CPSC 416 Distributed Systems

Winter 2023 Term 1 (November 16, 2023)

Tony Mason (fsgeek@cs.ubc.ca), Lecturer

Logistics

Teaching Assistants

Andy Hsu (andy.hsu@alumni.ubc.ca)

Hamid Ramezanikebrya (hamid@ece.ubc.ca)

Jonas Tai (jonastai@student.ubc.ca)

Cathy Yang (kaiqiany@student.ubc.ca)

Office Hours

Remember: Use Piazza for **all** official course-related communications

- Not on Piazza? Not official.
- Canvas "comments/messages" **are not monitored**

Office Hours:

Who	When	Where
Tony	Monday 14:00-15:00 Wednesday 16:00-17:00	Discord
Andy	Thursday 19:00-20:30	Discord
Hamid	Friday 16:30-18:00	Kaiser 4075
Jonas	Thursday 13:00-14:00	Online (see Piazza)
Cathy	Friday 09:00-10:30	X237

Self-Assessment

This week

- DP3 Implementation Report (Thu @ 23:59)

Next week

- Capstone Status Report (Tue @ 17:00)
- DP3: Peer Review Implementation Reports (Thu @ 17:00)
- Note: no self-assessment activity

Note:

- You are strongly encouraged to collaborate with others on this
- You should use tools at your disposal to answer these questions
- **Do not forget to submit it.**

Today’s Failure

Former Student: Replication Lag

Source: student from CPSC 416 in Winter 22 Term 2.

Date: 2023/11/14 (reported)

Description: *Last month I literally witnessed a failure of the day scenario occurring live in my company. Our product suffered an outage due to replica database lagging behind to the point where commits were locked up.*

While not a lot of detail, it's good to know it is real.

Second comment: *We are also trying to deal with our high database load with sharding, so the concepts you taught is playing out in front of me.*

Finding success through teaching about failure.

