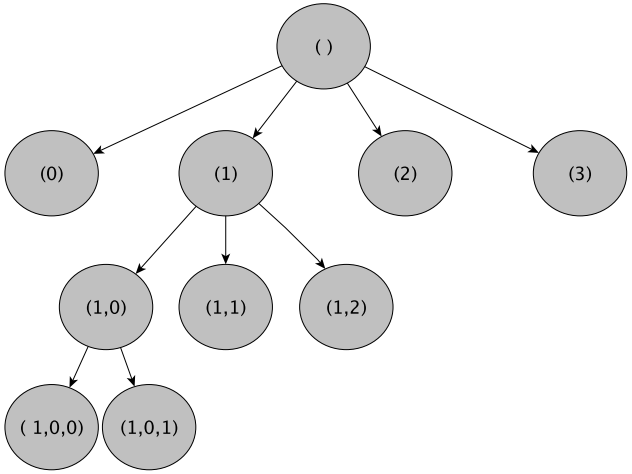
Our Contributions:

1. Changing the embedding algorithm
2. Changing the content key generation

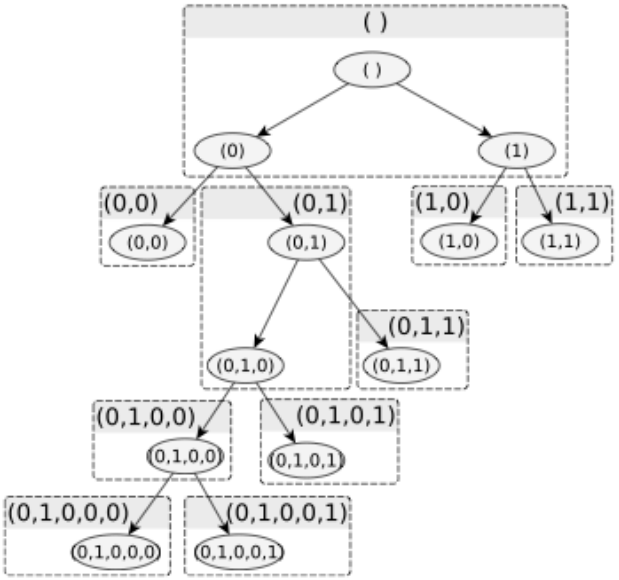
Prefix Embedding

- Prefix Embedding:Isometry of spanning tree
  1. Root has empty vector as ID
  2. Node with ID id enumerates children
  3. i-th child receives ID id||i
  4. Distance between nodes in tree

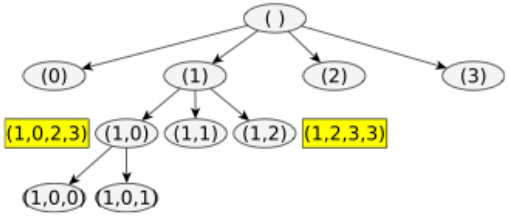
     dist(s,t,)=|s|+|t|-2commonprefixlength(s,t)



Prefix Embedding: Extensions



- Virtual binary trees for bit strings as IDs
- Routing modification for virtual trees:
  Forward to parent if not responsible but no closer neighbor



- Content is stored on node closest to its key
- Content keys are longer than IDs -> all content stored on leaves and nodes with only one child

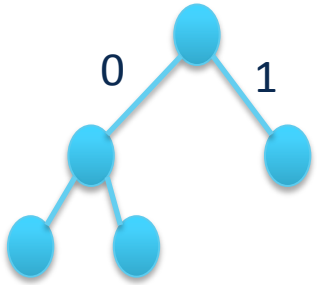
Prefix-S Embedding

- Store content on internal nodes
- Use two types of IDs: routing ID and storage ID
- Routing IDs are IDs received from parent
- Internal nodes with d children generated d +1 IDs, choose first one as their storage ID
- Leaf nodes use routing id for storage
- Greedy routing with slight modification is guaranteed to succeed

Topology-Aware Keys

- Nodes on higher levels of tree responsible for more files
- Integrate topology in keys of content
  - Consider hash function h
  - Cpl = common prefix length
  - For content f, h_i (f XOR i)
  - i-th bit of content key

$$b_{i+1} = \begin{cases} 0, \frac{h_{i+1}}{2^z} \leq \frac{|\{v \in V : cpl(ID(v), d_i||0) = i+1\}|}{|\{v \in V : cpl(ID(v), d) = i\}|} \\ 1, otherwise \end{cases}$$



