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

Liang Wang, Otto Waltari, Jussi Kangasharju

Department of Computer Science, University of Helsinki, Finland

Background of Information-Centric Network

Our current Internet was originally designed for point-to-point communication. However,

- Nowadays, Internet is mostly used for content distribution.
- Large amount of multimedia files emerges everyday.
- Users want the content to be delivered fast and efficient.
- Users want the content to be safe and authenticated.
- But users do not care where the content is from.

A Clean-Slate Solution to the Current Internet

Current Internet design is clumsy and inefficient when it is confronted with the novel applications for content dissemination.

- Information-Centric Network (ICN) was proposed to get around the issues (i.e. efficiency, security and simplicity)
- A clean-slate redesign of the current Internet. Several independent proposals: NDN, DONA, NetInf and PSIRP.
- The core idea of the different proposals is essentially the same -- accessing content by name; universal caching.
- We focus on Content-Centric Network (CCNx) in discussion.

How Does CCNx Work?

CCNx implementation complies with general design of ICN.

- Content is requested by hierarchical name.
- User sends out an Interest packet for the requested content.
- Content name is embedded in the header of an Interest.
- Three key data structures:
  - Pending Interest Table (PIT)
  - Forwarding Information Base (FIB)
  - Content Store (CS)

Is CCN a Perfect Solution?

CCN tries to solve many existing issues like congestion and security, but it is not a silver bullet.

- CCN's design is receiver-driven, so it inherently solves receiver's mobility.
- The lost packet can be recovered by universal caching. Technically, by simply retransmitting the lost Interest.
- However, if the data source is mobile, necessary name operations are needed in the network.
- CCN fails to give a satisfying solution -- updating and propagating names are expensive operations in network.

Why Solving Data Source Mobility is Important?

Solving data source mobility is NOT as trivial as it seems, it also leads to the solutions to other related issues.

- Real-time content publication and dissemination.
- Mobile content publication and dissemination.
- Adoption of connection-based communications.
- Disparity between enormous space of application names and scarce of routers' resources.

Greedy Routing as a Solution to Mobility Issues

In this work, we use Greedy Routing to solve mobility issues in CCN. Our contributions are

- We show that greedy routing can be implemented as routing policy in CCN with minimum modification to the existing routing protocol.
- We present MobiCCN, our mobility scheme, and evaluate it thoroughly in realistic settings.
- We compare MobiCCN with other schemes from literature, and show that it outperforms them.

What is Greedy Routing?

Greedy routing has a long history in mobile ad-hoc networks.

- In such networks, a node does not have global knowledge of the network topology and only knows its neighbors.
- Greedy routing makes it possible to route in the "dark".
- In greed routing:
  - Nodes are assigned virtual coordinates from a metric space.
  - Destination coordinate is embedded in the packet header.
  - Packets are routed to the neighbour closest to the destination.
  - Implemented as an underlay instead of an overleay.

Greedy Routing Relies on Graph Embedding.

Greedy routing heavily relies on graph embedding.

- To assign a coordinate to each node, we first need to embed the network topology in a metric space.
- The embedding can be done on the fly. E.g. nodes in mobile ad-hoc network use their actual geographical coordinates.
- Essentially, using geo-coordinates indicates using Euclidean space as underlying metric space.
- Cheap solution but suffers from "local minimum issue":
  - A node x itself is closer to the destination y than any of its neighbors even though y is not x's directly connected neighbor.
  - Therefore, using Euclidean space cannot guarantee 100% delivery.