Lesson Goals

Peer-to-Peer and Mobility

Tools for building distributed applications

Chord peer-to-peer system

Overlay networks for mobility

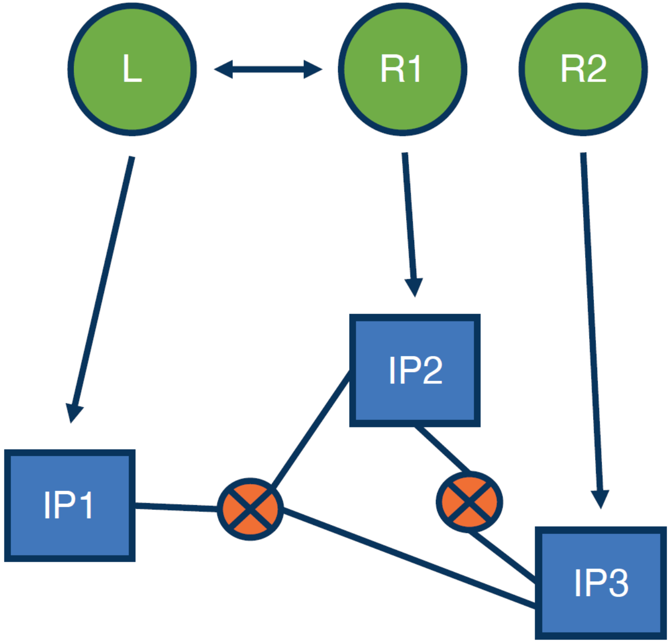
Network Abstraction

Application/service-level namespace

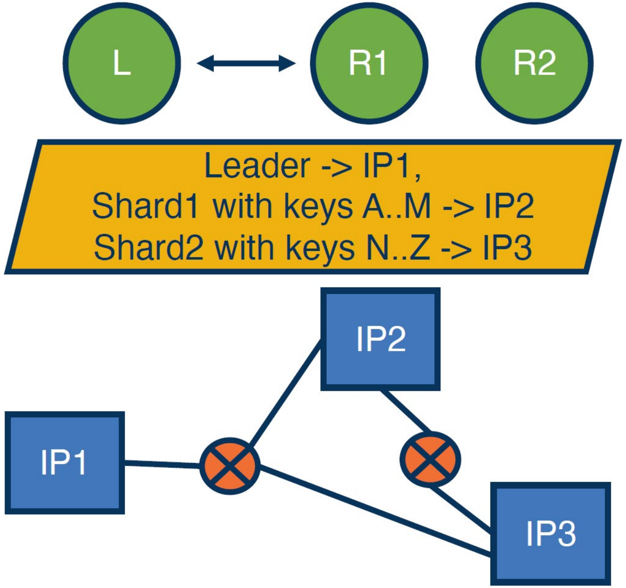
- Process names
- File names
- Object keys
- …

Network level

- IP addresses
- Network paths through switches & routers



Network abstraction



Metadata service

- Determines Overlay Network
- Part of control plane operation

Update on change

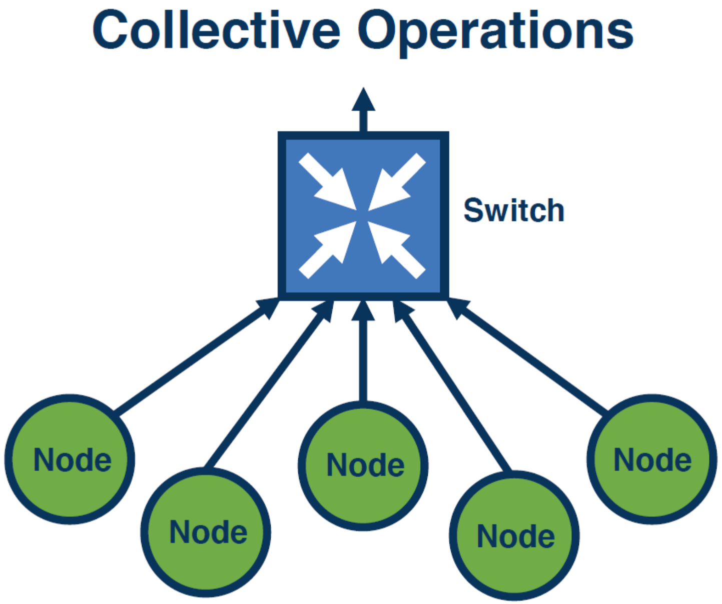
- Scale
- Geo-distribution
- Failures
- Multiple administrative domains

Interconnect Support

- Broadcast, multicast
- Gather/all-reduce
- Barrier
- Atomics (e.g., CAS)
- Timing
- RDMA (Remote Direct Memory Access)
- Direct cache injection (DDIO)

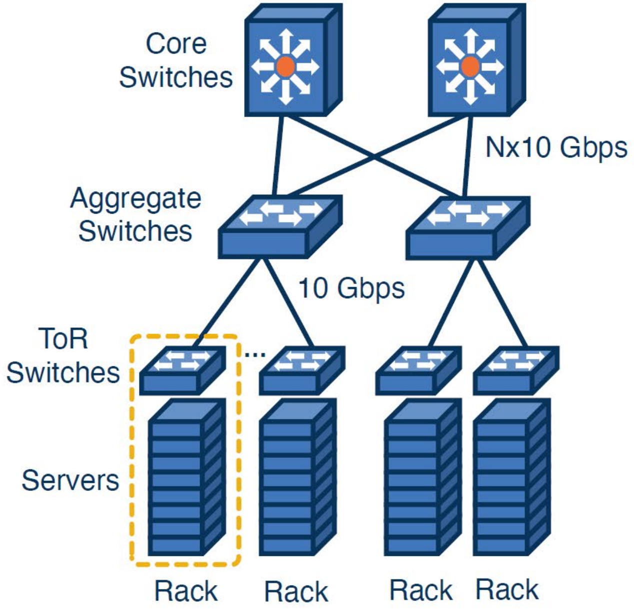
Hardware Scalable Implementations

- Separate dedicated networks
- Combining Tree Algorithms



Peer to Peer Systems

Datacenter Infrastructure



Wide-Area Distributed Infrastructure

Peer to Peer Systems

BitTorrent

Tor

bitcoin

ethereum

Peer-to-peer Connectivity

How do you find the right peer?

