CIS 9590 Ad Hoc Networks (Part IV)

Jie Wu and Shaojie Tang

Department of Computer and Information Sciences

Temple University

Philadelphia, PA 19122

Table of Contents

- Introduction
- Infrastructured networks
  - Handoff
  - location management (mobile IP)
  - channel assignment

Table of Contents (cont'd.)

- Infrastructureless networks
  - Wireless MAC (IEEE 802.11 and Bluetooth)
  - Ad hoc routing protocols
  - Multicasting and broadcasting
  - Coverage
  - Security

Table of Contents (cont'd.)

- Infrastructureless networks (cont'd.)
  - Power optimization
  - Localization
  - Network coding and capacity
- Applications
  - Sensor networks
  - Cognitive radio networks
  - Pervasive computing
- Delay tolerant and social networks
- Sample on-going projects

Overview of Cognitive Radio Networks (CRNs)

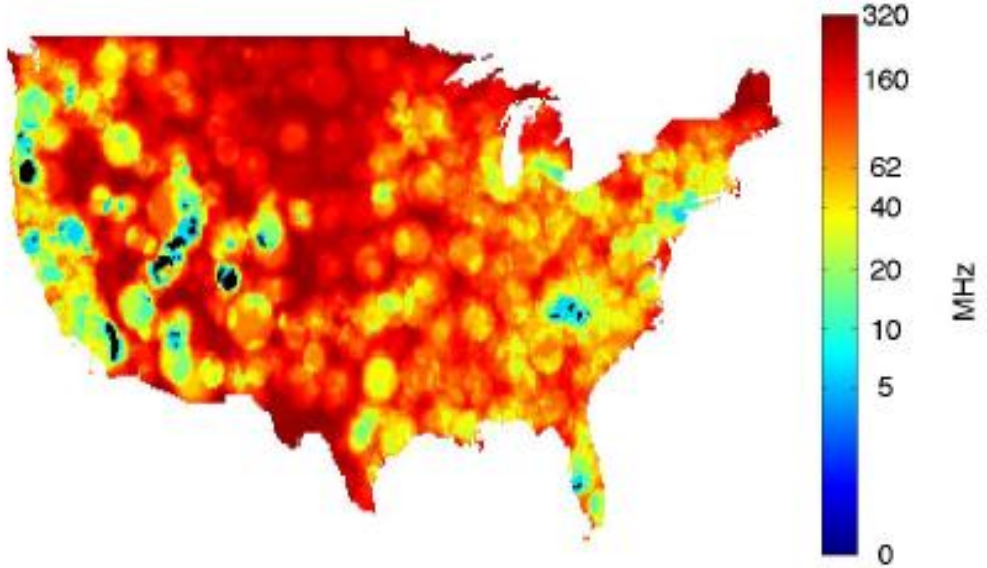


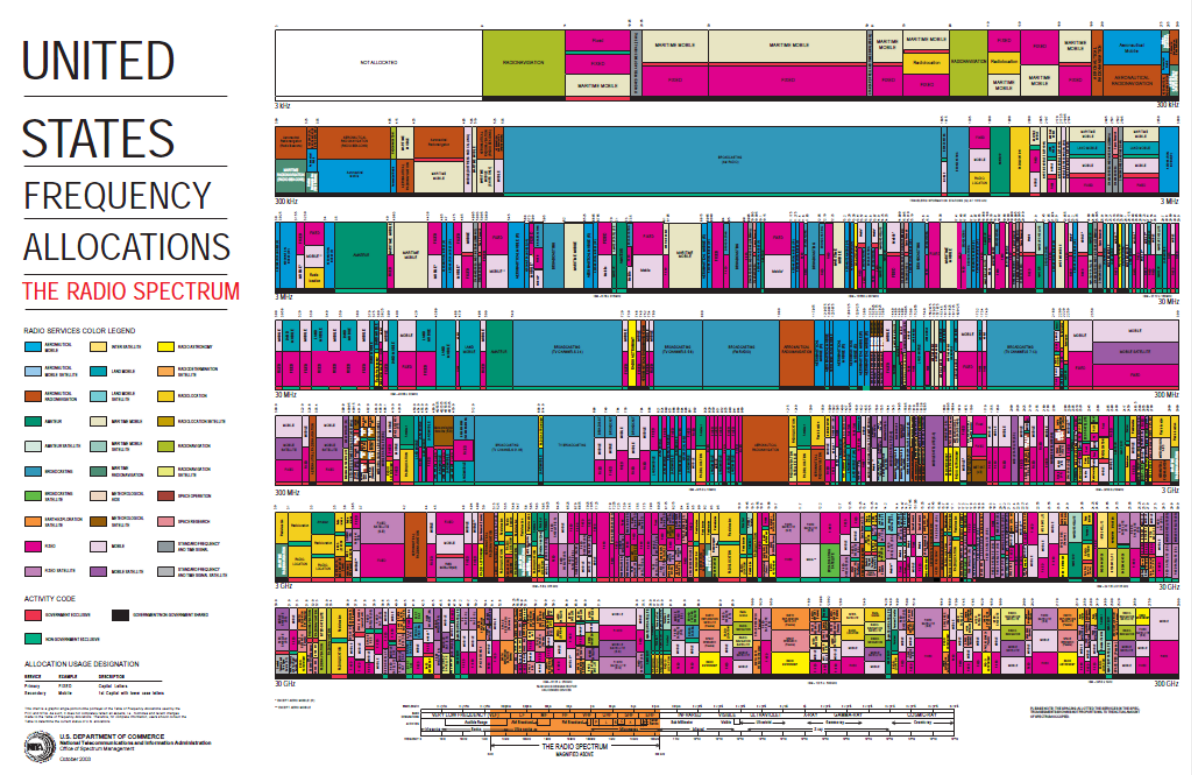
Fig. 1. A color-coded map of the continental USA with an estimate of the number of white-space channels allowed by the FCC's Nov 4th, 2008 ruling accounting for both co-channel and adjacent-channel protection. This map simply plots by latitude and longitude and does not use any other projection. The color legend is to the right and for comparison, the 62 MHz number is marked so that the white-space opportunity can be compared to the number of channels opened up in the 700MHz proceeding for wireless data providers.

Dynamic Spectrum Access (DSA)

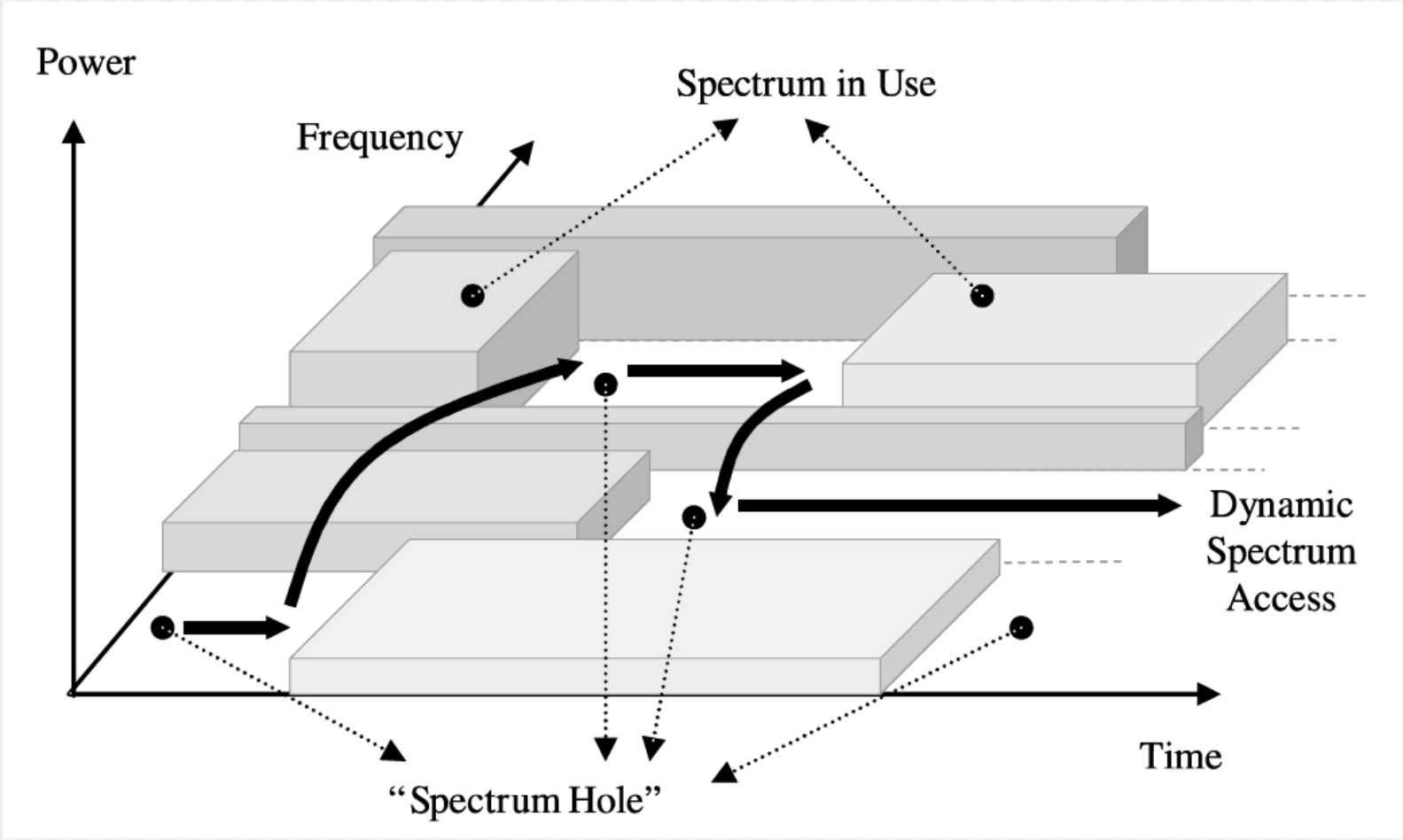
- PROBLEM : Underutilized spectrum resources due to fixed assignment strategy
- IDEA : Utilizing "white spaces" in the spectrum without disturbing licensed users [1]
- CHALLENGE :
  - Sensing the environment, detecting "white spaces"
  - "white spaces" frequently change by time and space
  - Changing radio parameters on the fly

