αRoute: A Name Based Routing Scheme for Information Centric Networks


*David R. Cheriton School of Computer Science, University of Waterloo
+Orange Labs

Presented By: Shihabur R. Chowdhury
Outline

- Background
  - Information Centric Networks (ICN)
  - Challenges in ICN
- Contribution Summary
- \textit{\textalpha}Route DHT
  - Partitioning
  - Routing
  - Mapping
  - Content Lookup
- Conclusion
Information Centric Networking (ICN)

- Also known as “Content Centric or Content Based Networking”, “Named Data Networking” etc.
- Contents are communication endpoints rather than hosts
- Host to content binding is transparent to the end users

Why ICN?

- Internet usage is becoming more “content oriented” rather than “host oriented”
  - More video streaming traffic than ssh traffic
- Efficient content distribution is through ad-hoc patches
  - CDN, P2P file sharing etc.
  - Little knowledge about the underlying network
Related Works

- TRIAD proposed to avoid DNS lookup and use object names to route to object sources [2000]
- DONA improved on TRIAD and proposed a secure and hierarchical name based routing architecture [2007]
- Named Data Networking project at PARC initiated to develop a protocol specification for ICN [2009]
- A number of projects are working on different ICN architectures
  - PSIRP, 4WARD, SAIL, COMET, PURSUIT, NetInf, CONVERGENCE
Challenges in ICN

- **Content Naming**
  - How to uniquely and securely assign identifiers to contents?

- **Routing**
  - How to route content request based on content names?

- **Routing Scalability**
  - Routing table size
    - $O(n)$ is very expensive, $n \sim 10^{12}$ (even more).
    - Content names are hard to aggregate
  - Network traffic
    - How to efficiently serve content requests?
Contribution Summary

- We address the routing scalability issue in ICN
- We propose $\alpha$Route, a name based Distributed Hash Table (DHT) to route based on content names
- $\alpha$Route provides
  - Logarithmic routing table size and content lookup hops
- We also propose an algorithm for mapping $\alpha$Route to a physical network
Three important issues in a DHT design

- How to **partition** the name (or key) space among the DHT nodes?
- How to **route** a get or put query between the DHT nodes?
- How to **map** a logical DHT overlay topology to the underlying physical network?
\textbf{\textit{\textalpha{}Route: Partitioning}}

We treat the names as unordered set of alphanumeric characters

- `book1.pdf` => `{b, o, k, 1, p, d, f}`

We build a partitioning tree

- Each level takes partitioning decisions based on presence/absence of a subset of characters

The final partitions are mutually exclusive
αRoute: Partitioning (cont..)

- A subset of the alphabet, $S_i$, is assigned at each level $i$
- Example: Initially we have only one node and a partitioning set $S_1 = \{r, c\}$

\[ S_1 = \{r, c\} \]
**αRoute: Partitioning (cont..)**

- There are $2^{|S_i|}$ possible character presence combination at each node at level $i$.
- Each character presence combination may form the edges to nodes in level $i + 1$.
  - The root has at most $2^2 = 4$ children.
  - We assign another partitioning set, $S_2 = \{e\}$ to level 2 nodes.

\[ S_1 = \{r, c\} \]

\[ S_2 = \{e\} \]
Each node in level 2 has at most $2^1 = 2$ children
For $S_3 = \{k, t\}$, each node in level 2 will have at most $2^2 = 4$ children
And so on
αRoute: Partitioning (cont..)

- Leaf nodes are labeled with concatenation of all the labels on root to leaf path
- These concatenated labels represent a partition
- Labels of the leaf nodes are assigned to the DHT nodes

\[
S_1 = \{r, c\} \\
S_2 = \{e\} \\
S_3 = \{k, t\}
\]

Names that have c and k but not r, e and t

Diagram:
- \(\bar{r}c\)
- \(\bar{e}\)
- \(\bar{k}\)
- \(\bar{r}c\bar{e}\bar{k}\bar{t}\)
αRoute: Partitioning (cont..)

$S_1 = \{r, c\}$

$S_2 = \{e\}$

$S_3 = \{k, t\}$

Responsible for names
Matching the pattern $\bar{r}cekt$
αRoute: Routing

Requests `rocket.jpg`

`rocket.jpg` does not match pattern. Where to forward?
αRoute: Routing (cont..)