Find a Greedy Embedding for Greedy Routing

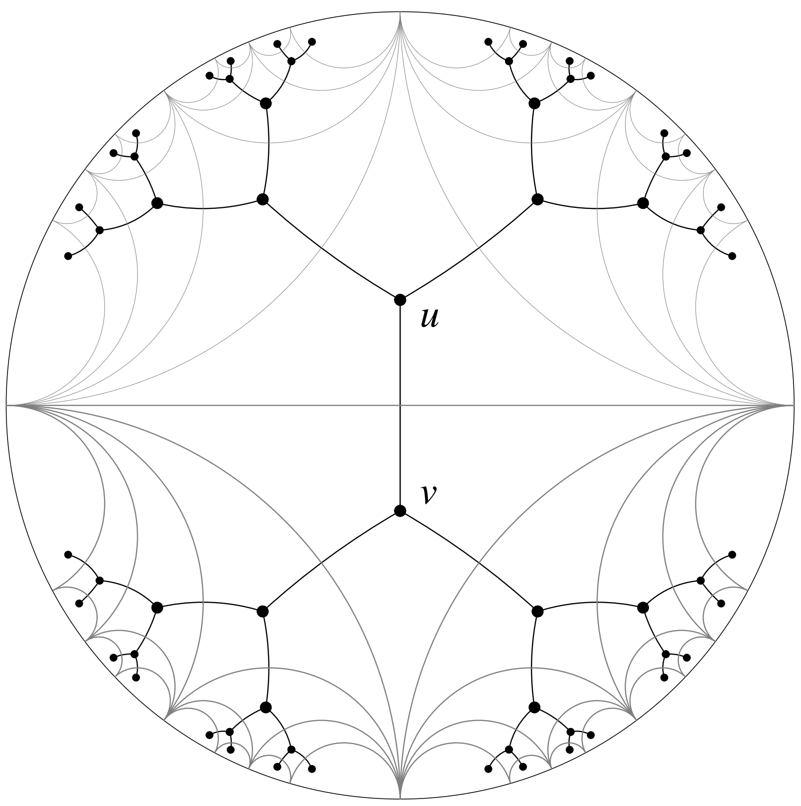
However, greedy routing does NOT necessarily use Euclidean space as underlying metric space.

- Any well-defined metric space works.
- To avoid local minimum issue, we need to find a graph embedding such that:
  - Given any destination which is not directly connected, a node can always find a directly connected neighbor who is closer than himself.
- The graph embedding with such property is called Greedy Embedding.

Use Hyperbolic Space

- Kleinberg proved in [1] that there exists a greedy embedding for arbitrary topology, given the underlying metric space is hyperbolic.

- **Poincaré disc** is a model for hyperbolic space. Right figure illustrates a 3-tree embedded into the hyperbolic space.



[1] R. Kleinberg. Geographic routing using hyperbolic space. In IEEE INFOCOM, pages 1902 –1909, may 2007.