Frequency Allocation Charts

- In Turkey: http://www.tk.gov.tr/Duzenlemeler/teknik/marfl/_marfl1.asp



White Space Concept



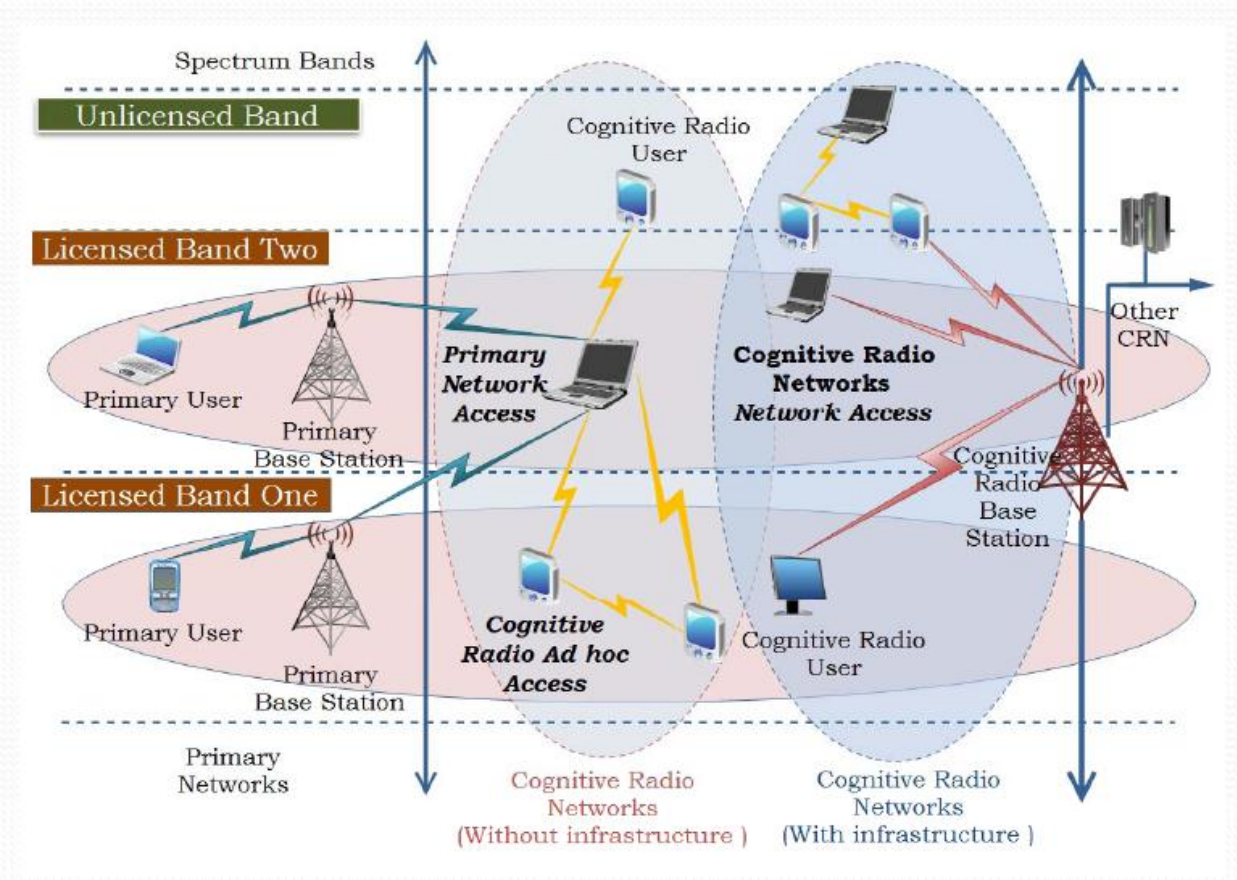
Cognitive Radio (CR) Concept

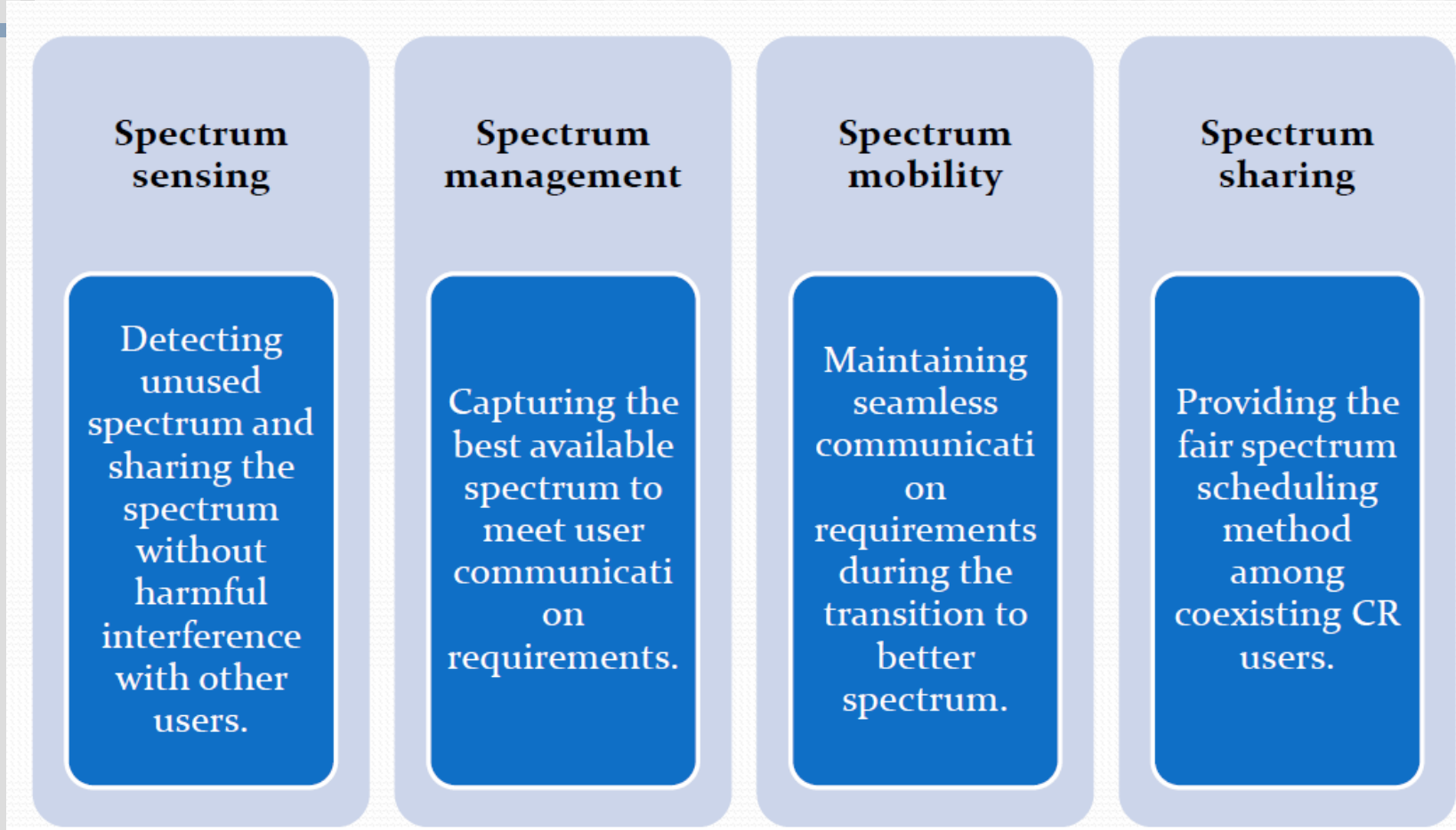
- Initially proposed by Mitola [2]
- Next step evolution of *Software Defined Radio (SDR)*
- *Cognitive Radio (CR):* Intelligent devices that can [2]:
  - Sense and autonomously reason about their environment
  - Adapt their communication parameters accordingly
  - Realize DSA concept

