Some Routing Challenges in Dynamic Networks

Jie Wu
Dept. of Computer and Information Sciences
Temple University

IEEE MASS 2011

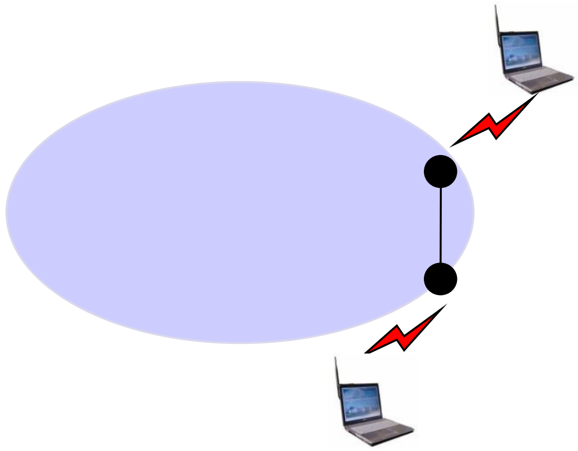
Overview

1. Current State of Networking
   - More Wireless
   - Mobile and Opportunistic Applications
2. New Routing Challenges
   - Sample Challenges
   - Hybrid Solutions
3. Graph Algorithmic Solutions
   - Discrete Contacts: Dijkstra
   - Probabilistic Contacts: Bellman-Ford
4. A Bigger Picture
   - Future Applications and Challenges

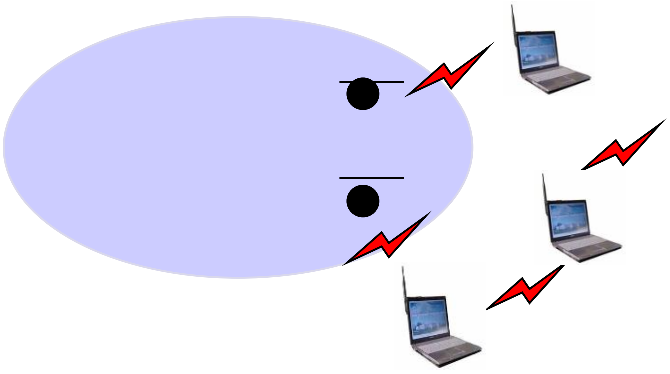
1. Current State: More Wireless

- Current
  - Different types: smartphone, BlackBerry, iPad
  - Internet connections: more wireless
  - Dynamic link quality
- (Near) future
  - 1 billion vehicles
  - 5 billion RFID
  - 10-15 billion sensor/embedded devices
- Future: anytime, anywhere

(a) Edge of the Internet



(b) General way of data transmission

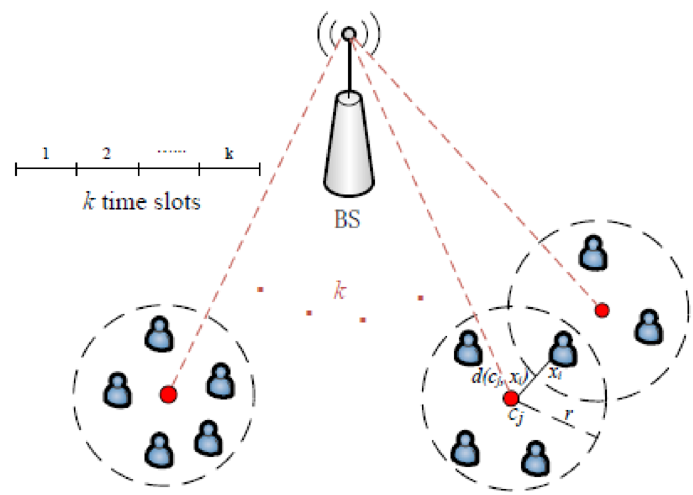


Mobile and Opportunistic Applications

- Node mobility
  - MANET (relatively dense)
  - DTN (relatively sparse)
- Opportunistic contact
  - Temporal nature
  - Fixed and opportunistic contacts

NSF grant: Mobile Content Sharing Networks: Theory to Implementation, 2011

- Explore both wireless channels and mobility
- The whole is greater than the sum of parts!
- Maximizing happiness (Wang, Guo, and Wu, ICPP 2011)

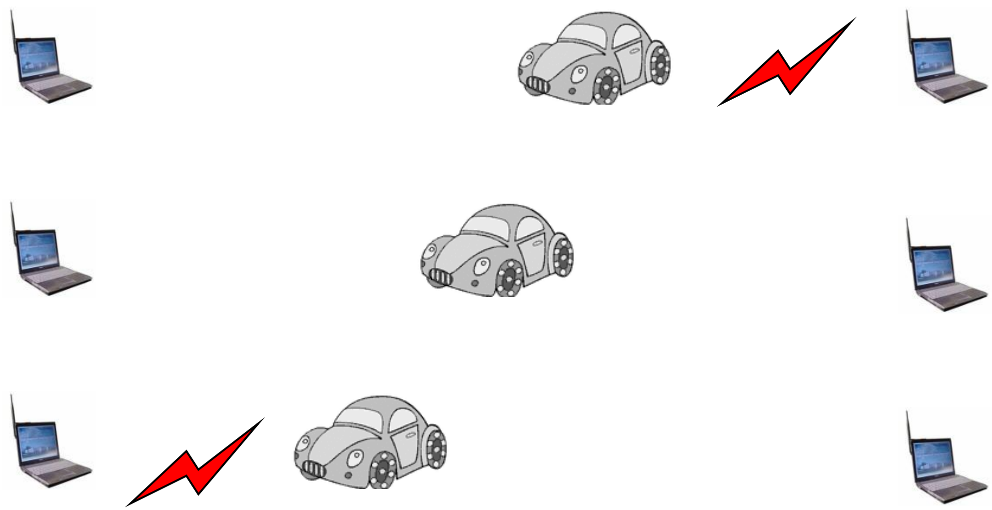


2. New Routing Challenges

- Protocol design
  - Store-and-forward vs. store-carry-and-forward
- Mobility
  - Random movement vs. controlled mobility
- Search Efficiency
  - Improve search efficiency
    - connected-dominating-set (CDS)
  - Reduce search time
    - replication

Protocol Design

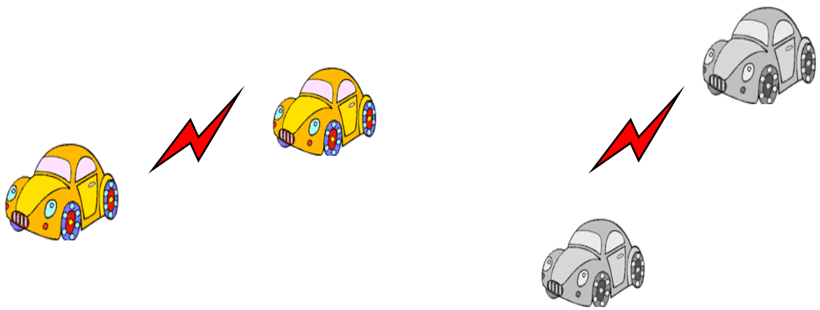
- TCP/IP connection-based: store-and-forward
- Non-connection-based: store-carry-and-forward
  - Store
  - Carry
  - Forward



Mobility

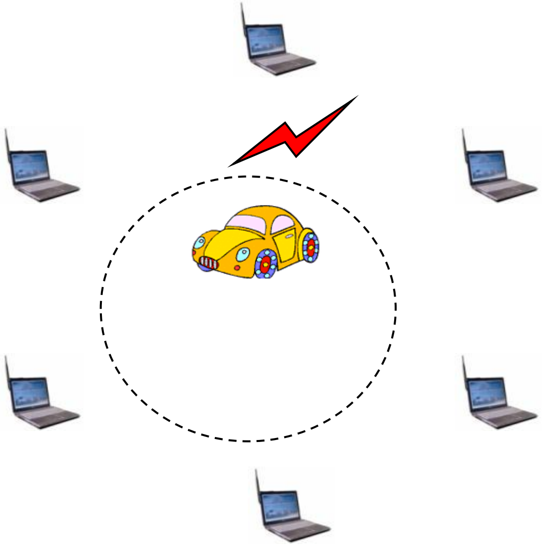
Random: Epidemic Routing

- Nodes store and exchange the date when they meet
- Data is replicated by way of a random walk