MobiCCN Design

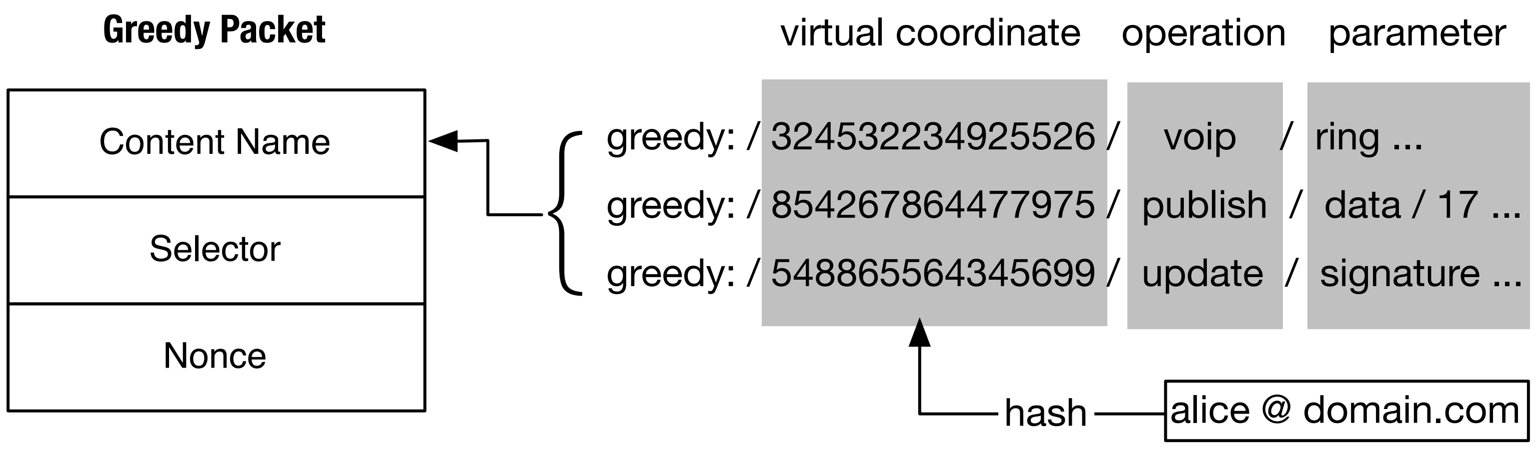
Pre-Embedding of a Network Topology

- To embed a network, we first get a spanning tree of the network.
- But the spanning tree of a non-trivial topology is not unique, which one shall we use?
- Different spanning tree leads to different stretch, choose the one of small stretch.
- In our work, we use Maximum-Weight Spanning Tree (MWST) algorithm, which tries to embed the node with high betweenness centrality close to the center of the disc.
- Node’s ID in hyperbolic space is a tuple (x,y). It is generated by hashing a node’s name, then splitting it into two parts.

Greedy Routing Protocol – Message Format

There are two routing protocols in MobiCCN,

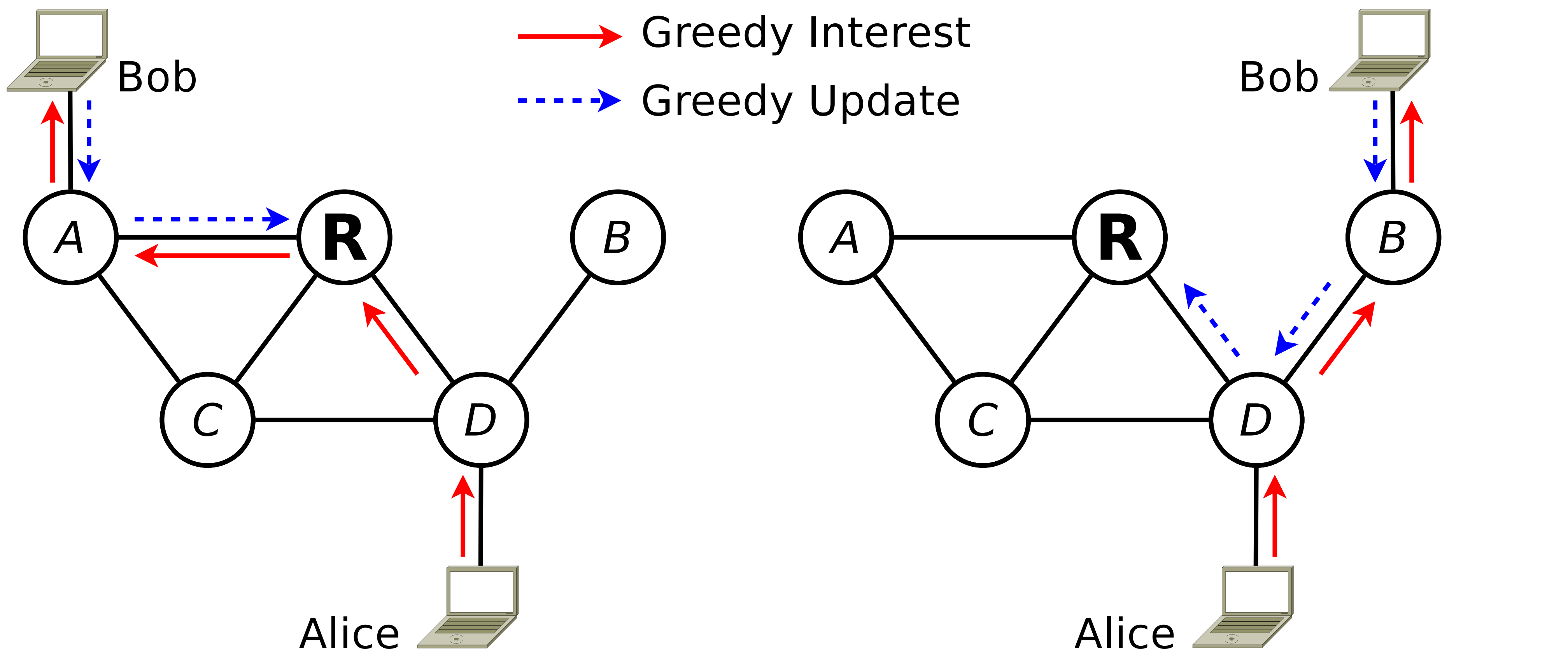
- The standard CCN protocol uses prefix ccnx:/.
- The greedy protocol reserves prefix greedy:/.
- When to activate greedy routing protocol?
  - Timeout event if standard protocol fails.
  - Sending out parallel Interests during the initialization phase.
- The general format of greedy messages:



Greedy Routing Protocol – Operations

- Greedy routing protocol is activated whenever a message with the prefix “greedy:/” arrives.
  - First check whether there is an entry in FIB using the longest prefix matching.
  - If the result is positive, it means the distance has been calculated before, and packet is forwarded to the next hop stored in FIB.
  - Each user has a dedicated router who is closest to him in hyperbolic space as his host router in the network.
  - The host router serves as rendezvous point and relays traffic for him.
  - If a data source moves to a new access point, it sends out an Update packet to its host router. The Update has a name like greedy:/vc/update, indicating it is an update operation.
  - Each router the Update passes by will update the corresponding entries of that data source in its FIB accordingly; then Interests towards the source can be forwarded correctly to the new domain.

VoIP as an Example



- Alice (Caller) and Bob (Callee) use their email addresses as their unique IDs. R is Bob's host.
- When Bob is the data source and he moves into a new domain B, he sends an update Interest with the name greedy:/vc_bob/update after the handoff.
- Alice's Interest reaches D before R, and D already updated its FIB from Bob's greedy Update.
- From both perspective, they always use the same names for communications.

Security in MobiCCN

CCN is built on the notion of content-based security. Each piece of content can be authenticated by the digital signature embedded in the packet header.

- MobiCCN is inherently able to prevent malicious users from exploiting Update packets to disturb the normal communication.
- The sender is required to sign every Update packet.
- Whenever an Update packet arrives, the router needs to check whether the sender is the actual owner of the name so that he has the right to update his corresponding entries in FIB.
- The signature can either be appended to the content name (as MobiCCN does), or stored in the additional field in the header.

MobiCCN Eveluation

Evaluation Standards

We performed thorough evaluation on MobiCCN.

- **Performance**: Achieve both low average latency and low handoff delay.
- **Compatibility**: Coexist with standard CCNx routing protocol.
- **Complexity**: Minimum modification to the current CCNx architecture.
- **Flexibility**: Handle simultaneous handoffs of both sender and receiver.
- **Scalability**: Handle continuous handoffs.