b_1=0 with p=3/4

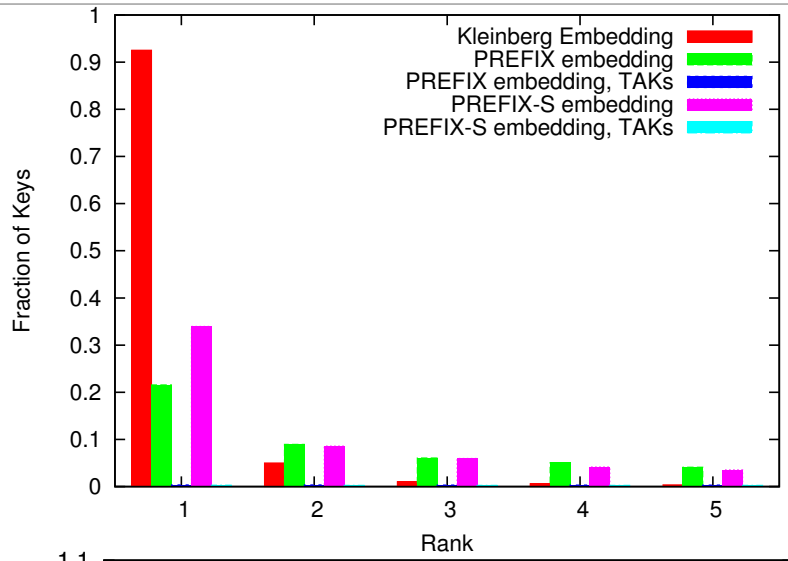
=> Uniform load

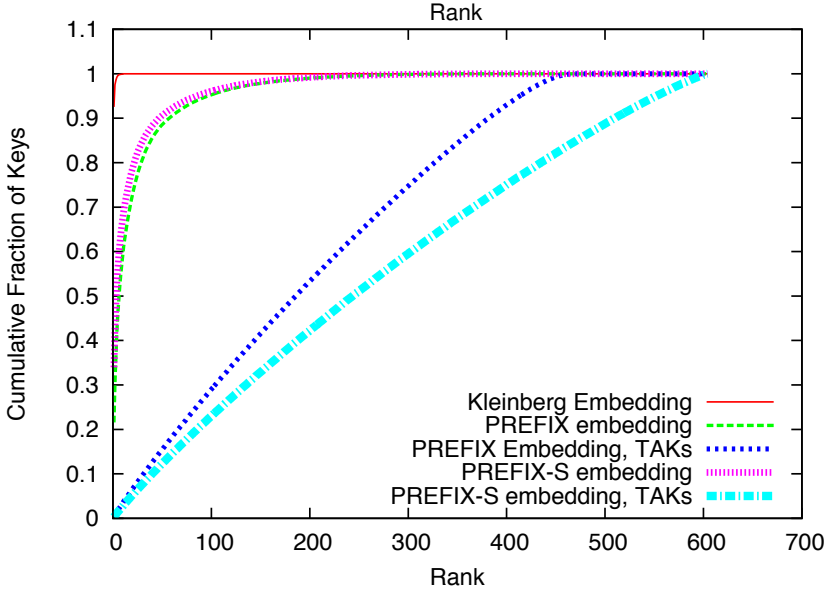
Evaluation

Evaluation (Static Simulation):

1. Generate random contents
2. Embed AS topology
3. Compute key of content and store
4. Execute queries for content
5. Metrics:
    - Fraction of content pieces per node
    - Fraction of queries forwarded per node
    - Routing hops

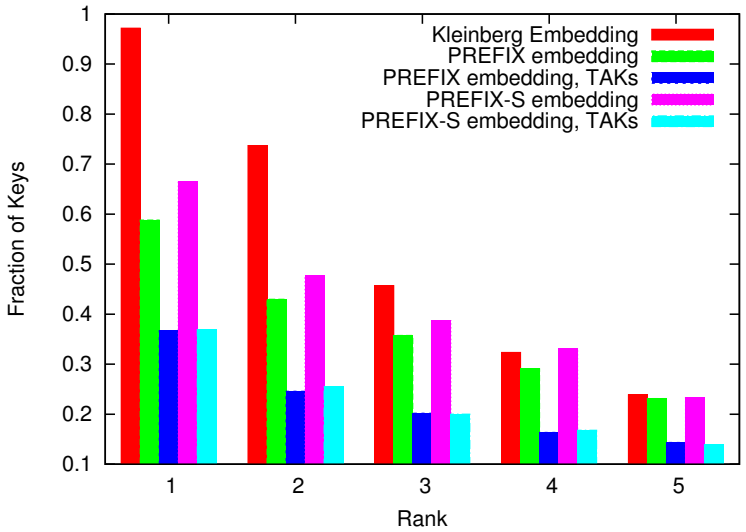
Storage Load

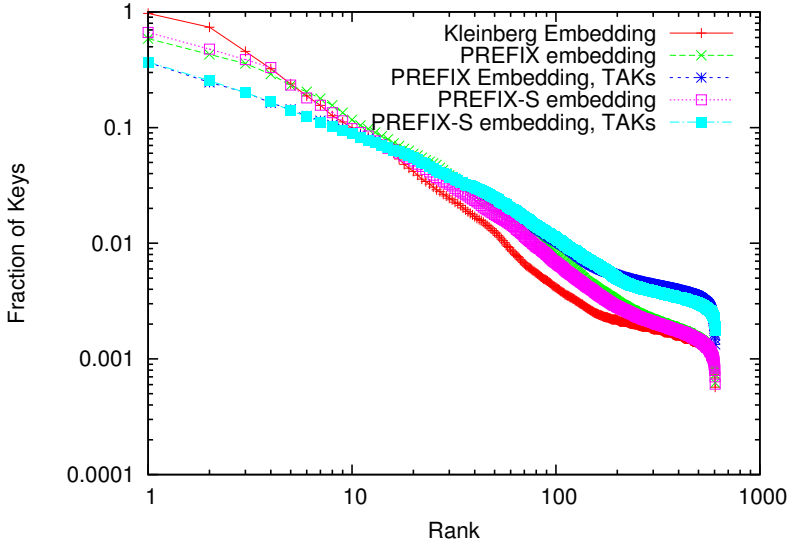




- Hyperbolic: more than 95 % content on 1 node
- Prefix Embedding: still unbalanced
- Topology-aware keys: uniform load (aka close to straight line)

Forwarding load + routing length





- Hyperbolic: close to 98 % of queries pass root
- Prefix Embedding: still around 70 %
- Topology-aware keys: not uniform but better balanced

- Routing length is increased from roughly 3-4 to 4-5 hops by using topology-aware keys

Conclusion

- Problem: Source mobility in xCNs
- Proposed Solution: Embeddings
- Improved the load balancing by
    1. Modifying embedding
    2. Topology-aware keys
- Can now prevent overload, single point-of-failure
- Future work: Evaluation in testbed to see the effect on actual congestion