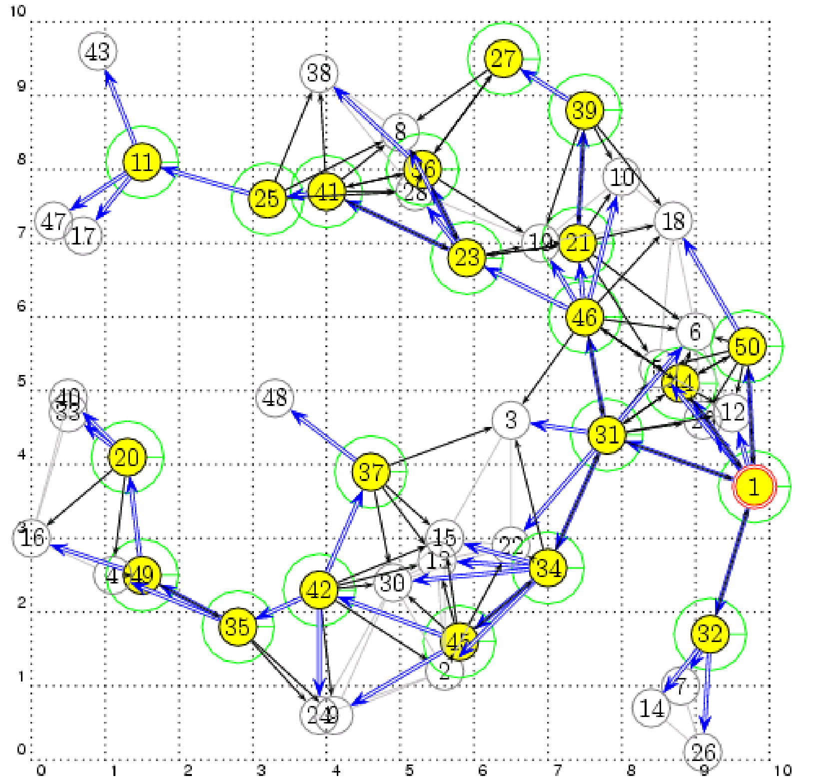
Controlled: Message Ferrying

- Special nodes (ferries) have completely predictable routes in the geographic area



CDS: Improve Search Efficiency

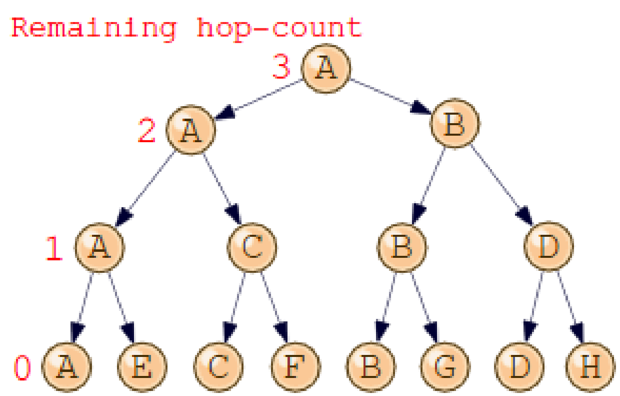
- Connected Dominating Set (CDS)
  - CDS: a connected subset
  - Any other node has a CDS neighbor
- Features
  - Virtual backbone
  - Local formation of a CDS
  - Efferent routing/broadcasting
- Extensions
  - k-connected, k-dominated CDS
    - Controlled redundant coverage
  - Directional CDS
    - Reduce interference
    - Save energy



NSF grant: A Dominating-Set-Based Routing Scheme in Ad-Hoc Wireless Networks, 2000

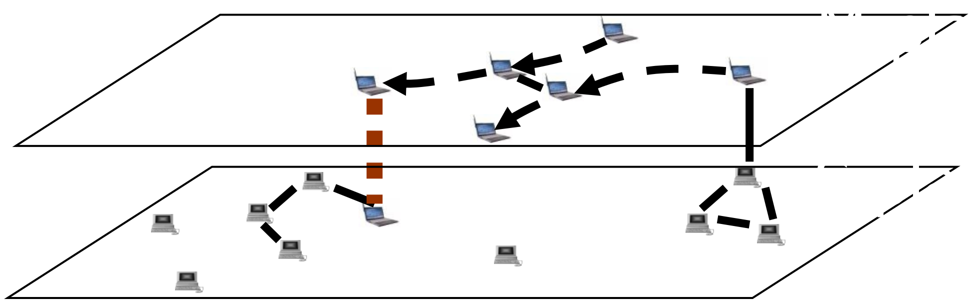
Replication: Reduce Search Time

- Delegation forwarding
  - Each node has its own estimated distance to the destination
  - A packet holder will only forward the packet to a node with a higher quality
  - Number of forwardings: $O(\sqrt{N})$
- Extended delegation forwarding
  - Relative forwarding qualities among all possible future contacts
  - Optimal stopping rule
    - Hop-count limited
    - Time-to-live (TTL)
  - Best time to forward: backward induction (Liu and Wu, MASS' 2009)



Hybrid Solutions

- Store-and-carry and store-carry-forward (Wu, 2005)
  - M(mobile)-plane: mostly store-carry-and-forward
  - S(stationary)-plane: store-and-forward
  - Two types of nodes: carriers and keepers



NSF grant: Mobility-Assisted Routing in Mobile Networks, 2006

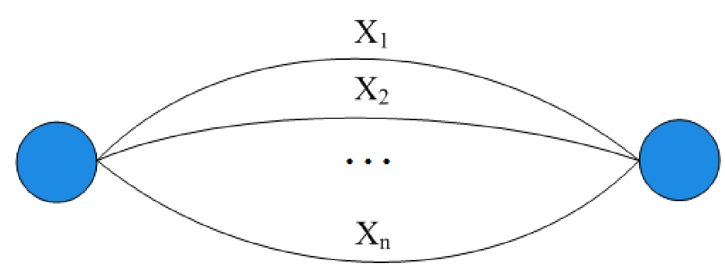
- Replication-based and non-replication-based (MobiCom'11, UMass)
  - Non-replication: MANET

    $$\mu = \min\{E[X_1], E[X_2], \ldots, E[X_n]\}$$

  - Replication: DTN

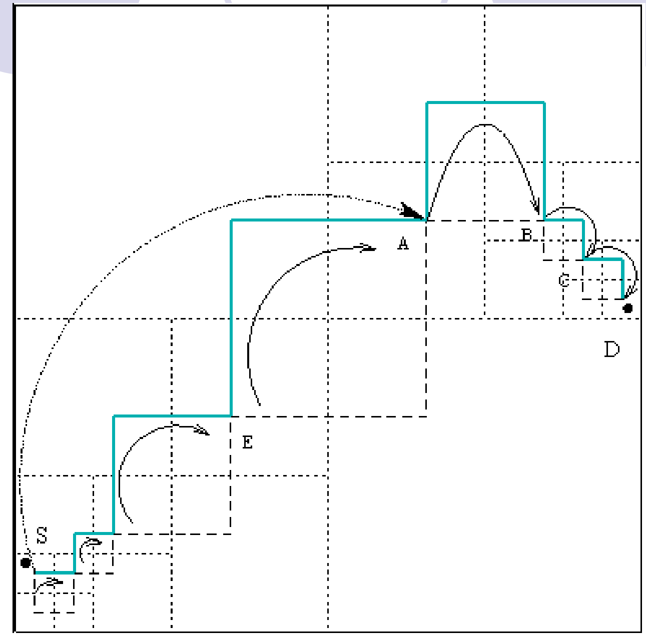
    $$\mu_{(1)} = E[\min\{X_1, X_2, \ldots, X_n\}]$$

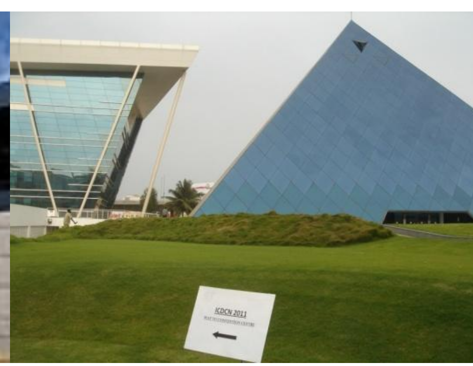
  - Replication gain: $\mu / \mu_{(1)}$
  - State transition: replication vs. forwarding based on estimated and actual delays