- With other proposed schemes in the literature:
  - **MC**: MobiCCN; **IF**: Interest Forwarding; **RP**: Rendezvous Point; **IP**: Indirection Point; **SD**: Sender-Driven Msg

Evaluation Setup

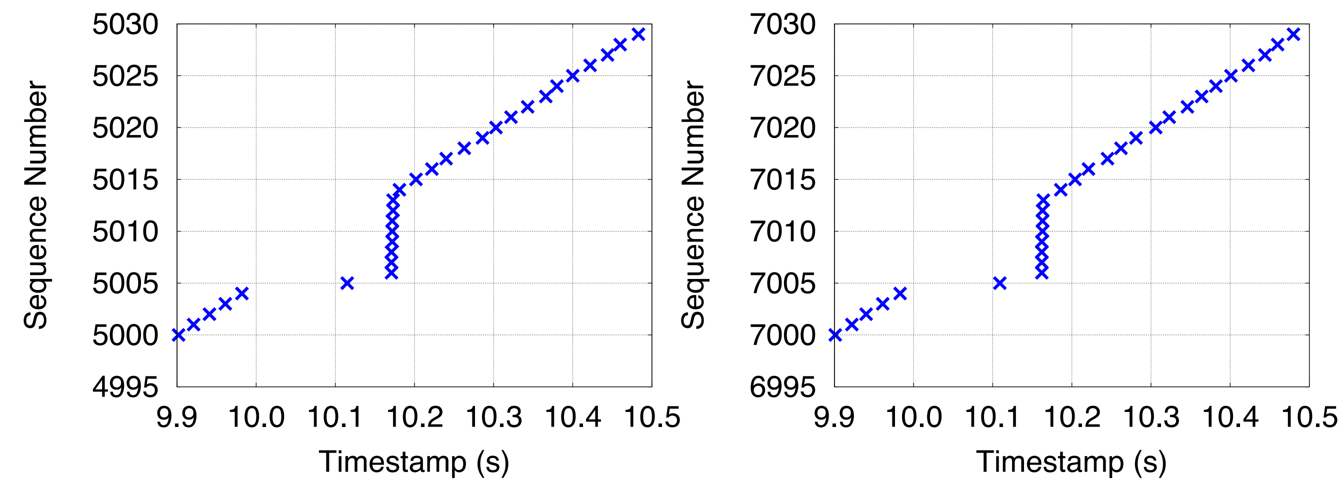
- We implemented MobiCCN in the CCNx prototype as a plugin.
- We choose four realistic networks (Exodus, Sprint, AT&T and NTT) in the evaluation. The topology files are form Rocketfuel project.
- All the experiments are performed on our department cluster consisting of 240 Dell PowerEdge M610 nodes.

Network	Routers	Links	POPs	Diameter	Avg. Path
Exodus	338	800	23	12	5.824
Sprint	547	1600	43	12	5.182
AT&T	733	2300	108	11	6.043
NTT	1018	2300	121	14	6.203

Simultaneous Handoff Delay

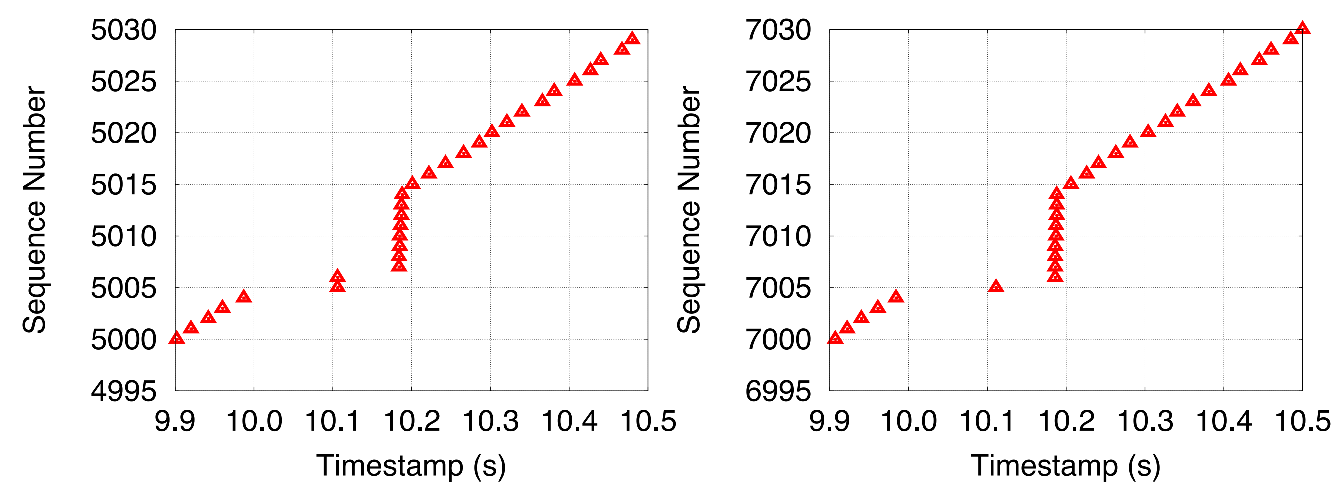
- The link delay is set to 5 ms.
- The initial placement of the sender and receiver is arbitrary.
- The selection of the next access point of the mobile sender is among the nodes within a 2-hop radius.
- Layer 2 handoff delay is set to 100 ms, and loss detection timer is also set to100 ms
- Both caller and callee perform a simultaneous handoff at 10 sec.