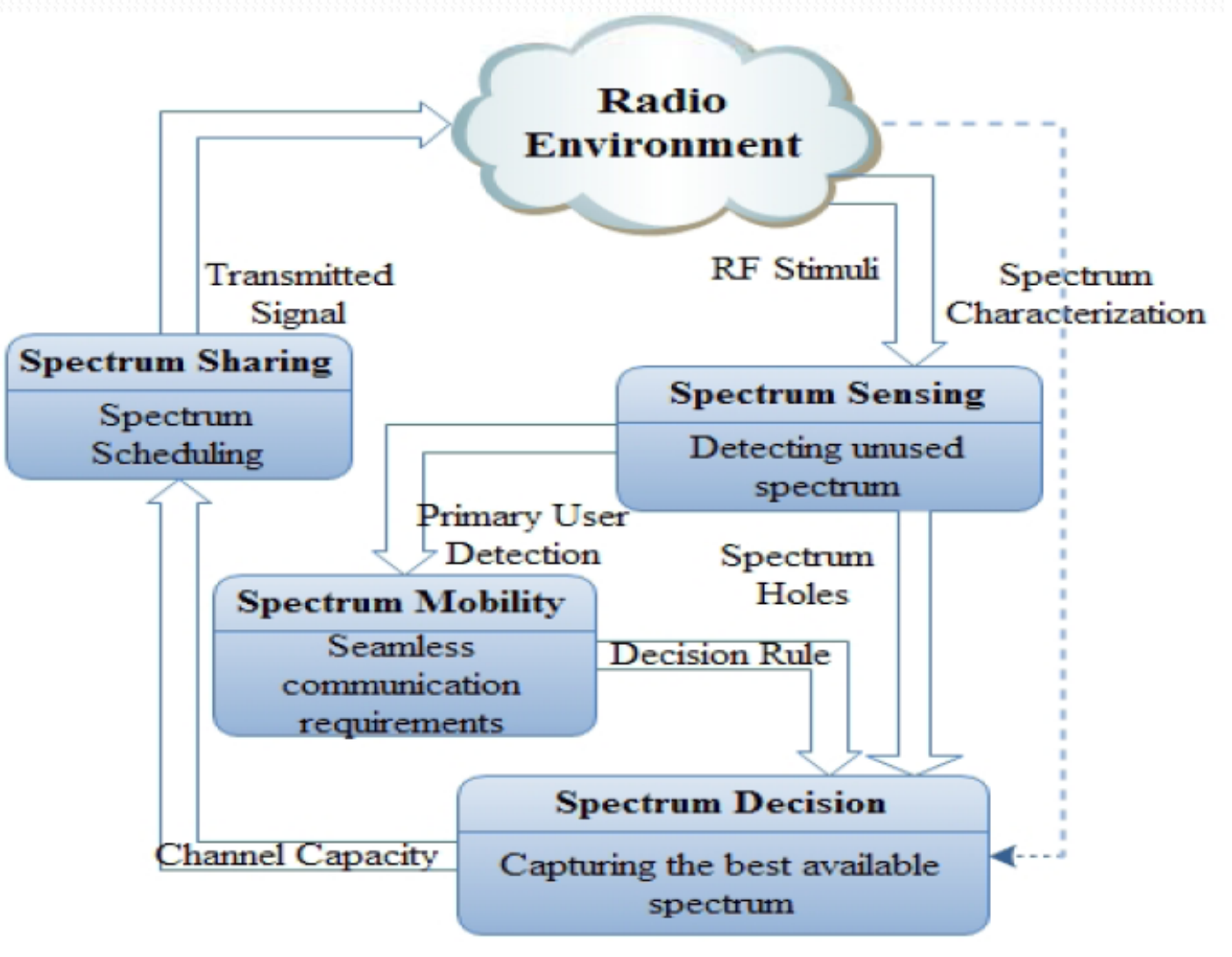
Terminology

- Primary User (PU):
  - Licensed user
  - Has exclusive rights for the spectrum
- Secondary User (SU):
  - Unlicensed user
  - Opportunistically utilizes the *white spaces*
  - Has to vacate the spectrum band as soon as a PU appears
  - Also called *cognitive user*

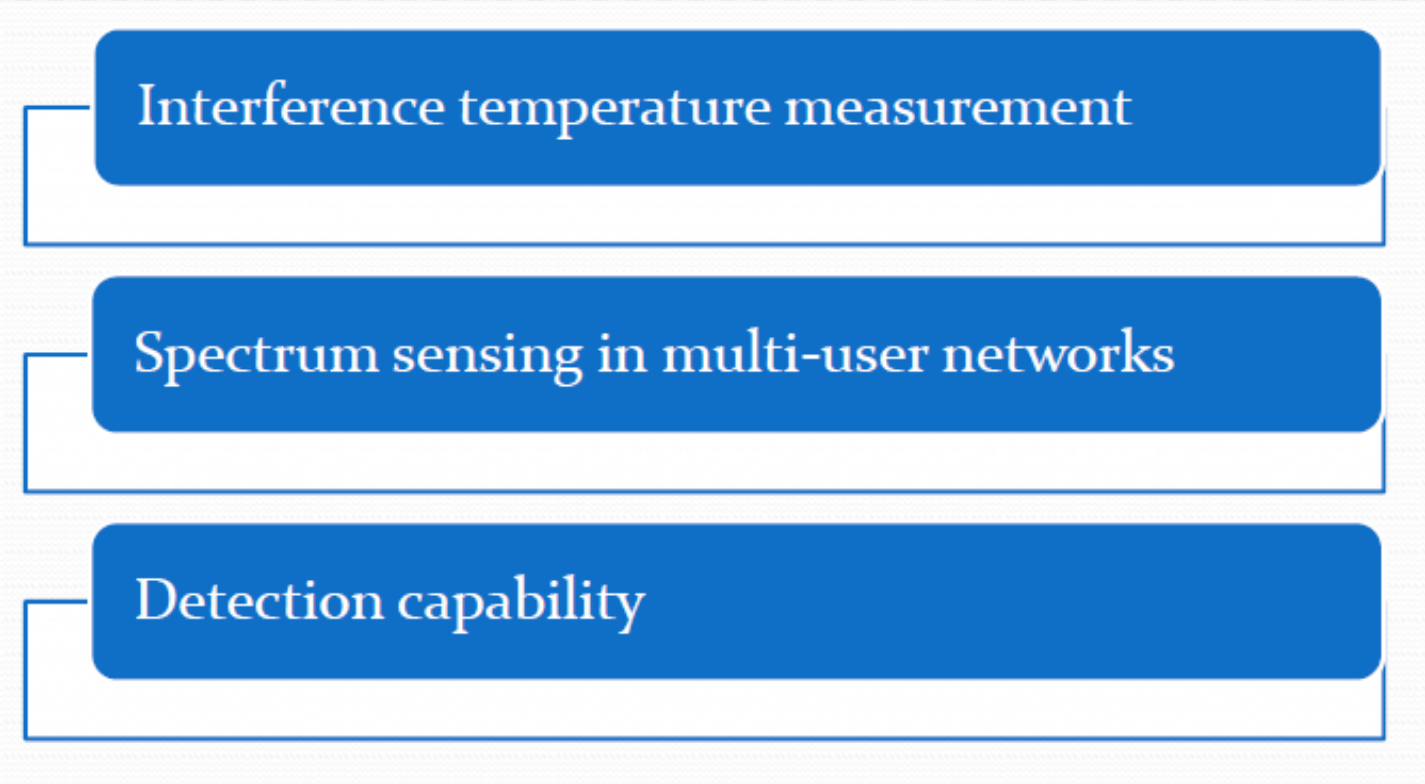
IEEE 802.22 Network Architecture




Spectrum sensing
Detecting unused spectrum and sharing the spectrum without harmful interference with other users.
Spectrum management
Capturing the best available spectrum to meet user communicati on requirements.
Spectrum mobility
Maintaining seamless communicati on requirements during the transition to better spectrum.
Spectrum sharing
Providing the fair spectrum scheduling method among coexisting CR users.

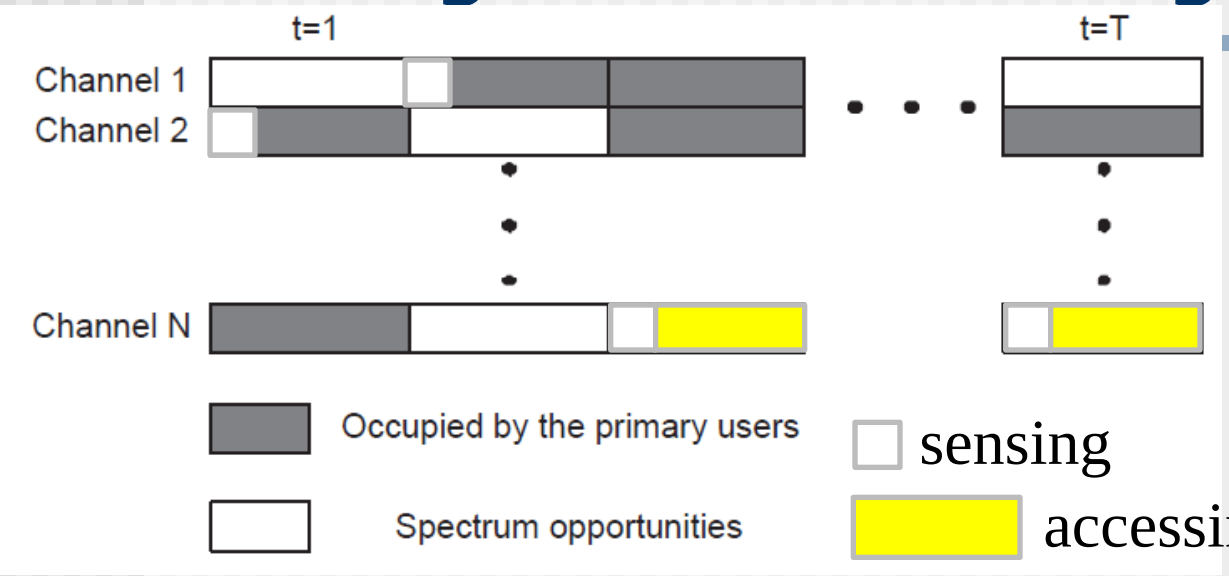

Radio Environment
Transmitted Signal
RF Stimuli
Spectrum Characterization
Spectrum Sharing
Spectrum Scheduling
Spectrum Sensing
Detecting unused spectrum
Primary User Detection
Spectrum Holes
Spectrum Mobility
Seamless communication requirements
Decision Rule
Spectrum Decision
Capturing the best available spectrum
Channel Capacity

Challenges in Spectrum Sensing



Almost Optimal Dynamically-Ordered Multi-Channel Accessing for Cognitive Radio Network