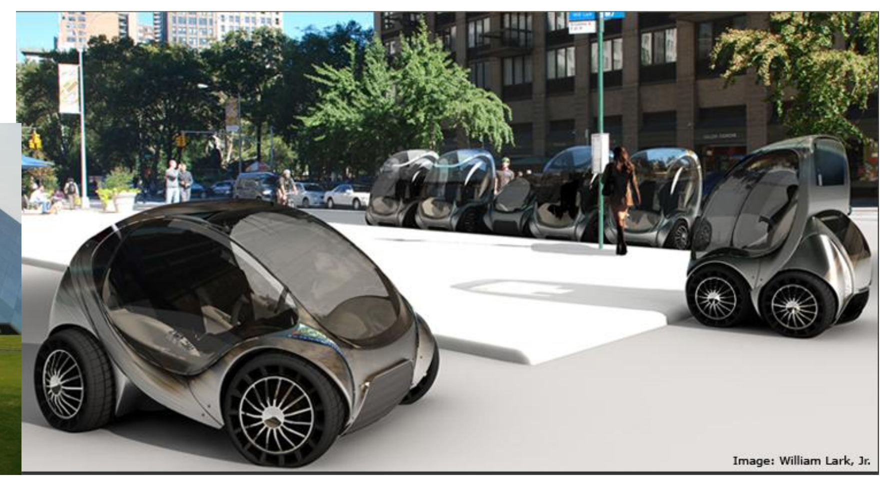


Hybrid Solutions

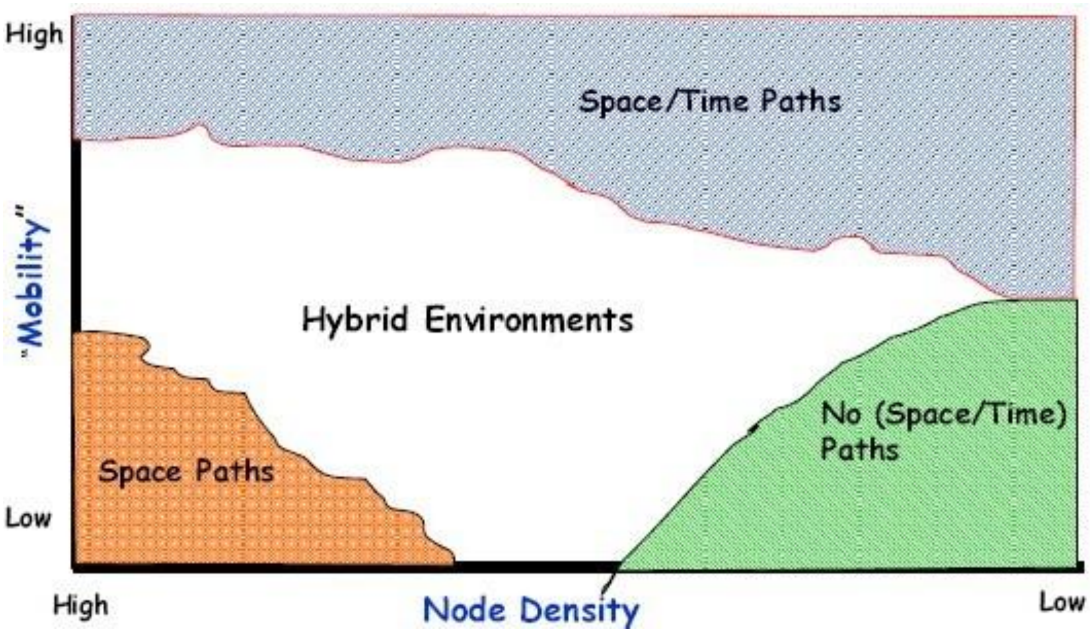
- Fixed and on-demand routes
  - Fixed route: bus
  - On-demand: taxi
  - Hybrid: mixture of fixed and on-demand (Wu, Yang, and Dai, TPDS'07)
- Future transportation
  - CityCar – MIT Smart City
  - Semi-on-demand taxi – Microsoft campus
  - Public bike - Infosys

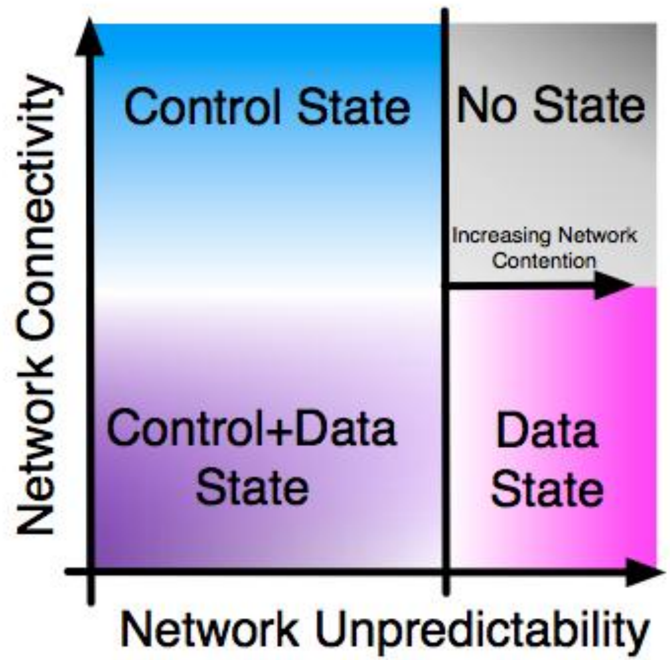






Sample Classification





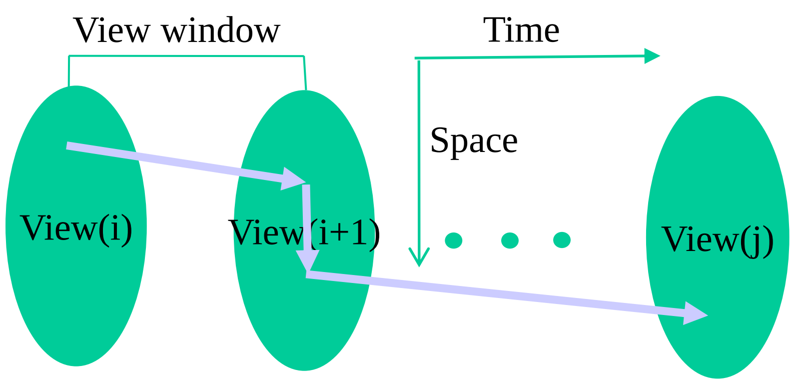
M. Ammar, MASS 2007 keynote

Link-up entropy, H(g_{t+i} | g_t) (Mobicom'11, UMass) g_t: arbitrary link up/down in G_t

3. Graph Algorithmic Solutions

- Extend the classical graph model to the time-space model
  - Discrete contact model: Dijkstra algorithm
  - Probabilistic contact model: Bellman-Ford algorithm

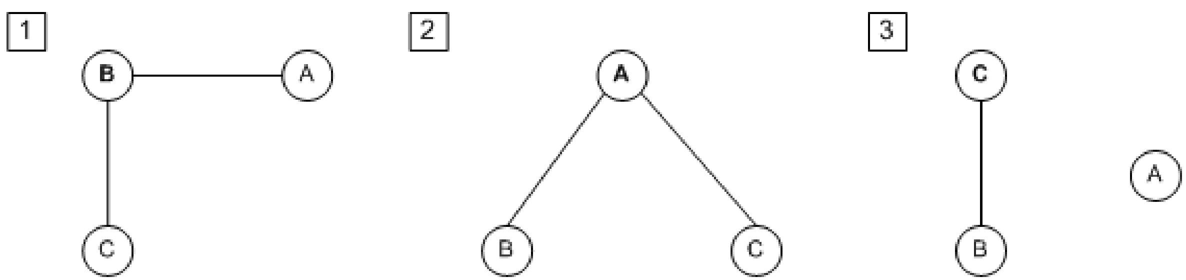
- (u,v) - connectivity under time-space view
  - All i, (u(i), v(i))
  - Exist i, (u(i), v(i))
  - Exist i, j, (u(i), v(j))

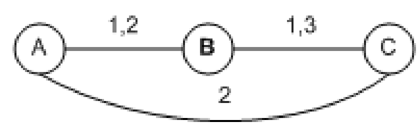


Graph Extension: Time-Space Model

- Time sequence: $t_1, t_2, \ldots, t_L$
- $G_i = (V_i, E_i)$: subgraph in $[t_i, t_i + \Delta]$
- Evolving graph: G: $G_1, G_2, \ldots G_L$

  where $(u,v) = \{i \mid (u, v) \in E_i\}$ (i: time label)