MC



(a) Caller (b) Callee

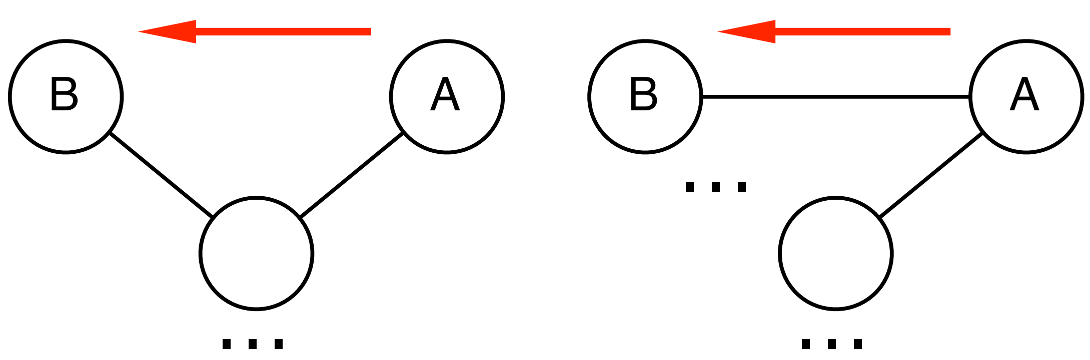
IF



(a) Caller (b) Callee

Interest Forwarding is Subject to Topology.

