MobiCCN: Mobility Support with Greedy Routing in Content-Centric Networks

by Liang Wang, Otto Waltari, and Jussi Kangasharju

Daiki Kaneko
Park Lab., Waseda Univ.
Thursday, July 17, 2014
Outline

• Introduction

• Greedy Routing

• MobiCCN Architecture

• Example – VoIP Scenario

• Evaluation
  – Evaluation about Handoff Delay
  – Evaluation about Scalability

• Conclusion

• References
Introduction

• Content-Centric Networking shifts the current Internet paradigm from point-to-point communication to content-centric dissemination

• Naming poses significant challenges
  – The number of content name prefixes is larger than that of domain names
  – Hierarchical names don’t improve the performance when the names are not related with the underlying network topology

• Publisher mobility is difficult to achieve
Introduction

- Greedy routing
  - First applied to mobile ad-hoc network
  - Can be also used in wired network

- In this paper, the authors introduce a greedy routing into CCN architecture → MobiCCN
  - Provides an efficient mechanism for seamless mobility
  - Solves the gap between huge space of names and scarce routers’ resources
Greedy Routing

- Greedy routing enables nodes to route packets by assigning coordinates to nodes.

- The underlying metric space is not specified.
  - Actual geographical coordinates may be used (→ geographical routing).

- The destination’s coordinate is embedded into the packet header.

- A node calculates the distance between the destination and each of its neighbors, and selects the one closest to the destination as the next hop.
Greedy Routing

• Geographical coordinates have local minimum issue
  – A node itself is closer to the destination than any of its neighbors but the node and the destination is not directly connected each other
  – The node cannot decide the next hop

• If we use a hyperbolic space, a node can always find a neighbor closer to the destination than itself [2]
MobiCCN Architecture

- Standard CCN protocol and greedy protocol
- A greedy packet is just a normal CCN Interest
- “greedy:/” for greedy protocol, “ccnx:/” for standard one
- The format of the name of a greedy packet
  
  \[ \text{greedy:/vc/operation/parameters/…} \]

- Name examples
  
  greedy:/324532234925526/voip/ring …
  greedy:/854267864477975/publish/data/17 …
  greedy:/548865564345699/update/signature …
MobiCCN Architecture

- Each router is assigned a vc (virtual coordinate) from the underlying hyperbolic space $\mathbb{H}$

**Fig. 1** a 3-tree embedded into the hyperbolic space in the Poincaré model

**Fig. 2** a 4-tree embedded into the hyperbolic space in the Poincaré model
Algorithm 1 Greedy routing protocol - Interest

Input: Greedy Interest GI
Output: Forward decision
dst ← destination coordinate of GI
if (dst, face) in FIB then
    oface ← face
else
    for each directly connected neighbor $N_i$ do
        $d_i$ ← distance between dst and $N_i$
    end for
    oface ← $N_i$ with smallest $d_i$
end if
Update (dst, oface) pair in FIB
Forward GI to oface

- Each router maintains a small table of its neighbors’ coordinates
- A router calculates the distance between the destination and each of its neighbors
- When a greedy packet arrives, a router checks FIB first to reduce computational time
MobiCCN Architecture

• Each user has a unique ID to identify himself

• Greedy routing maps the user ID into the same hyperbolic space $\mathbb{H}$ to obtain its virtual coordinate

• Then, each user has a dedicated router who is closest to him in $\mathbb{H}$ as his host router
MobiCCN Architecture

**Algorithm 2**  Greedy routing protocol - Update

- **Input:** Greedy Update GU
- **Output:** Update FIB of the routers between a mobile user and its corresponding rendezvous point

iface ← ingress of GU
dst ← destination coordinate of GU
Update (dst, iface) pair in FIB

for each directly connected neighbor \( N_i \) do
  \( d_i \) ← distance between dst and \( N_i \)
end for

oface ← \( N_i \) with smallest \( d_i \)
d ← destination between dst and current router

if \( d < \min(d_i) \) then
  Rendezvous point, stop forwarding
else
  Forward GU to oface
end if

- When a data source moves to a new attachment point, it sends out a greedy Update packet to its host router.

- The name of an Update packet

  \[ \text{greedy:/vc/update/} \]

- Each router the Update passes by will update its FIB.
MobiCCN Architecture

• It is required that every greedy Update packet is signed for security reasons
  – Technically, this means Interest is signed, not Data
  – A signature can be appended to a content name, so no problem
Example – VoIP Scenario

Fig. 3 MobiCCN VoIP scenario

1. Alice sends Interest (ccnx:/domain/voip/bob) but Bob is not home
2. Bob has already send greedy Update packet
3. After timeout event, Alice sends greedy:/vc_{bob}/voip/ring and it finds Bob
4. Bob moves from A to B and sends greedy Update
5. When Alice sends greedy Interest, it doesn't go to R because D's FIB knows Bob is toward B
Evaluation

• Environment
  – Four real-world ISPs network topologies
    • Exodus, Sprint, AT&T and NTT
  – 240 Dell PowerEdge M610 nodes
    • Each has 2 quad-core CPUs and 32 GB memory, and is connected to a 10-Gbit network
    • Ubuntu SMP with 2.6.32 kernel
    • Multiple virtual routers in a physical node

<table>
<thead>
<tr>
<th>Network</th>
<th>Routers</th>
<th>Links</th>
<th>POPs</th>
<th>Diameter</th>
<th>Avg. Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exodus</td>
<td>338</td>
<td>800</td>
<td>23</td>
<td>12</td>
<td>5.824</td>
</tr>
<tr>
<td>Sprint</td>
<td>547</td>
<td>1600</td>
<td>43</td>
<td>12</td>
<td>5.182</td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>733</td>
<td>2300</td>
<td>108</td>
<td>11</td>
<td>6.043</td>
</tr>
<tr>
<td>NTT</td>
<td>1018</td>
<td>2300</td>
<td>121</td>
<td>14</td>
<td>6.203</td>
</tr>
</tbody>
</table>

TABLE I: Graph properties of the four selected ISP networks
Evaluation about Handoff Delay

• Results are topology-independent, they only show the results on AT&T network
• Interest Forwarding has been shown to be superior to the others [], they only compare against it

• Parameters
  – Link delay: 5 ms
  – The initial placement of the sender and receiver is arbitrary
  – Mobile sender moves within 2 hops
  – L2 handoff delay: 100 ms
  – Loss detection timer: 100 ms
  – Both caller and callee perform a simultaneous handoff at 10 sec
  – Caller and callee send out Interests at 50 packets/sec
**Fig. 4** MobiCCN handoff delay

**Fig. 5** Interest Forwarding [3] handoff delay
Evaluation about Scalability

- Callee is fixed and caller moves
- They measured the path stretch between caller and callee
- The figure shows the average result

Fig. 6  Average stretch as a function of number of handoffs
Conclusion

• By embedding network topology into hyperbolic space, the authors distribute the rendezvous points and name resolution functionality into the network.

• They compared MobiCCN to other proposed CCN mobility schemes and showed its advantages.

• They used their own prototype to evaluate, and they are now implementing MobiCCN in CCNx.
References