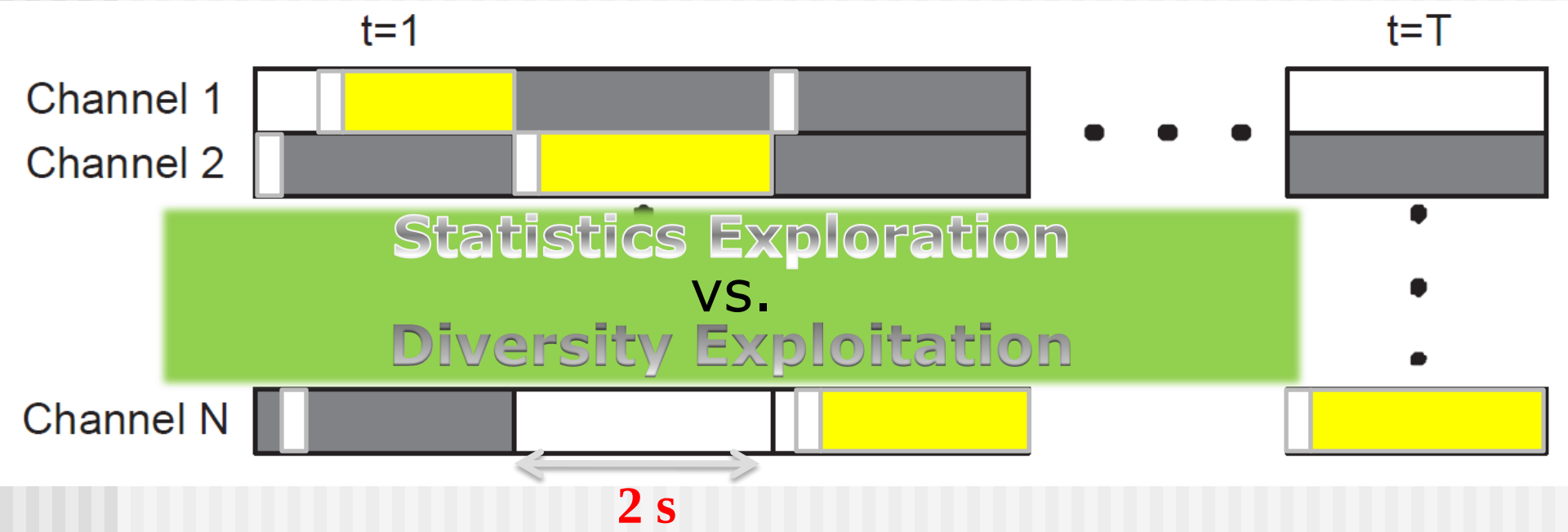
Background: existing work



- Online channel selection
- MAB problem is used to model DSA
  - Each arm is a potential/candidate channel

SSA: Sequential Sensing and Accessing

- Channel sensing time << slot time
- Multichannel diversity



Online SSA

- How to select an channel sensing order for SSA process, when statistics is unavailable?
  - **Online learning of optimal sensing order**
    - **Challenges**
      - Exponential number of sensing orders
      - Non-linear correlated rewards
      - Randomized number of channels observed in each slot

Outline

- System Model and Problem Formation
- Our Schemes
- Simulation Results
- Some work not included in this paper

System Model and Problem Formation

- N Channels in set $S = \{1, 2, \ldots, N\}$
- Channel availability is in {0,1} & Channel idle probability is initially unknown
- CAT transmission model:
  - Constant access time slot T
  - Each channel sensing time τ_s
  - Maximum sensing steps in a slot

$$K = \min\left(N, \left\lfloor \frac{T}{\tau_s} \right\rfloor\right)$$

SMAB Formulation

- Each sensing order is an arm $\Phi_m = (\psi_1^m, \psi_2^m, \ldots, \psi_K^m)$
- The total number of arms $M = \binom{N}{K} K!$
- Expected reward by choosing order Φ_m

$$\mu_m = E\left[r_{\Phi_m} \right] = \sum_{k=1}^{K} \left\{ (1 - k\alpha)\theta_{\psi_k^m} \prod_{\kappa=1}^{k-1} \left(1 - \theta_{\psi_\kappa^m}\right) \right\}$$

SMAB Formulation：Goal

- Max reward over L steps $\max \lim_{L\to\infty} \sum_{j=1}^{L} \mu_{\pi(j)}$
- Min the regret $\min \lim_{L\to\infty} \rho_{\pi}(L) = L\mu^{*} - \sum_{j=1}^{L} \mu_{\pi(j)}$

Solution1：UCB1

- Update the statistics on the reward of arms
- At the very beginning, choose each sensing order only once
- After that, select the order Φ_m that maximizes

$$\hat{\mu}_m + \sqrt{\frac{2\log j}{n_m}}$$

Solution1：UCB1

- Theorem 1: The expected regret of SSA under policy UCB1 is at most

$$\left[8\sum_{m:\mu_m<\mu^*}\left(\frac{\log L}{\xi_m}\right)\right]+\left(1+\frac{\pi^2}{3}\right)\left(\sum_{m:\mu_m<\mu^*}\xi_m\right)$$

where $\xi_m=\mu^*-\mu_m$.

- Regret is in $O\left(N^K\log L\right)$ ($M=\binom{N}{K}K!$)
- Storage complexity: $O\left(N^K\right)$

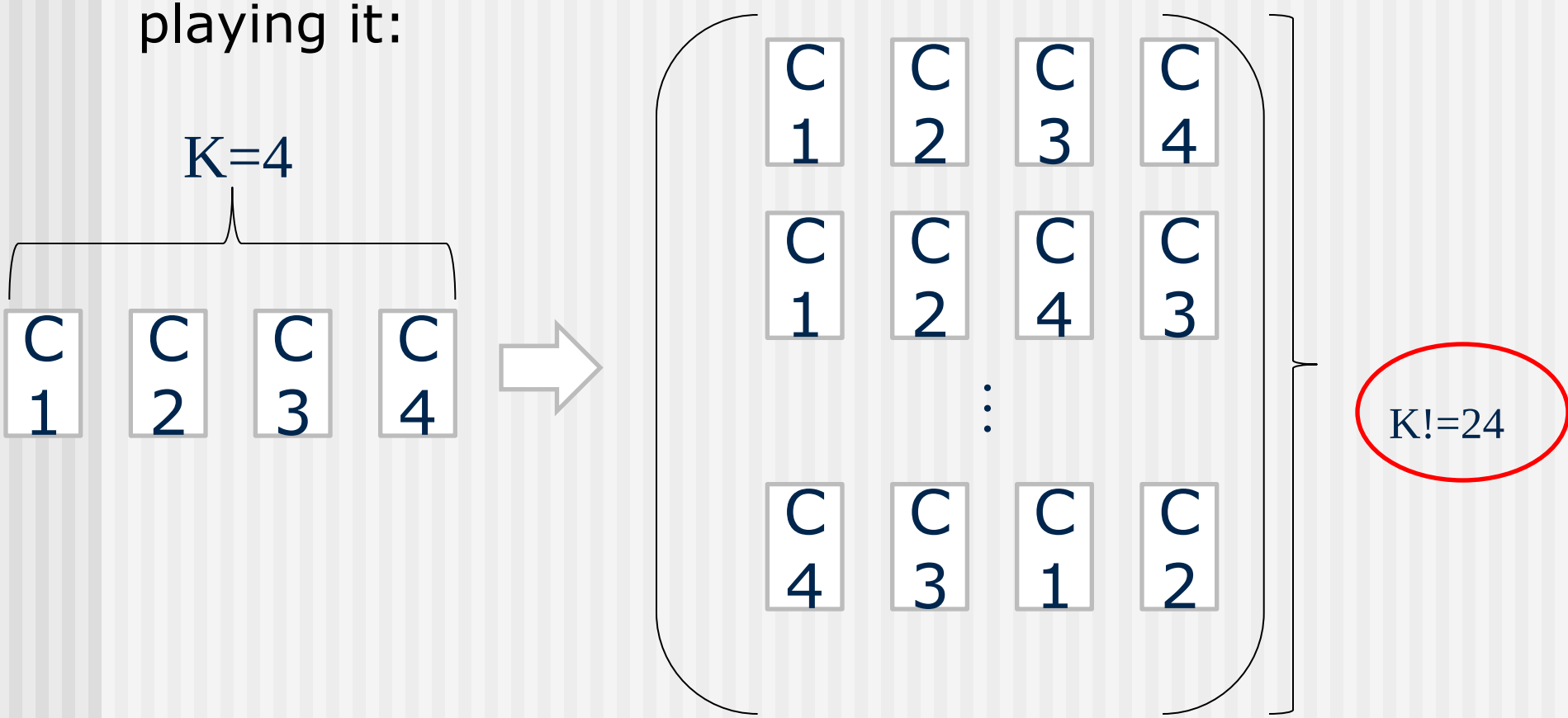
Solution2：UCB1-VS

- Arms are correlated
- Exploring dependency: learning information about multiple arms by playing one arm