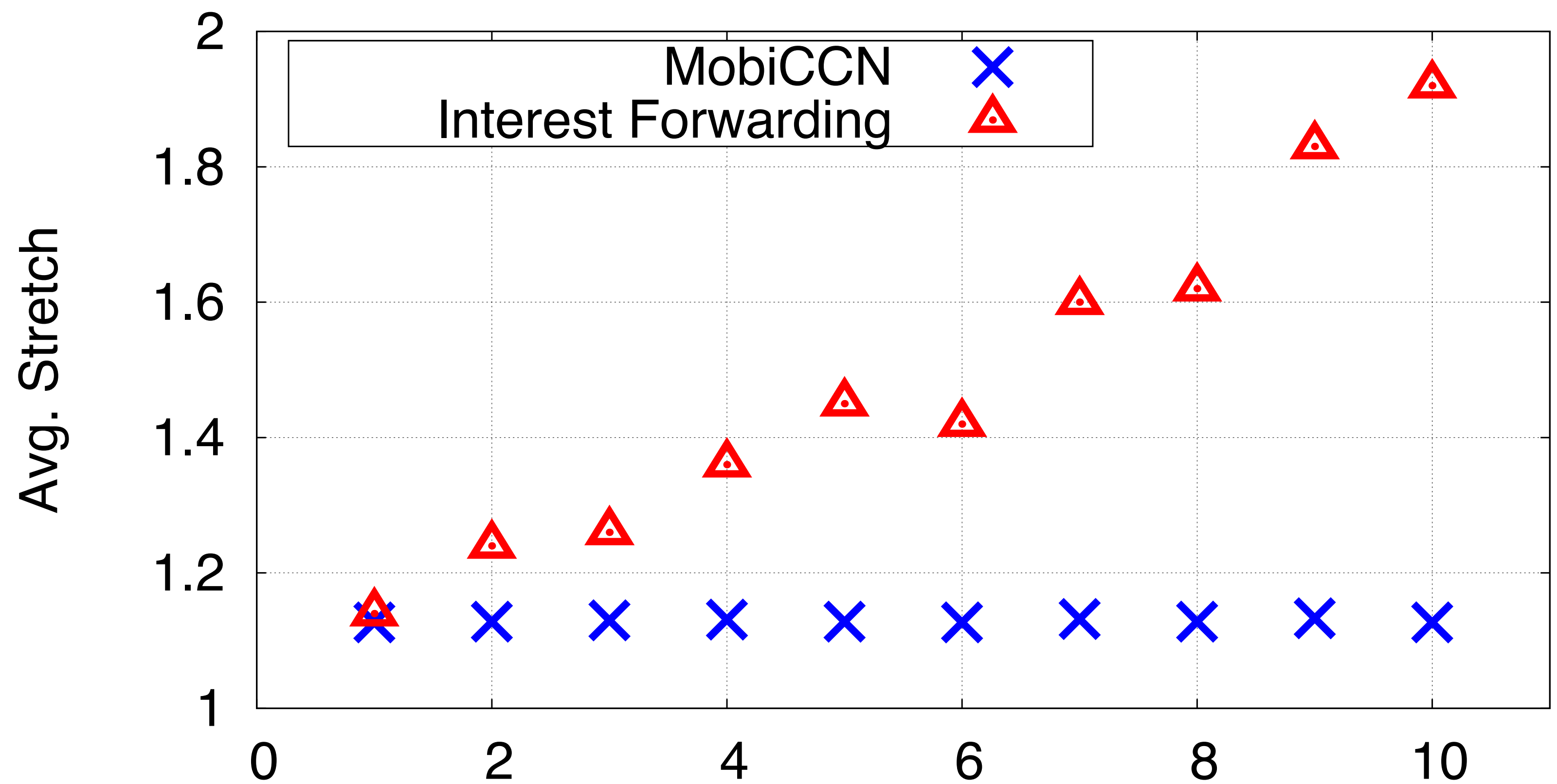
IF's simultaneous handoff delay is consistently larger than MobiCCN. The reason is path between data source and data receiver will grow if there is continuous move.



(a) Topology α (b) Topology β

If data source moves from A to B, topology α will not increase the path length. However, topology β will increase the path length by 1. Triangular routing cannot be eliminated in IF.