A. Ferreira, Building a Reference Combinatorial Model for MANETs, IEEE Networks, Oct. 2004

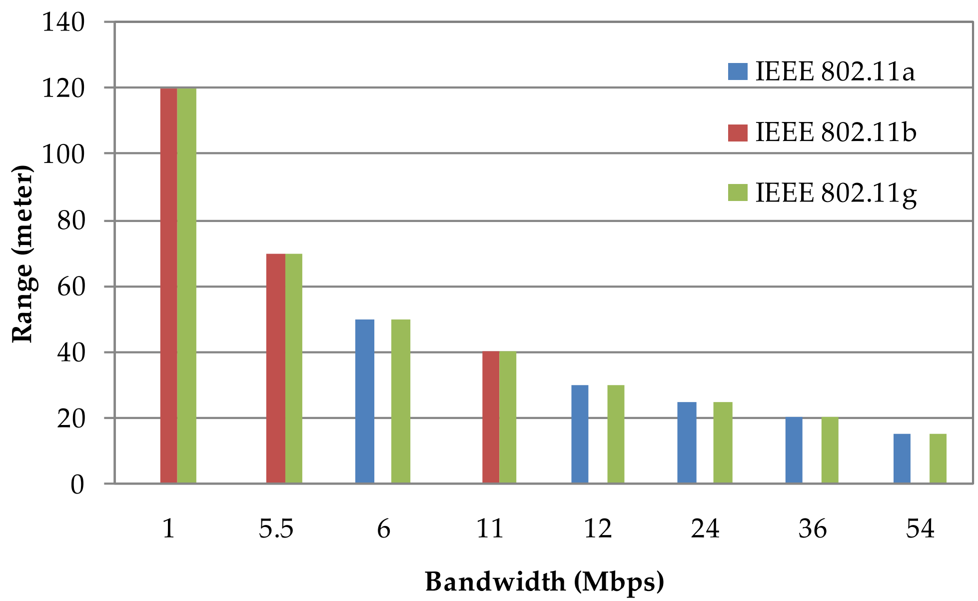
Extended Evolving Graph

- Weighted evolving graph

$$(u, v) = \{(i, w_i) \mid (u, v) \in E_i\}$$

where weight label w_i is the bandwidth, reliability, latency, etc

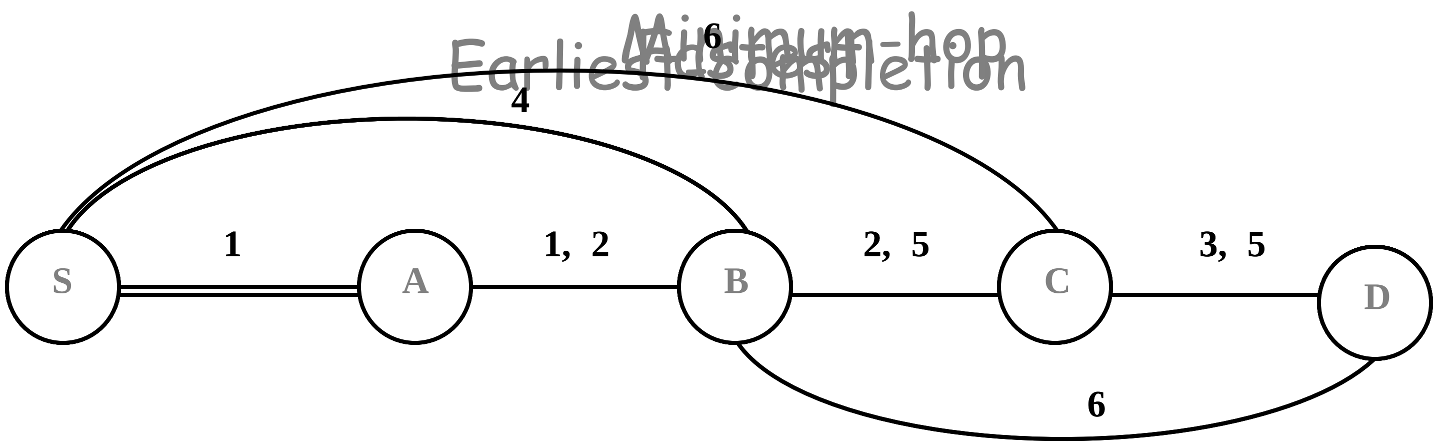
NSF grant: A New Algorithm an Graph Model for Networking in Challenged Environments, 2009



Optimization Problems

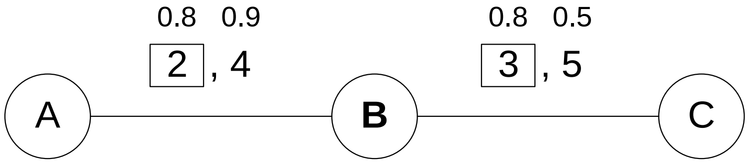
Optimization

- Earliest-completion
- Fastest (min. time span)
- Minimum-hop
- Maximum-bandwidth
- Maximum-reliability



Challenges

- Greedy
  - Greedy-choice property: a globally optimal solution by making locally optimal (greedy) choices
- Optimal greedy
  - Optimal prefix principle: an optimal solution to a problem contains optimal solutions to sub problems within it
- Optimal prefix principle failure



Slicing and Virtualization

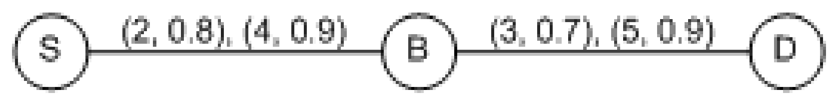
- Virtualization
  - Enlarge G to G′ through virtualization
  - Solve G′ which includes a solution for G
- Slicing
  - Partition G into G_1, G_2, …, G_i
  - Select the best among i solutions for G_i

Wu, Graph, Combinatorics, and Computing Conf. 2010

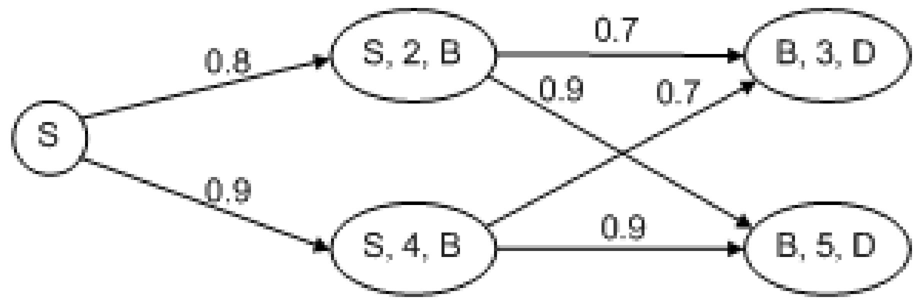
Maximum Reliability

Virtual Graph (G')

- Each node hosts a certain number of virtual nodes



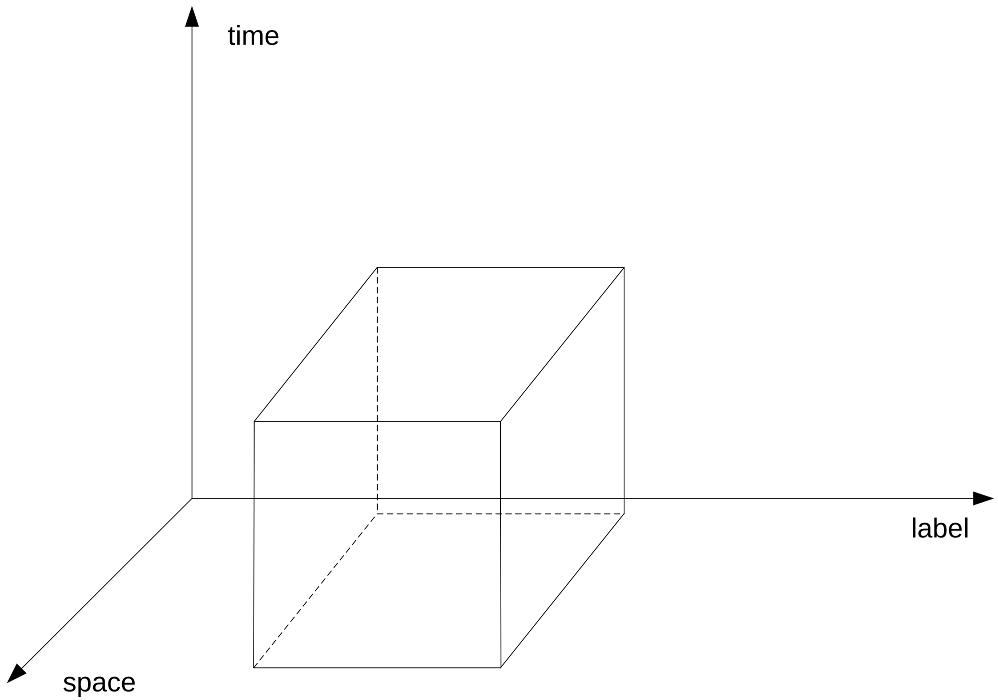
- For a node v in (u, v) with labels $\{l_1, l_2, \ldots, l_{L'}\}$, L' virtual nodes (u, l_i, v) are used for each v



- Dijk(G'), where G'=(V', E'), $|V'| = \Delta L|V|$, and $|E'| = \Delta L^2|E|$
- High complexity!