Centralized Registry:

- Single round trip time (RTT) to find the peer IP
- Requires a centralized trusted authority

Example: Napster

Peer-to-peer connectivity

How to find the right peer?

Flood or Gossip based protocols

- No single point of failure
- No bound on lookup time

Examples:

- Gnutella
- Bitcoin

Peer-to-peer connectivity

How to find the right peer?

Provide Structured Routing Trees: Distributed Hash Table (DHT)

- Decentralized index
- Probabilistic bounded lookup time

Examples:

- Chord
- Kademlia
- Amazon DynamoDB

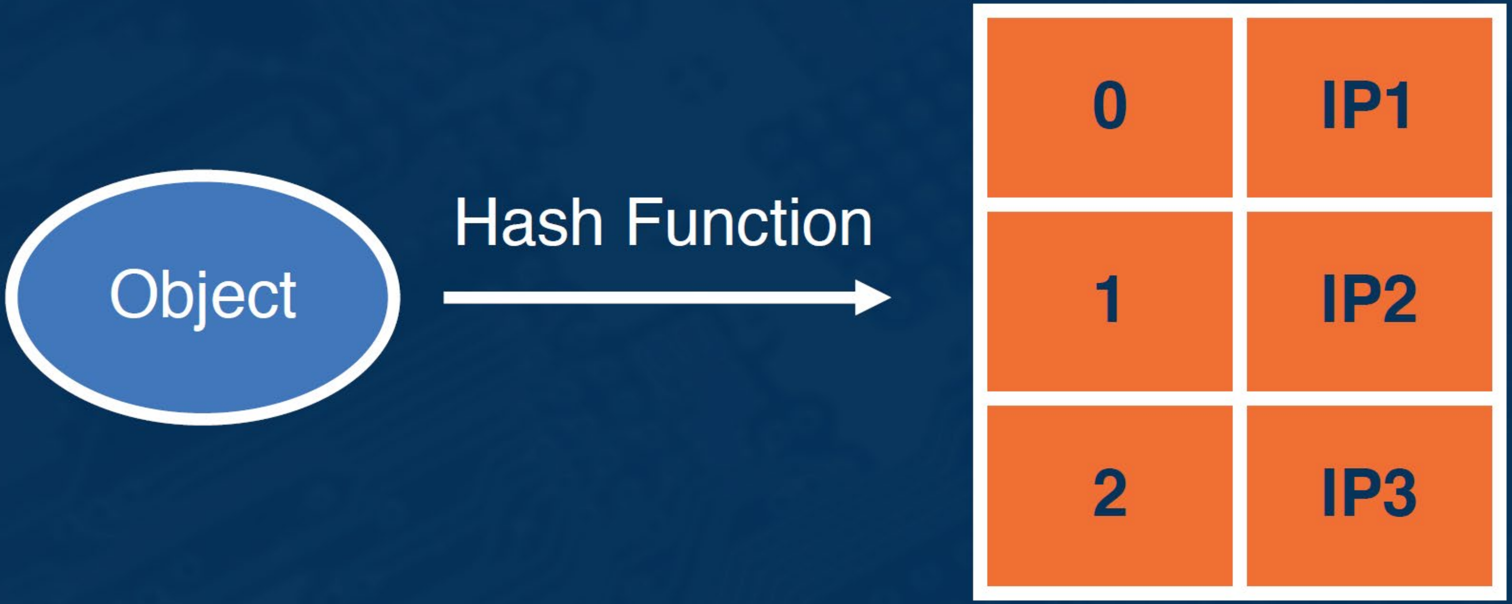
Distributed Hash Table

Hash Function:

- Maps a thing to a unique number within a range
- Key namespace to number namespace
  - File names
  - Song names

Uniform hash function use:

- Same mapping

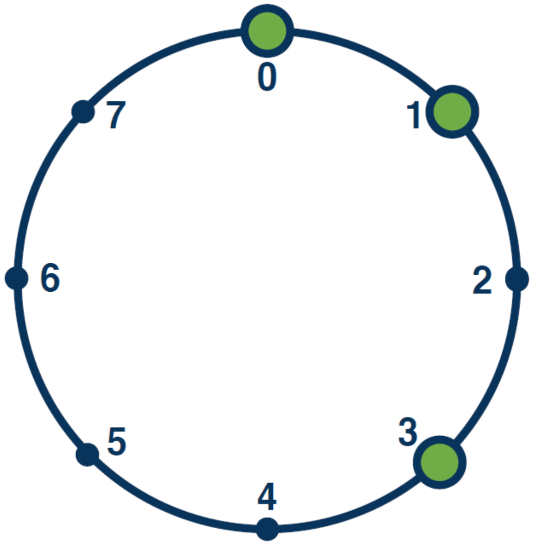

Object
Hash Function
0
IP1
1
IP2
2
IP3

Chord Distributed Hash Table Ring

Use cryptographic secure hash algorithm (SHA)

- Maps keys to a fixed length numeric value
- Maps IP addresses to a fixed length numeric value

Ring is N nodes {0,…N-1}

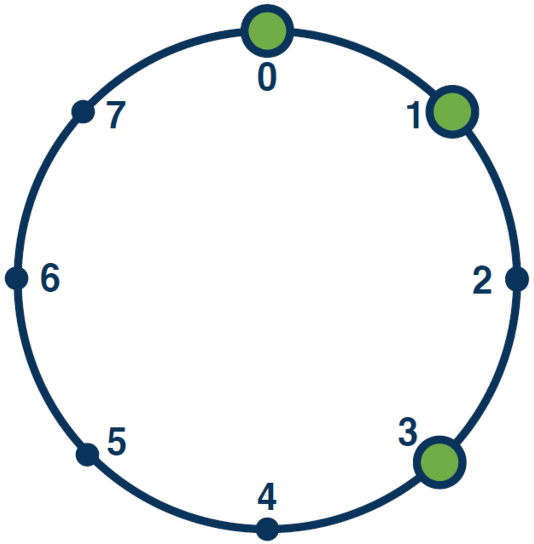


Insert Operation

SHA(key) = value

If node exists at value: update

Else: update successor node



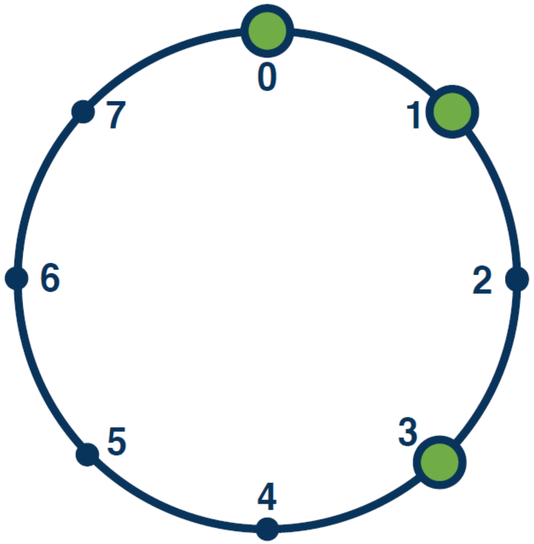
Lookup Operation

SHA(key) = value

If node exists at value: lookup

Else: lookup at successor node

Question: Can we improve over O(N)?