- Each node has a set of logical neighbors
- Neighbor list of a leaf node is determined by taking all possible character presence combination of each sub-label from root to node path

\[ S_1 = \{ r, c \} \]
\[ S_2 = \{ e \} \]
\[ S_3 = \{ k, t \} \]
**αRoute: Routing (cont..)**

- If a leaf node corresponding to a pattern does not exist then select the leaf node having longest matched prefix with the pattern’s representative string.

*S_1 = \{r, c\}*

*S_2 = \{e\}*

*S_3 = \{k, t\}*

- Logical node (no physical existence)
- Indexing node (DHT nodes)

| \(\bar{r}c - \bar{e} - k\bar{t}\) | \(rc - \bar{e} - k\bar{t}\) |
| \(\bar{r}c - \bar{e} - k\bar{t}\) | \(r\bar{c} - \bar{e} - k\bar{t}\) |
| \(\bar{r}c - \bar{e} - \bar{k}t\) | \(\bar{r}c - e - k\bar{t}\) |
| \(\bar{r}c - \bar{e} - \bar{k}t\) | \(\bar{r}c - \bar{e} - k\bar{t}\) |
| \(\bar{r}c - \bar{e} - k\bar{t}\) | \(\bar{r}c - \bar{e} - \bar{k}t\) |
αRoute: Routing (cont..)

$S_1 = \{r,c\}$

$S_2 = \{e\}$

$S_3 = \{k,t\}$

Logical node

Indexing node

Logical link

Physical link
\textbf{αRoute: Mapping}

- αRoute DHT nodes have almost equal number of logical neighbors.
  - i.e., overlay graph is regular
- Underlay graph is the Internet graph (AS level). It is reported to be power law distributed.
- Underlay graph nodes have tier ranking.
- Embedding a regular overlay graph on a power law distributed graph is hard.
αRoute: Mapping (cont..)

- Mapping Algorithm
  - Initiated by a central naming authority, similar to ICANN in current Internet naming.
  - The partition tree, T is initially grown based on some corpus.
    - The partitioning sets at each level are selected based on character frequency in the corpus.
  - The central authority assigns partitions to Tier-I ASs only.
\( \alpha \text{Route} : \) Mapping (cont..)

- Initially the tree is grown to support the number of Tier-I ASs only.
- Partitions are assigned to Tier-I ASs along with possible next levels of extensions.
**αRoute: Mapping (cont..)**

- Tier-I ASs extend their partition with additional levels in the tree
- The extended partitions are assigned to Tier-II AS.

```
Tier-1

Tier-2

Tier-I

Tier-II

Tier-III
```

- **Logical node**
- **Indexing node**
\textbf{\textit{\textcolor{red}{$\alpha$Route}}: Mapping}

- Conflict Resolution

\[ \overline{rc} - \overline{e} - k\overline{t} \]

\[ \overline{rc} - \overline{e} - kt \]

\[ rc - e - k\overline{t} \]
**αRoute: Content Lookup**

A node, \( n \), receives a content request.

The content name is transformed to matching pattern, \( p \).

\[ p = rcekt \]

\( n \) looks up in routing table to find a pattern \( q \) that has longest prefix match with \( p \).

Content is in \( r \).

\( n \) forwards \( p \) to a node \( m \), responsible for pattern \( q \). Forwarding continues until destination is found or

Request is redirected to the content’s actual location.

\( m \) contains an index, indicating the content’s actual location.
Routing in the Internet based on content name is challenging due to the large volume of contents.

Proposed \(\alpha\text{Route}\), a name based DHT that can route using content names.

\(\alpha\text{Route}\) provides guaranteed content lookup using logarithmic size routing table.

Also proposed a mapping algorithm that maps the DHT to physical network and assigns loads to network elements proportionally to their capacity.
Questions?
$S_1 = \{r, c\}$

$S_2 = \{e\}$

$S_3 = \{k, t\}$

**Diagram:**
- Logical node (no physical existence)
- Indexing node (DHT nodes)