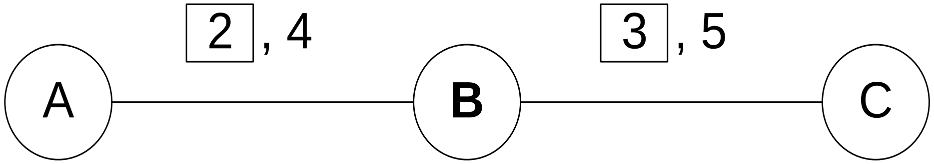
Slicing Methods

- Multi-dimension: time, space, labels
- Overlapping vs. non-overlapping



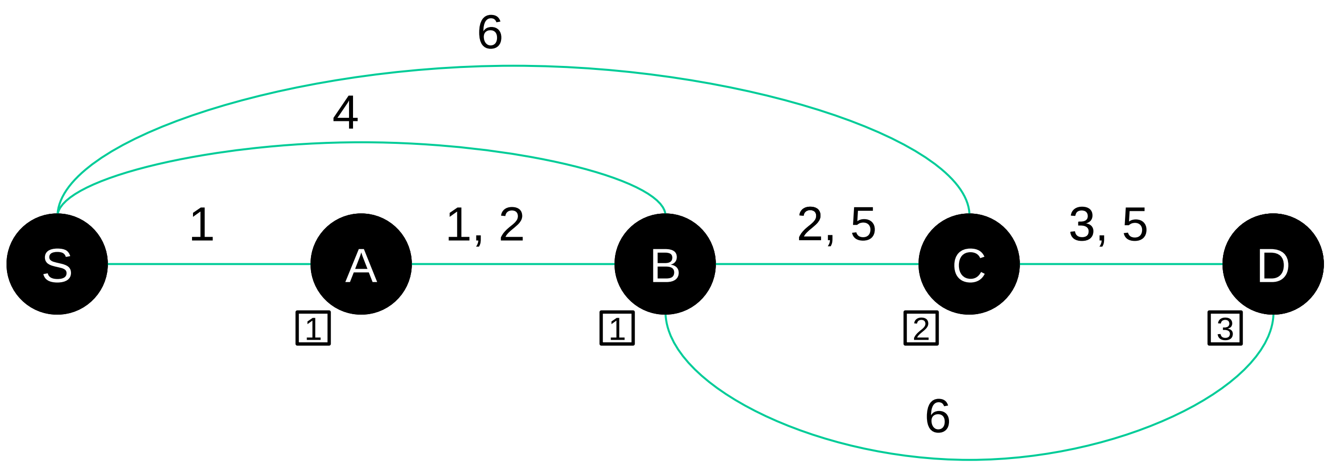
Journey

- Journey
  - A selection of non-decreasing link labels along a path
  - E.g., (2, 3), (2, 5), or (4,5)
- Earliest journey
  - A journal with the smallest last label



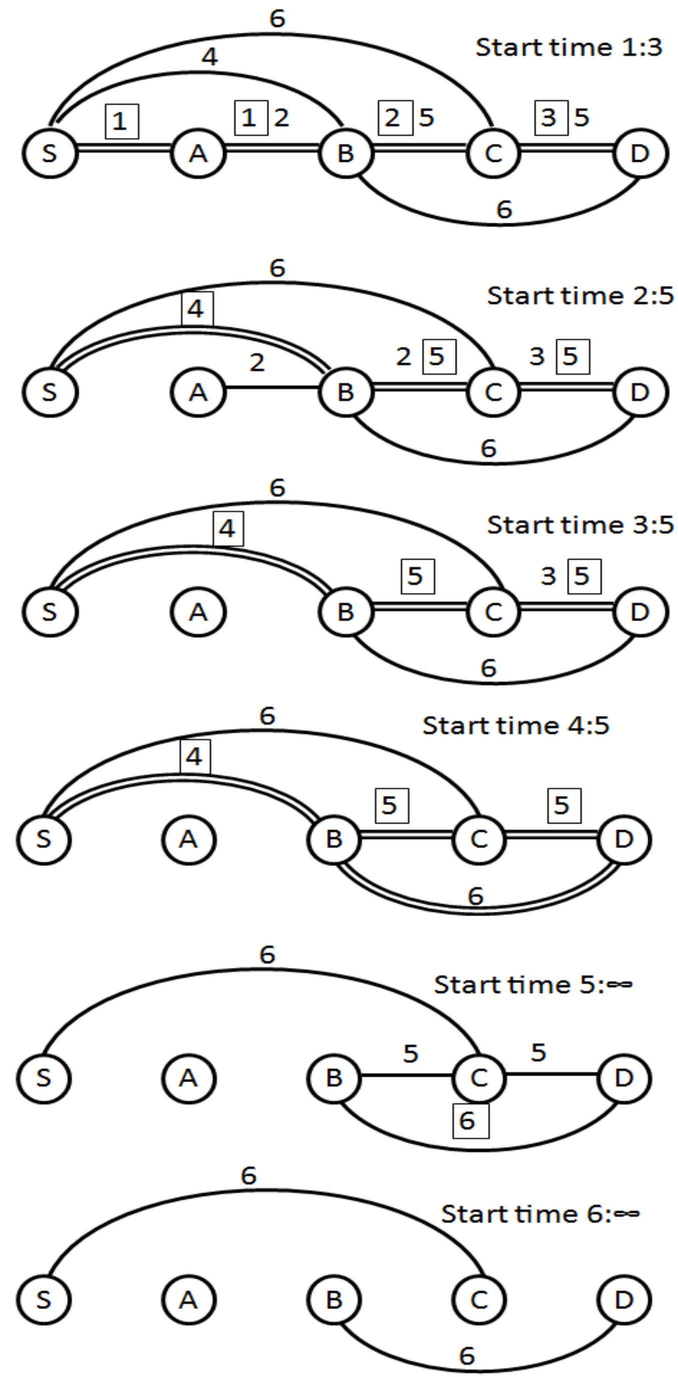
Earliest Completion Path

- Earliest completion path for G
  - Dijk (G) with "best" being the earliest journey of a path
  - Complexity
    - O(V log (LE)) using a heap
    - O(V log V + LE) using a Fibonacci heap



Fastest

- Sliding on start time:
  - Start time is t at s: G(t), a subgraph of G
  - Apply Dijk(G(t)) for earliest completion time
  - Suppose completion time for d is f_t, then time span is $s_t = f_t - t$
  - Fastest: $\min\{s_t\}$
  - Complexity: L times of Dijk



Minimum Hops

- Slicing on time label i of (i, w_i):

G(i ≤ l): a subgraph with labels ≤ l

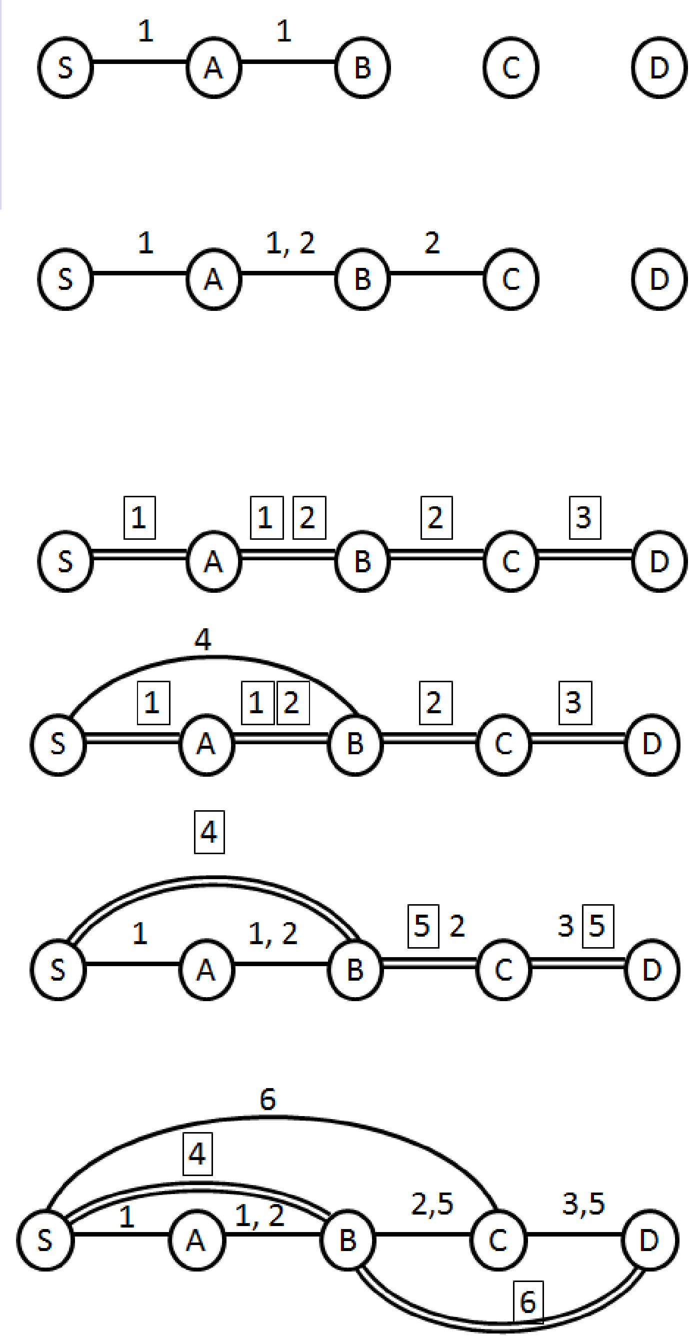
- Dijk(G(i ≤ 1))
- Dijk(G(i ≤ 2)) on above results by relaxing only links with label 2