UCB1 –VS :
UCB1 with Virtual Sampling

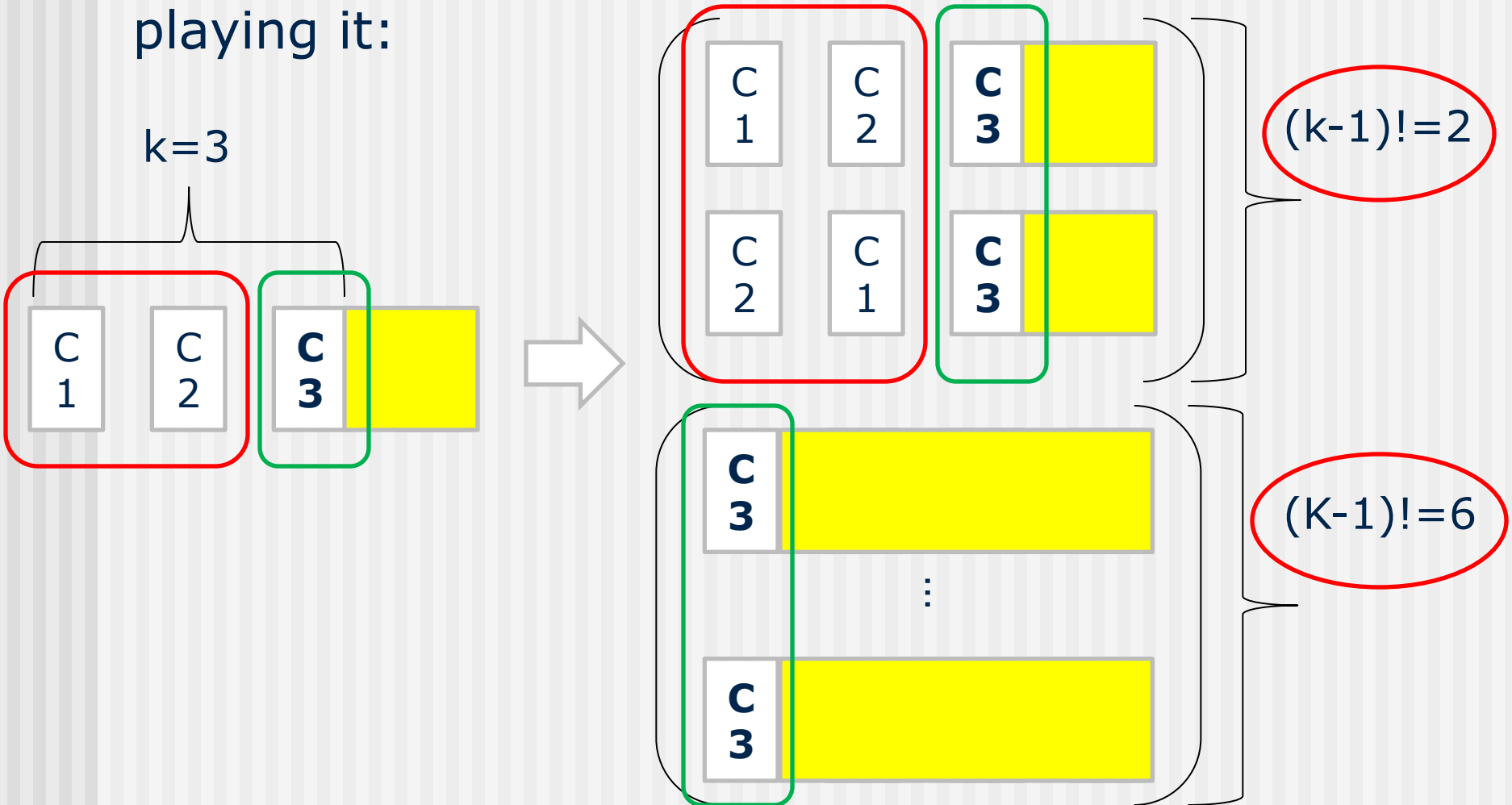
Solution2：UCB1-VS

- Virtual Sampling: update the statistics without playing it:



Solution2：UCB1-VS

- Virtual Sampling: update the statistics without playing it:



Solution2：UCB1-VS

- The advantage of UCB1 is preserved
- At least $(K-1)!$ arms are updated with one playing
- Storage complexity: $O(N^K)$

Solution3：SCB

- Aforementioned methods: arm selection based on arm-specific statistics
- Exponential regret and storage overhead
- Dynamically determine sensing order based on channel-specific statistics

SCB: Sequencing Confidence Bound

Solution3：SCB

- Update the channel idle probability estimation
- For each channel: $\theta_i^u(j) = \hat{\theta}_i(j) + \sqrt{\frac{2\log j}{n_i^s(j)}}$
- At the very beginning, run SSA until all channels are sensed at least once;
- After that, in slot j select the order Φ_m that maximizes

$$\sum_{k=1}^{K}\left\{(1-k\alpha)\theta^u_{\psi_k^m}(j)\prod_{\kappa=1}^{k-1}\left(1-\theta^u_{\psi_\kappa^m}(j)\right)\right\}$$

Solution3：SCB

- Theorem 2: The expected regret of SSA under policy SCB is at most

$$\Pi(L)K\left[N-\frac{K+1}{2}-\frac{\alpha(K+1)(3N-2K-1)}{6}\right]$$

where $\Pi(L)=\frac{8\log L}{\Delta_{min}}+\left(1+\frac{\pi^2}{3}\right)\Delta_{max}$, $\Delta_{min}=\min_{i,j}\left|\theta_i-\theta_j\right|$ ($i\neq j$), and $\Delta_{max}=\max_{i,j}\left|\theta_i-\theta_j\right|$.

$$O\left(NK\log L\right)$$

Solution3：SCB

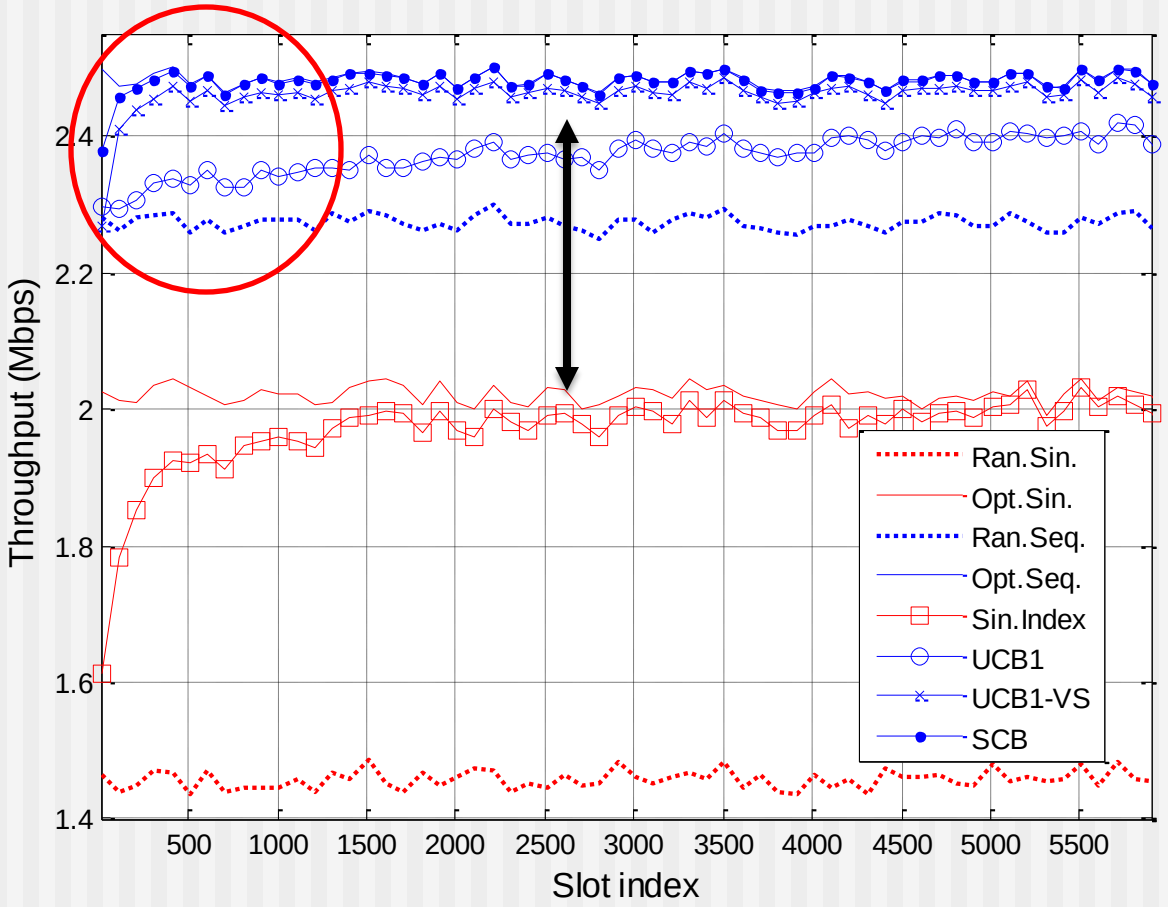
Solution	**Regret**	**Storage**
UCB1	$O\left(N^K \log L\right)$	$O\left(N^K\right)$
SCB	$O\left(NK \log L\right)$	$O\left(N\right)$

Simulation

- Scenario:
  - Number of channels N=3, normalized sensing cost $\alpha = 0.1$, channel idle probability $\theta_i \in [0\ \ 1]$, channel received SNR mean $\gamma = 10$ dB.
- Strategies:
  - **UCB1, UCB1-VS, SCB**
  - **Ran.Sin.**: choose a random channel in each slot
  - **Opt.Sin.**: always choose the best channel with highest θ
  - **Sin.Index**: using UCB1 to choose the best channel online
  - **Ran.Seq**.: choose a random sensing sequence
  - **Opt.Seq.**: always choose the best sensing sequence