Finger Tables

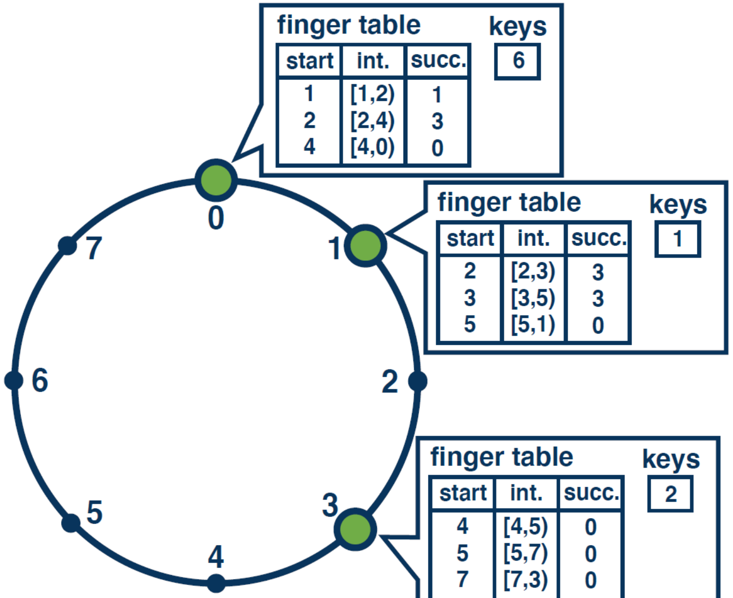
[Finger Tables]

Node ID for progressively longer ranges

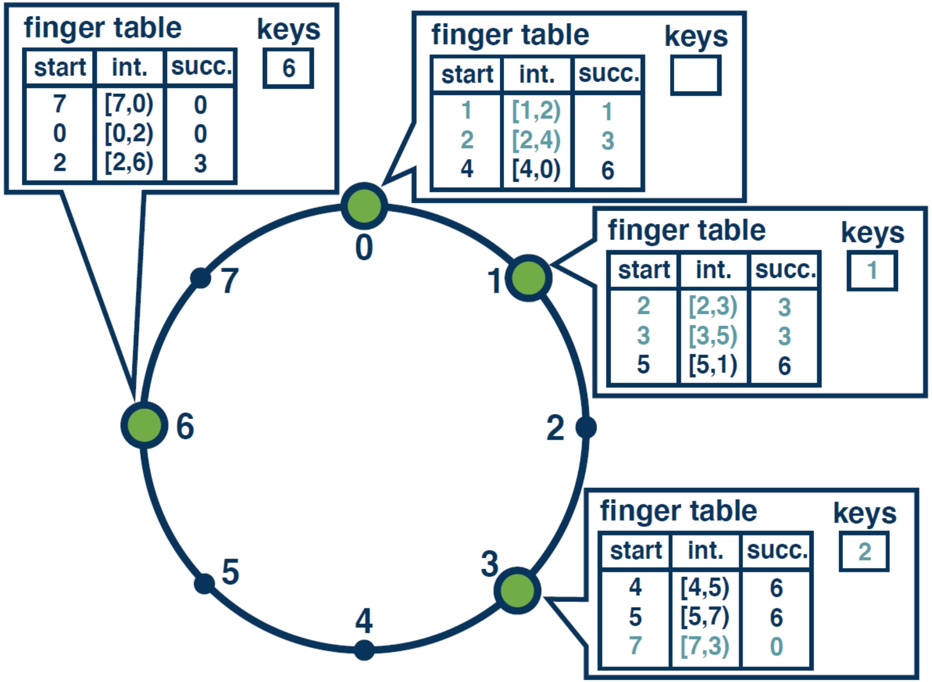
Finger table:

- At each node n
  - i-th finger entry starts at [n +2i]
  - For range of 2i elements

Lookup is O(log(N))



Chord: Managing the Ring



Nodes joining and departing

Redistributed data

Update finger tables

Improve performance with additional metadata

Probabilistic system performance guarantees

Hierarchical Designs

Cost of communications versus cost of overlay maintenance

Nodes with different properties:

- Point-to-point communication
- Stability, failure probability, mobility
- Number and type of nodes
- Communications patterns, locality

Hybrid approaches

- Large-scale datacenters
- Wide area
- Mobile networks

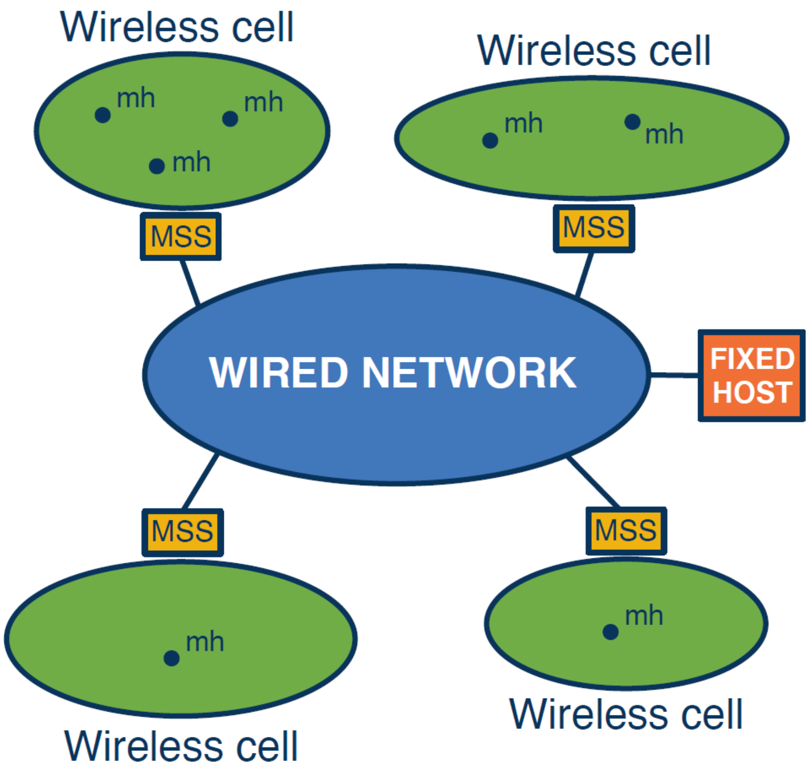
Mobile Network Model

Mobile Support Stations (MSS)

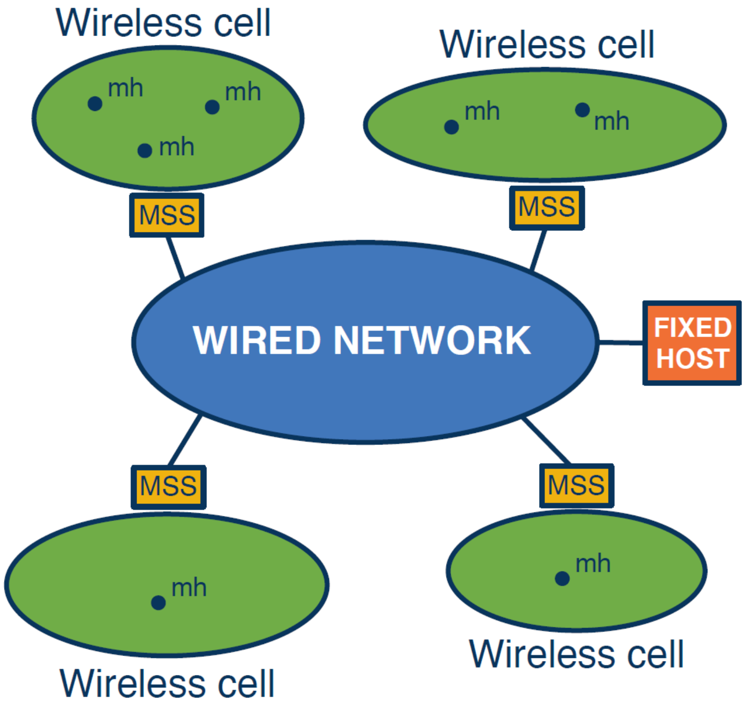
- Stationary
- High-speed wired network
- No power availability concerns