…

- Dijk(G(i≤ k)) on above results by relaxing only links with label k

The result the is minimum hop count to d after Dijk(G(i ≤ L))

Complexity: L times of Dijk*

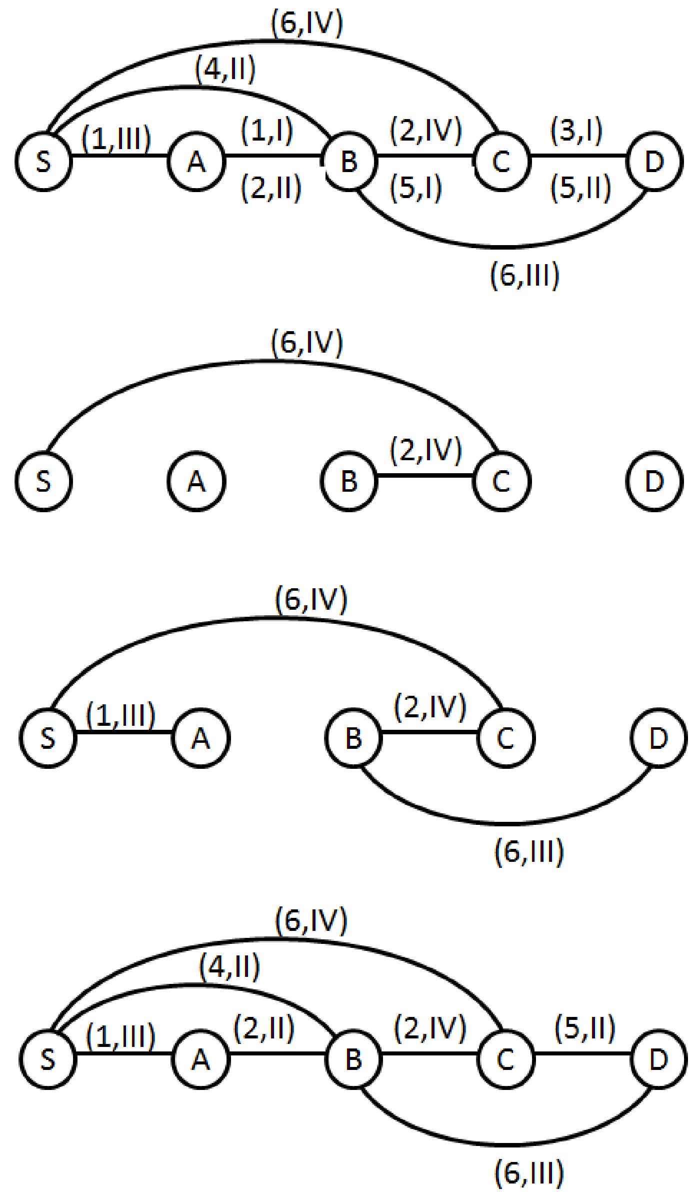


Maximum Bandwidth

- Slicing on weight label w_i of (i, w_i)

Round b (starting with the largest)

- Dijk($G(w_i \geq b)$) /* subgraph of labels with bandwidth ≥ b */
- Stop if d is reachable and return bandwidth b
- Otherwise, repeat the above for b = b-1
- Complexity: log L times of Dijk using the binary search

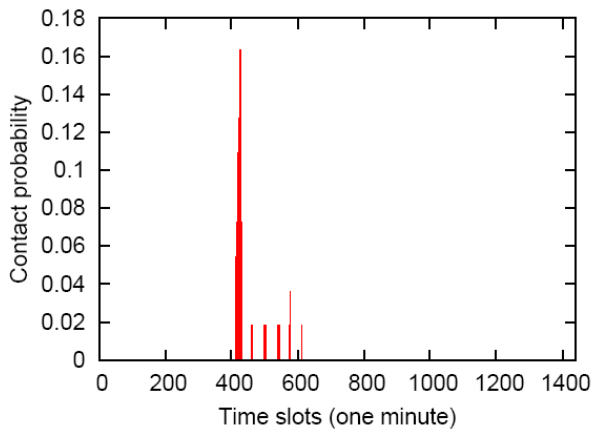


Virtualization vs. Slicing

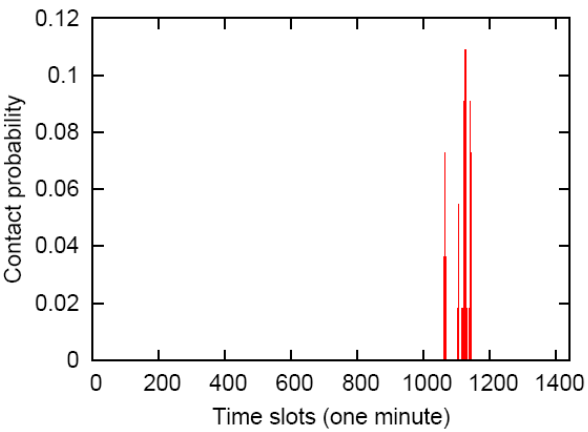
- Nature
  - Virtualization as a generalization
  - Slicing as a specialization
- Efficiency
  - Virtualization as a general purpose solution
  - Slicing as a special purpose solution
- Slicing as an art
  - Decompose the network into slices.
  - Apply the optimal greedy strategy to each slice.
- Other optimization problems
  - Minimum transmission delay

Graph Extension: Probabilistic Contacts

- Challenges
  - Routing efficiency in probabilistic time-space graphs
- Definition (t_i,p)
  - p is the contact probability of two nodes in t_i



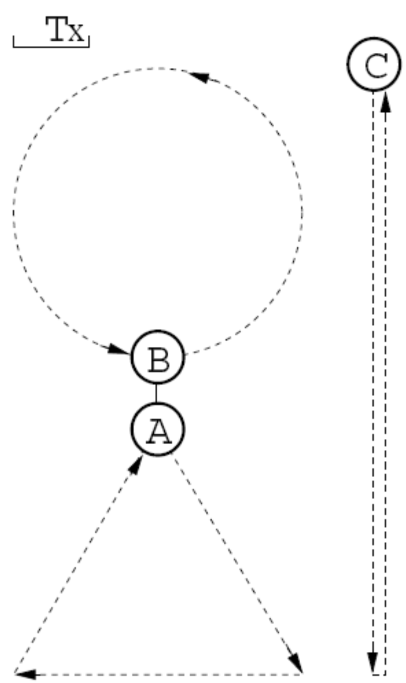
(a) Shifts 01/AM & 03/AM

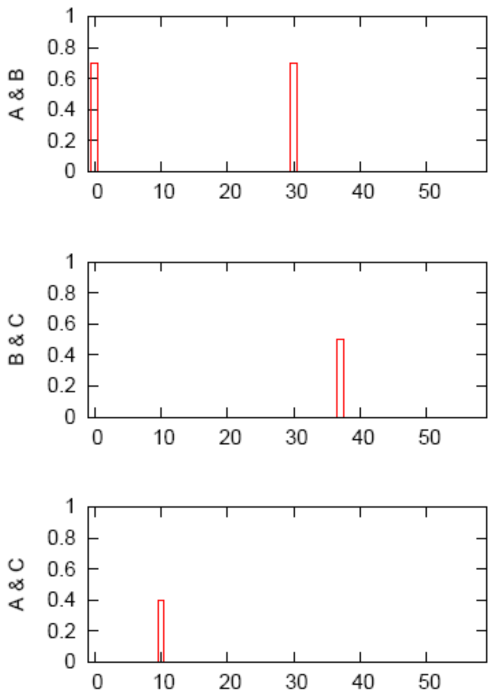


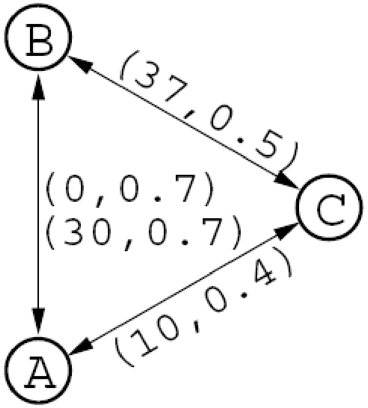
(b) Shifts 32/PM & 21/EVE

Probabilistic Time-Space Graph

- A common motion cycle **T** (=60)