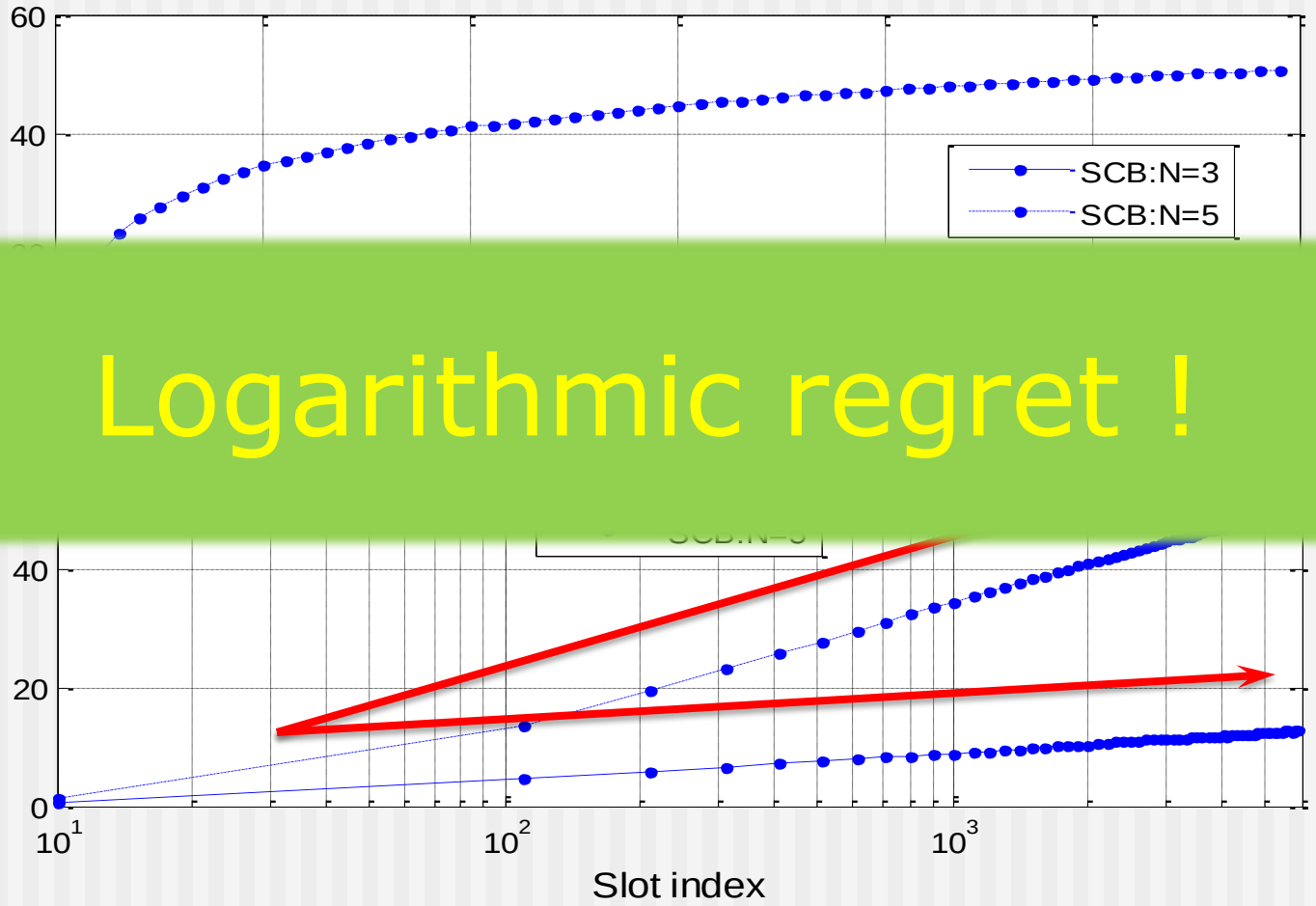
Results

- Throughput



Results

- Regret



Logarithmic regret !

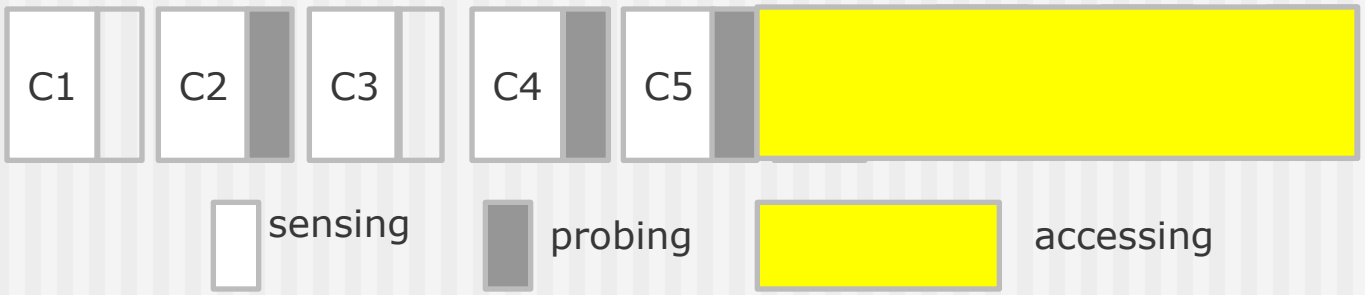
Results

- Impact of environmental factors
  - Strategies
    - SCB
    - Single Index
  - Factors:
    - number of channels and sensing cost
  - Performance metrics
    - Achievable throughput
    - Learning time for achieving 90% of the maximum achievable throughput

α=0.05
α=0.1
α=0.2
Sin.Index
SCB
Achievable throughput (Mbps)
Slots for achieving 0.9-LP
Number of channels
Higher & Faster !!!

Some work not included

- SSPA: Sequential Sensing/Probing and Accessing



- Sense and probe channel for acquiring channel availability and link quality
- Rayleigh channel's SNR q is exponentially distributed with an unknown mean γ .

Some work not included

- Reward: If access in the k^{th} sensing/probing step in the j^{th} slot using Φ_m

$$r_{\Phi_m}(j) = (1 - k\beta) a_{s_m^k}(j) \ln\left(1 + q_{s_m^k}(j)\right)$$

- Goal: dynamically determine the sensing order as well as accessing rule, guaranteeing optimal convergence as well as minimizing regret.

Some work not included

- Solution: SCB-OSP
- Integrate the core idea of SCB into the optimal stopping analytical model
  - Update estimations on channel idle probability & SNR mean
  - For each channel:

$$\hat{\theta}_i^u(j) = \min\left\{1, \hat{\theta}_i(j) + \sqrt{\frac{-\log\delta}{2n_i^s(j)}}\right\}$$

$$\hat{\gamma}_i^u(j) = \min\left\{q_{max}, \hat{\gamma}_i(j) + q_{max}\sqrt{\frac{-\log\delta}{2n_i^p(j)}}\right\}$$

Some work not included

- Solution: SCB-OSP (cont.)
  - Run SSPA until all channels are probed at least once
  - After that, in slot j select the order Φ_m that maximizes $\hat{\Lambda}_1^{m,u}(j)$, which can be derived step by step using

where

$$\hat{\Lambda}_k^{m,u} = \hat{\Lambda}_{k+1}^{m,u} + (1-k\beta)\hat{\theta}^u_{\psi_k^m} e^{\frac{1}{\hat{\gamma}^u_{\psi_k^m}}} \mathrm{Ei}\left(1, \frac{e^{\frac{\hat{\Lambda}_{k+1}^{m,u}}{1-k\beta}}}{\hat{\gamma}^u_{\psi_k^m}}\right)$$

$$\hat{\Lambda}_K^{m,u} = (1-K\beta)\hat{\theta}^u_{\psi_K^m} e^{\frac{1}{\hat{\gamma}^u_{\psi_K^m}}} \mathrm{Ei}\left(1, \frac{1}{\hat{\gamma}^u_{\psi_K^m}}\right)$$

Some work not included

- Solution: SCB-OSP (cont.)
  - Corresponding accessing rule can be described as a sequence of SNR threshold $\left(\Gamma_1,\ldots,\Gamma_K\right)$, which is given by

$$\Gamma_k=\begin{cases} e^{\frac{\hat{\Lambda}_{k+1}^{m,u}}{1-k\beta}}-1, & 1\le k<K \\ 0, & k=K \end{cases}$$

  - If $q_{\psi_m^k}\left(j\right)\ge\Gamma_k$, then access the channel; otherwise, skip to sense another channel;

Some work not included

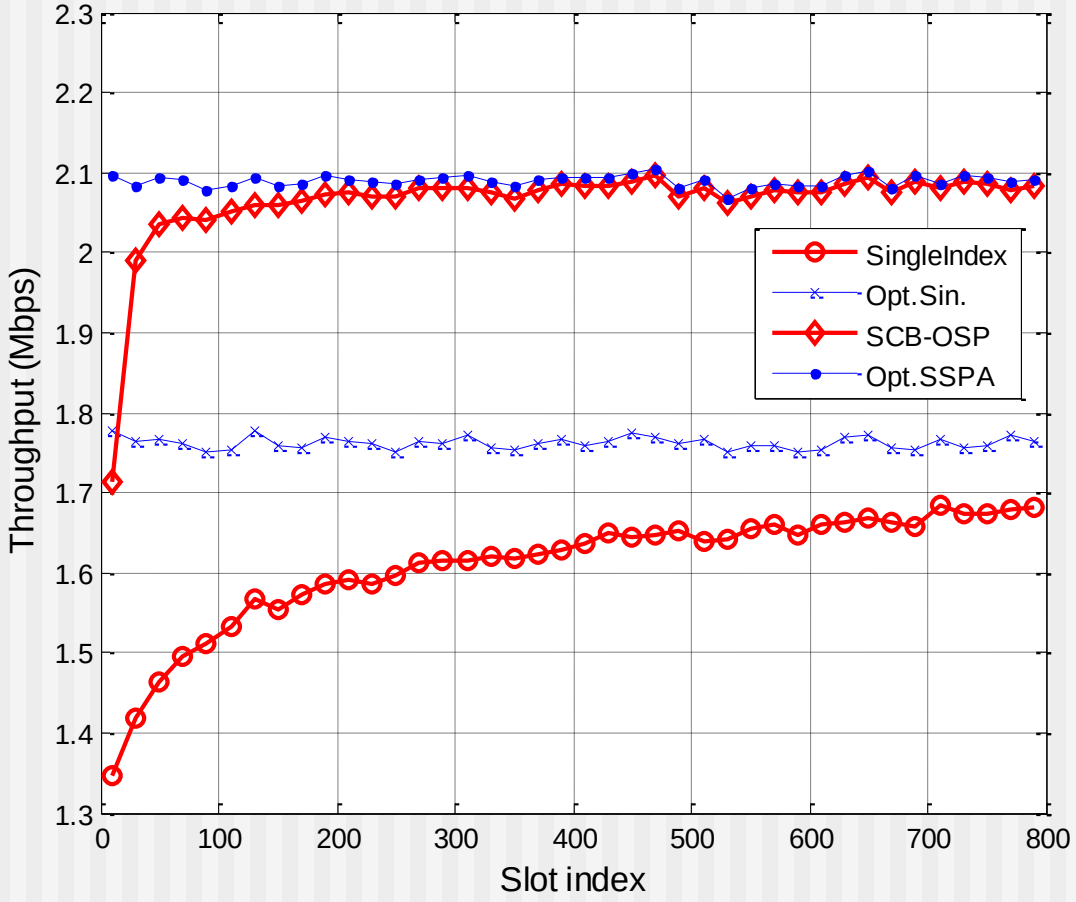
- **Theorem 3**: Using SCB-OSP, system converges to the throughput-optimal SSPA policy with probability at least $(1-\delta)^{2(N-1)}$; Particularly, when $\forall i : \theta_i < 1$, it converges to optimal SSPA with probability at least $(1-\delta)^{2(N-K)}$.