Mobile Hosts (MH)

- Associated with an MSS
- Mobile
- Lower speed mobile network
- Battery power concerns



Mobile Network Model



Goal:

- Fast lookup of MH
- Low overhead update of overlay state
  - Communications overhead
  - Battery/energy/compute overhead

Heterogenous nodes have *different* concerns

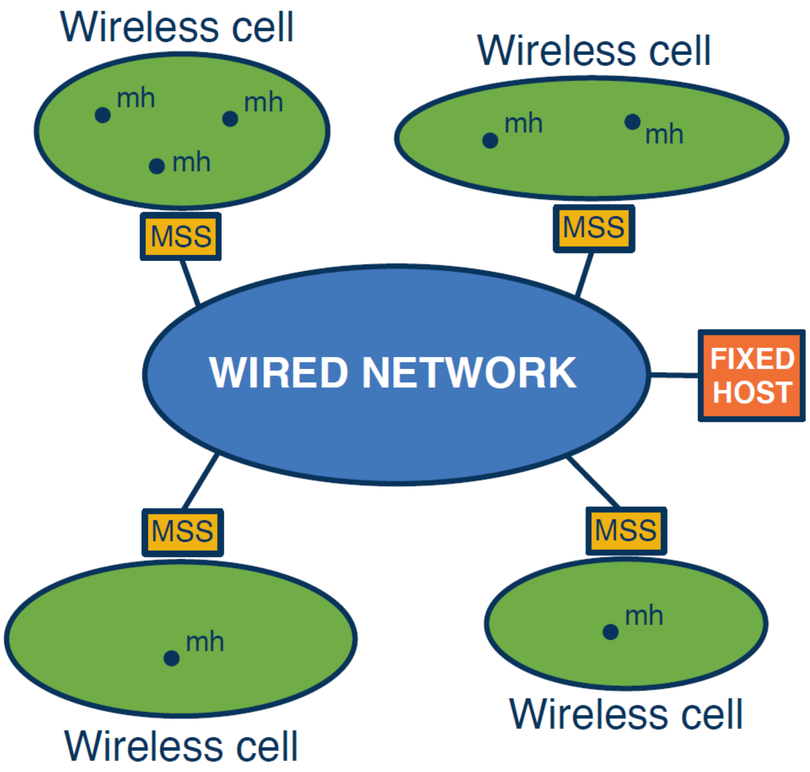
Different Algorithms

Metrics

- Search (lookup) cost
- Insert (add/remove) cost
- Mobility support update impact

Analysis Considerations

- $C_{wireless} >> C_{fixed}$
- $N_{mh} >> N_{mss}$



Communications Cost

$Cost_{communications} = 2 * Cost_{wireless} + Cost_{search}$

Algorithm 1:

- Logical ring of all mobile hosts
- $Cost_{search} \sim O(N_{mh}, Cost_{wireless})$

Algorithm 2:

- Two-tier hierarchical design
- Mobile Support Stations (MSS) in logical ring
- Each MSS knows about Mobile Hosts in its cell
- $Cost_{search} \sim O(N_{mss}, Cost_{fixed})$

Algorithm 2 is a clear winner here

Mobility Support Cost

Algorithm 1:

- Original MSS search for new MSS *on demand*
- No update on move, only when needs to reach MH

$Cost_{update} \sim O(Cost_{fixed\_search})$

Algorithm 2:

- New MSS informs original MSS each time a new MH joins
- Update needed each time MH moves

$Cost_{update} \sim O(\#moves * Cost_{fixed})$

Lesson Review


UBC
THE UNIVERSITY OF BRITISH COLUMBIA
THE UNIVERSITY OF BRITISH