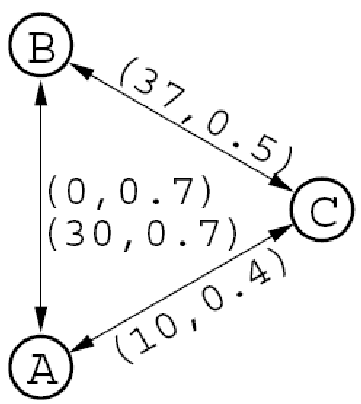




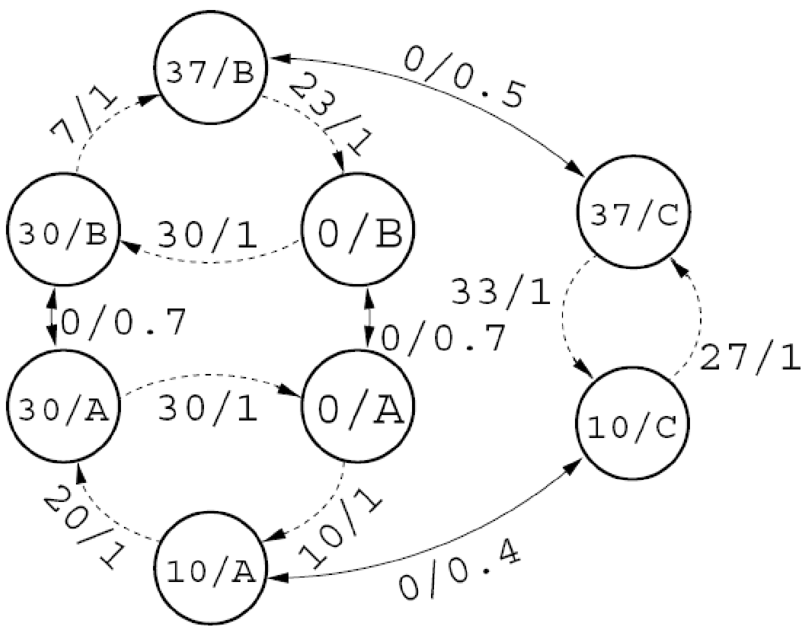


Probabilistic State-Space Graph

- Remove time dimension
- Links are labeled: d / p^{max} (delay/max transition probability)



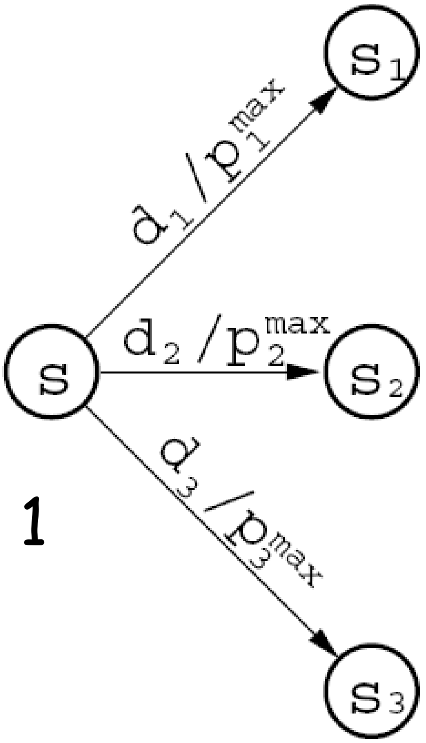
(a) G



(b) G'

Iterative Process

- Extended Bellman-Ford
  - Step t+1 based on step t
  - Ordering of neighbors based on distance to the destination
  - Selecting $p_i \leq p_i^{max}$ such that $\sum_i p_i = 1$

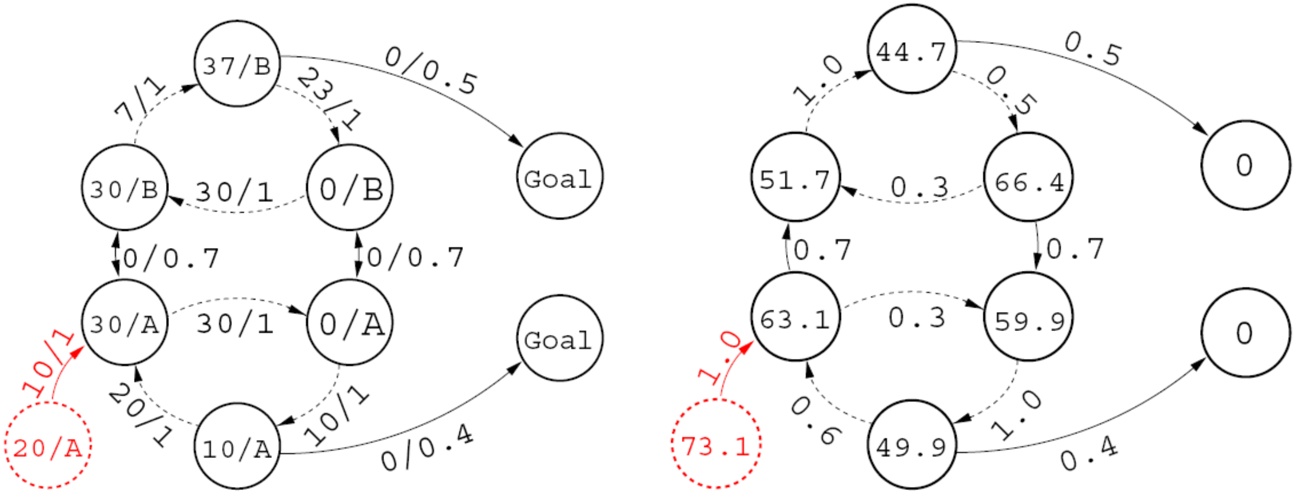


$$v_s^{t+1} \leftarrow \min_{p_1, p_2, p_3 \ldots} \{p_1 \times (d_1 + v_{s_1}^{t}) + p_2 \times (d_2 + v_{s_2}^{t}) + p_3 \times (d_3 + v_{s_3}^{t}) + \ldots\}$$

Expected Minimum Delay (EMD)

- Using the EMD as the delivery probability metrics

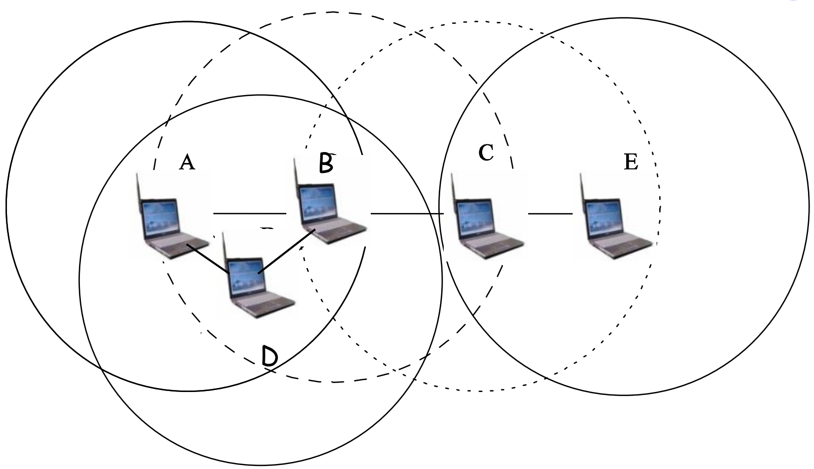
  (Liu and Wu, MobiHoc 2008)