Some work not included

- Simulation
  - scenario
    - Number of channels N=3, normalized sensing/probing cost $\beta = 0.1$, channel idle probability $\theta_i \in [0\ \ 1]$, channel received SNR mean $\gamma_i \in [0\ \ 18]$ dB.
  - Strategies
    - **Opt.Sin.**: always choose the best channel with maximum expected throughput
    - **SingleIndex**: using UCB1 to choose the best channel online
    - **Opt.SSPA**: throughput-optimal SSPA strategy derived with perfect channel statistics
    - **SCB-OSP**: the proposed learning policy

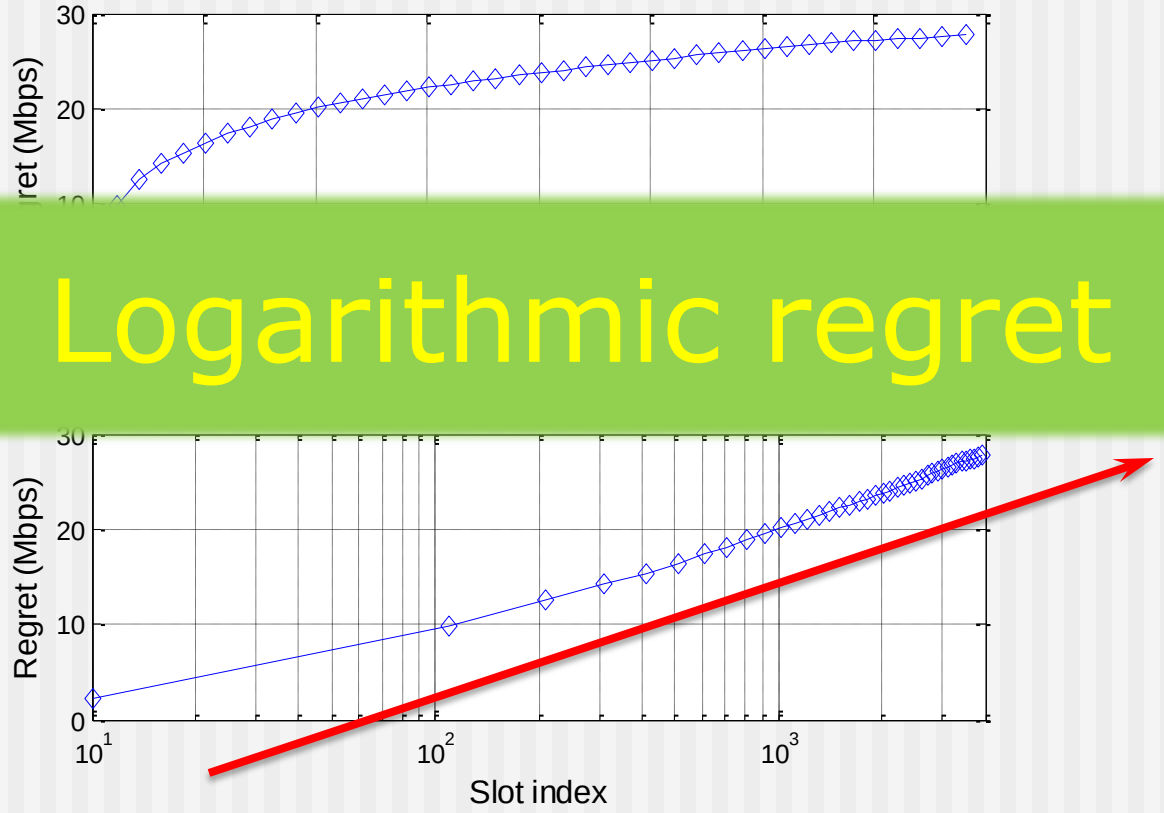
Some work not included

- Results:



Some work not included

- Results:



Logarithmic regret !

Conclusion

- Online learning of sequential channel sensing, probing and accessing
  - Jointly optimize: statistics learning & diversity exploitation
  - Jointly exploit: channel availability & link quality

Security Issues

- Protection from various protocol attacks.
- Application attacks.
- Unauthorized user introduction.
- Unauthorized access to system data.
- Denial of services (DOS) and Distributed Denial of services attacks (DDOS).

SimpleMAC: Jamming-Resilient MAC Protocol

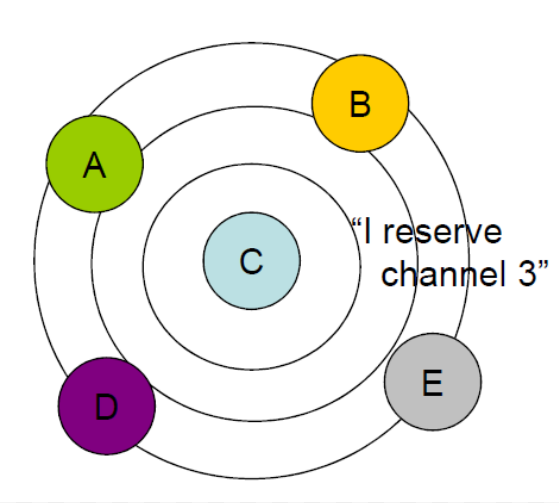
Sang-Yoon Chang, Yih-Chun Hu, Nicola Laurenti

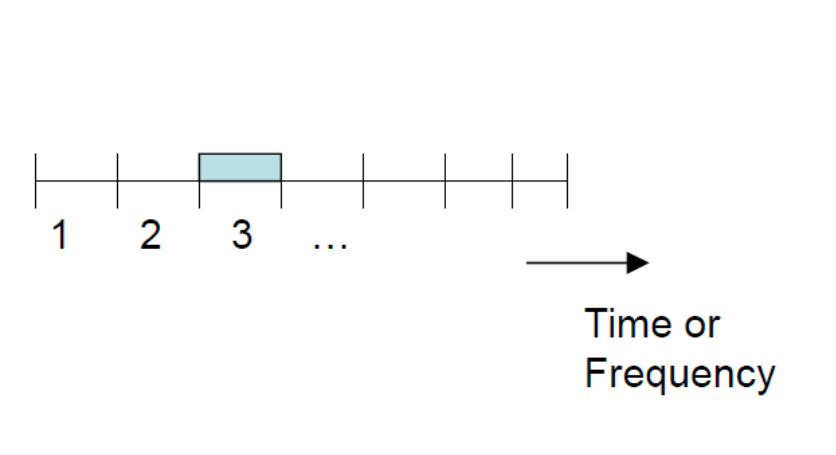
MobiCom 2012

*Slides are borrowed from authors.

MAC

- Multiple users competing for shared medium
- Channel use coordination
- Control messages to reserve channels
- Used in context of WiFi, WiMax, Bluetooth, OOB





MACs Built For Collaborative Nodes

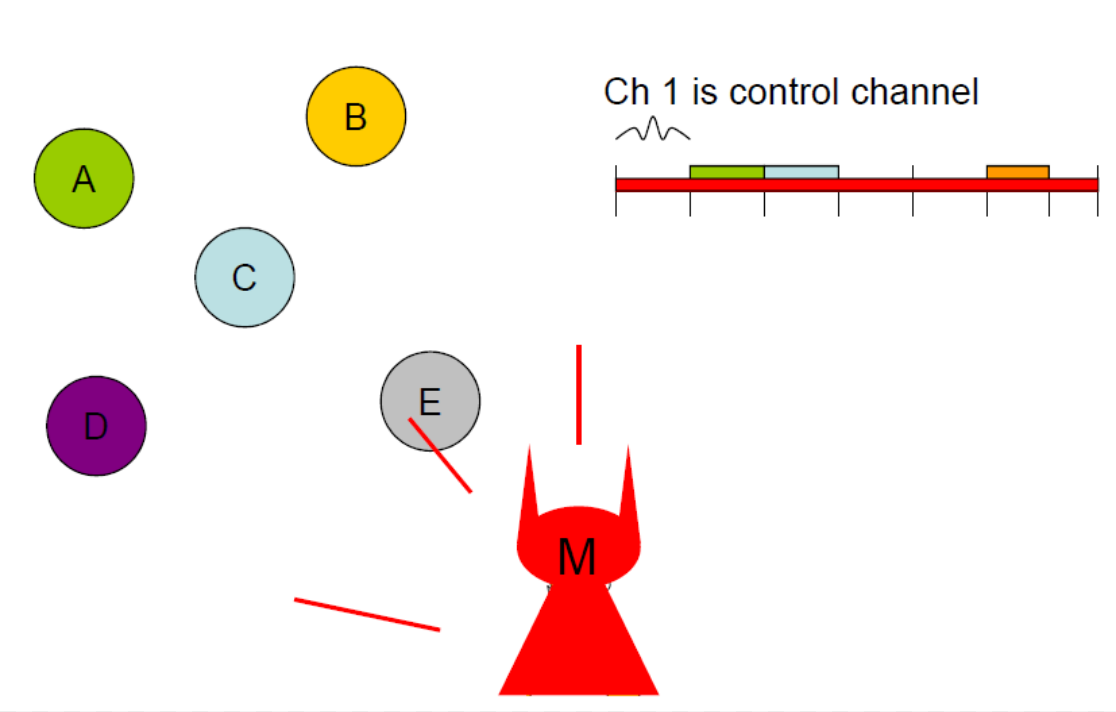
- Among collaborative nodes, MACs help in all but the most noise-limited regimes
- If all nodes are *selfish, the Nash equilibrium is to* transmit full wideband all the time, without MAC
  - § Selfish user does not benefit from diverging from Nash equilibrium unilaterally
  - § Tragedy of the commons
- Existing MAC protocols fail completely *when attacked*