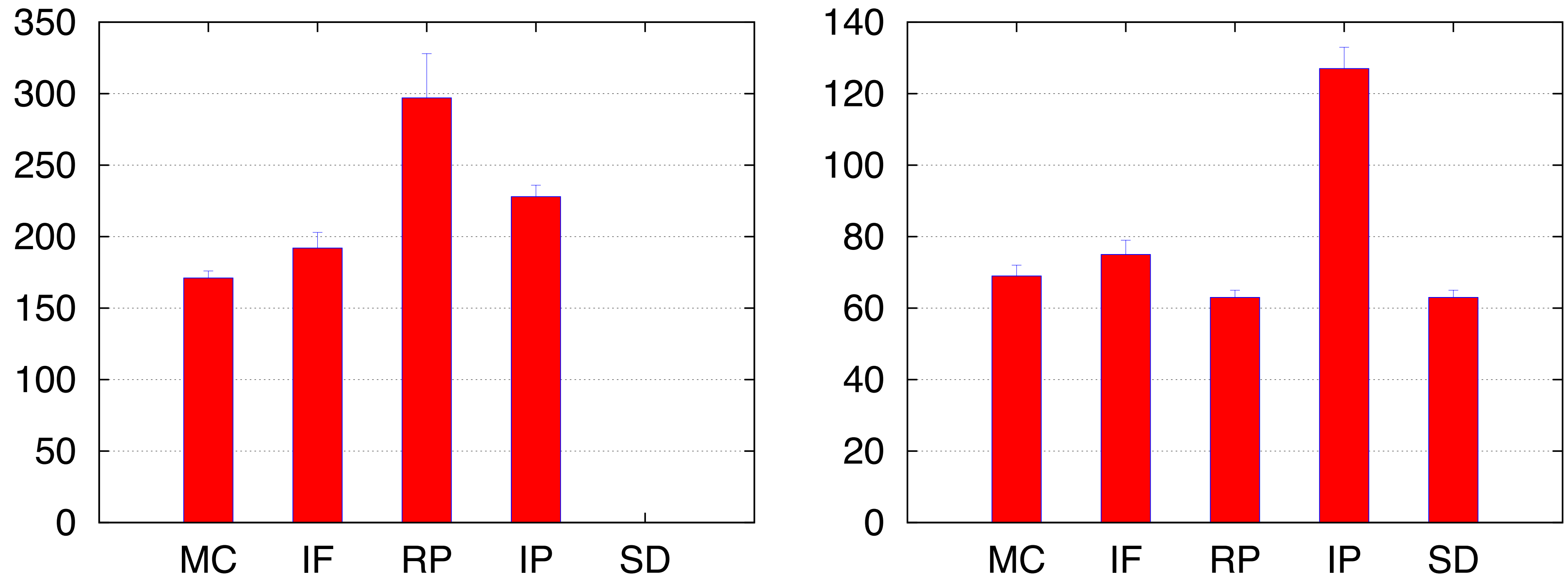
Scalability – Continuous Moves



Average stretch as a function of number of handoffs

In the experiment, callee is fixed and caller moves N times. Every 10 sec, caller moves to a new access point. We measured the stretch between caller and callee after each handoff.

Compare with Other Schemes



(a) Handoff delay (ms) (b) Average latency (ms)

- Comparison of latencies in different schemes. **MC**: MobiCCN, **IF**: Interest Forwarding, **RP**: Rendezvous Point, **IP**: Indirection Point, **SD**: Sender-Driven Message.
- In summary, MobiCCN outperforms the other solutions in terms of delay and (for the most part) latency.

Conclusion & Future Work

- We use greedy routing to solve mobility and mobile content publishing and dissemination issues in CCN.
- With greedy embedding, we distribute the rendezvous points and name resolution functionality into the network.
- We compared MobiCCN to other proposed mobility schemes and showed that it outperforms the existing schemes both in terms of handoff delay and communication latency.
- Future work:
  - Greedy routing is beneficial in terms of compact routing, small routing table. But embedding into hyperbolic space is expensive operation.
  - Develop better embedding algorithms of low stretch.
  - Sacrifice accuracy for efficiency, use other metric spaces.

Thank you!

Questions?

Summary of Comparison

	Avg. Latency	Handoff Delay	Simultaneous Handoff	Scalability	Single Point of Failure	Complexity
MobiCCN	Medium	Low	Yes	High	No	Medium
Sender-Driven Msg	Low	High	No	High	No	Low
Rendezvous Point	Low	Medium	Yes	Low	Yes	Low
Indirection Point	High	Medium	Yes	Low	Yes	High
Interest Forwarding	Medium	Low	Yes	Medium	No	High

Comparison of different mobility schemes. MobiCCN achieves good trade-off point from various perspective.