Other Challenges

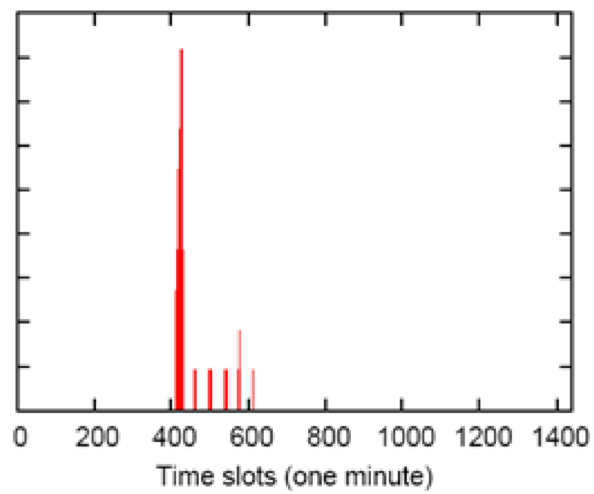
- Mobility
- Connectivity
- Complexity
- Bandwidth/storage
- Latency
- Security
- Multicast/broadcast

(Wu and Wang, MASS 2010)



Complexity

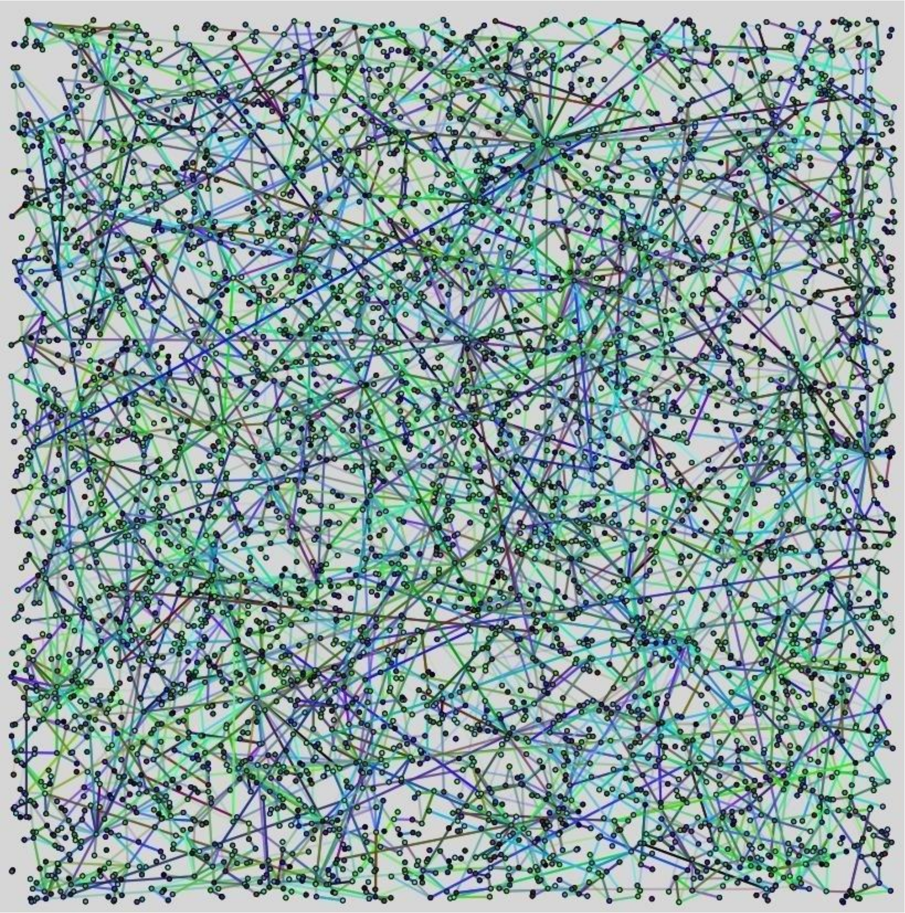
- Discrete contact graphs
  - Lossless translation method
    - Graph virtulization
      (state explosion issue)
  - Lossy comprehension method
    - Contact information compression
- Probabilistic contact graphs
  - Lossy compression through clustering (Liu and Wu, ToN 2011)
  - One cluster per contact event
    - Hierarchical clustering
    - K-mean partitioned clustering



4. A Bigger Picture

- Data Management
  - In-network processing
  - Tradeoffs among communication, computation, and storage
  - Cloud computing & data centers
- Theory
  - Rigorous model and scaling properties
  - Swarm intelligence
- Social Networks
  - Small-world (six degrees of separation)
  - Scale-free networks (power-law)

Multi-disciplinary

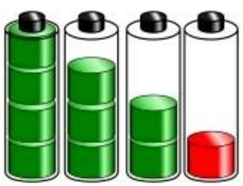


Cloud Computing & Data Centers

- Cloud computing: software, platform, infrastructure
- Wireless and mobile applications in cloud computing (MobiCom'11, Cambridge) and data centers (SIGCOMM'11, Microsoft Research)

Mobile and Wireless

- Portable: Follow us everywhere
- Rich in context
- Connect mobile and cloud computing

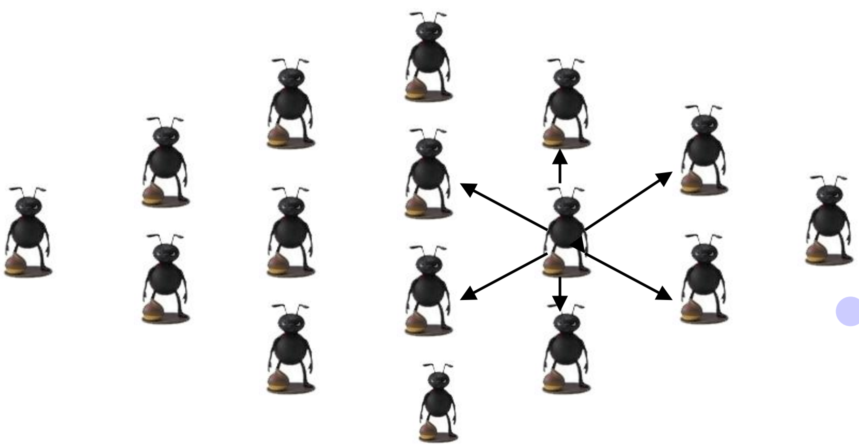


Cloud computing

- Huge data centers
- Rich in resources
- Lots of reliable power, storage, and computing capabilities

Swarm Intelligence: Self-Organization

- Local view
  - P2P and simple interaction (mostly local)
- Global functionality
  - Adaptive, robust, and scalable



- Four paradigms

(Prehofer and Bettstetter, Comm. Magazine, July 2005)

  - P_1: Local interactions with global properties
  - P_2: Minimizing the maintained state
  - P_3: Adaptive to changes (self-healing)
  - P_4: Implicit coordination

- CDS construction for four paradigms (Wu, Yang, and Dai, TC'09)

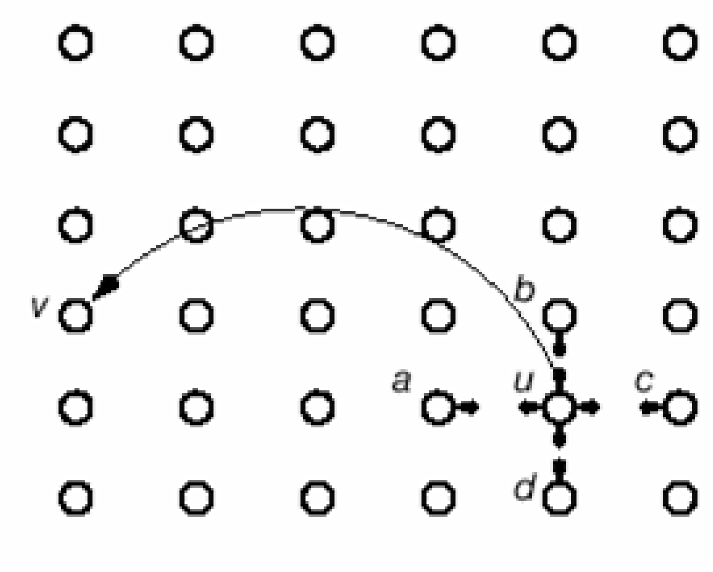
Social Networks: Small World

- Six degree of separation: short path length
- Clustered: neighbors of a node being neighbors
- Local CDS processing work well: nodes with unconnected neighbors are marked

Average node degree: 10

P	CC	CDS
0.01	0.96	1.05
0.02	0.95	1.08
0.03	0.91	1.1

- P: percentage of rewiring
- CC: clustering coefficient



Questions