Jamming Model

- Jammers are ***adversarial***
- Jammers are ***power-limited***
- Jammers might be ***insiders and may collude***
- Jammers can perform ***reactive jamming***
- Jammers' strategy can be ***mixed and dynamic***
- Threat model:
  - § MAC-oblivious Attack #0
  - § MAC-aware Attack #1, #2

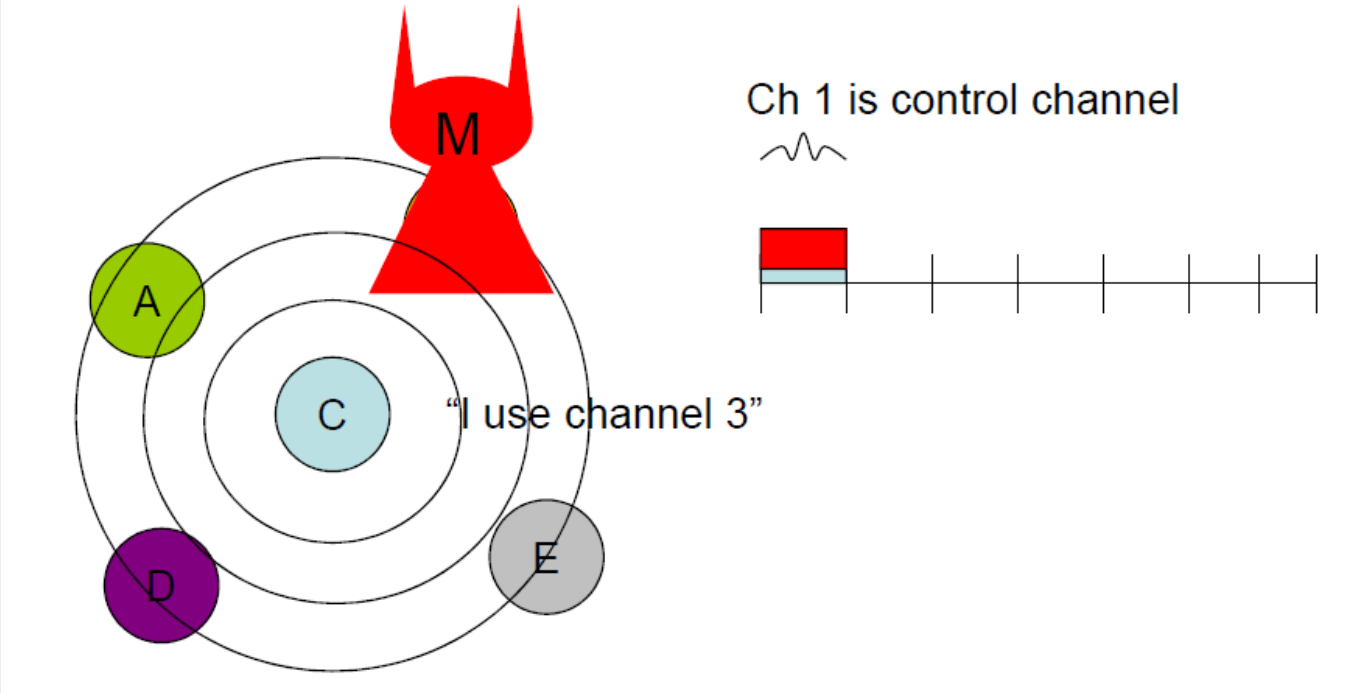
Attack #0: Jamming All Channels

- No need for insider information



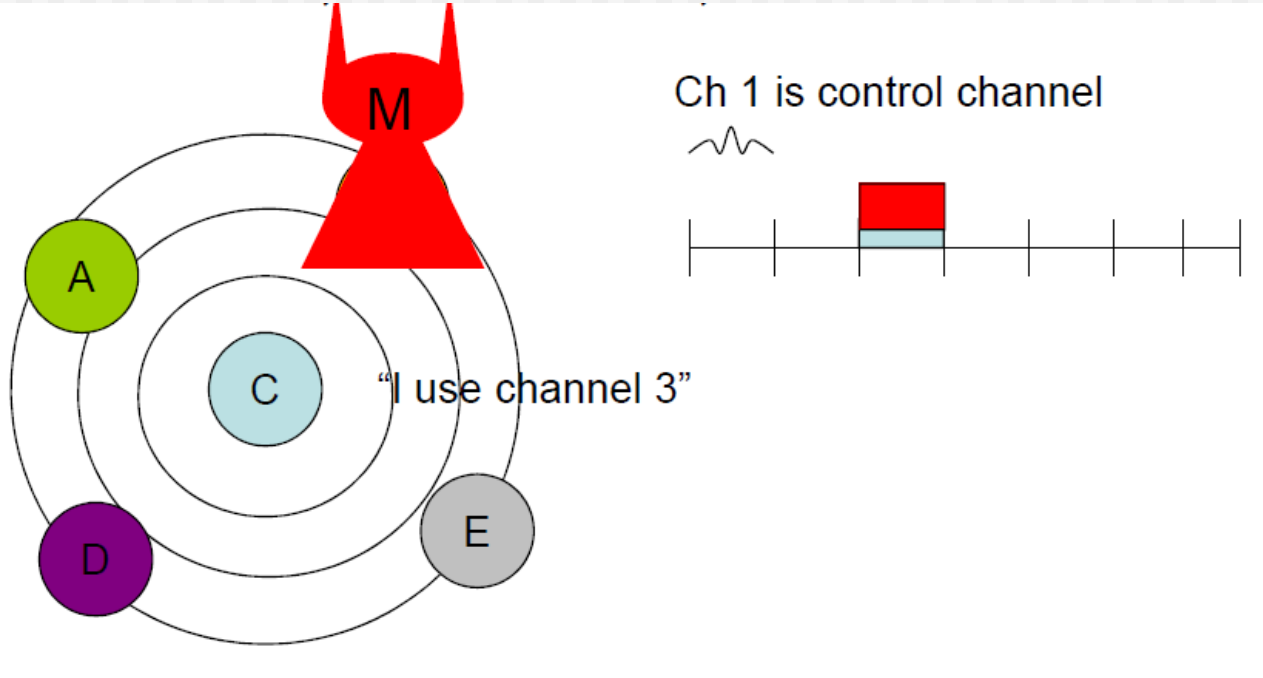
Attack #1: Control Channel Jamming

- Common control channel has known location



Attack #2: Jamming-Relevant Information

- Attackers know which channel is used for data transmission, for instance, channel 3 for user C



Attack #2: Jamming-Relevant Information

- Channel coordination introduces vulnerabilities:
  - § Attack #1: known common control channel
  - § Attack #2: broadcast the use of data channel
- Users can simply disable channel coordination (attackers can still perform Attack #0)
  - § This is the Nash equilibrium and our *baseline strategy*

Contributions

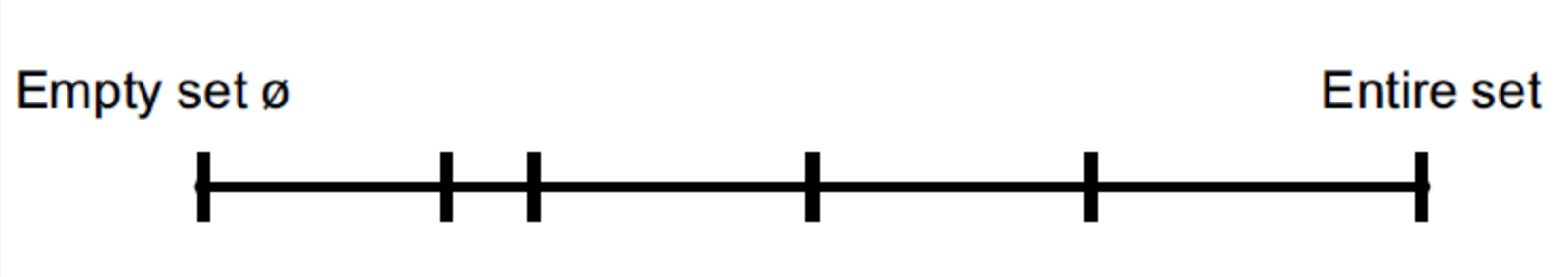
- We model a scenario where collaborative users coexist with protocol-deviating users
- We minimize:
  - § unintentional interference from benign users
  - § intentional interference from malicious users
- SimpleMAC stops:
  - § **Attack #1 of jamming the control channel,** through jamming resilience in SSS
  - § **Attack #2 of jamming based on controlchannel information, through a STS that** chooses a subset of nodes (the *recipient list)* to which coordination information is shared

Traditional Approach

- Traditional MAC protocols perform channel coordination either with everyone or no one
- With coordination, MAC-aware jammers coordinate jamming with user transmission
- Without coordination, legitimate users cannot avoid interfering with others

SimpleMAC Approach

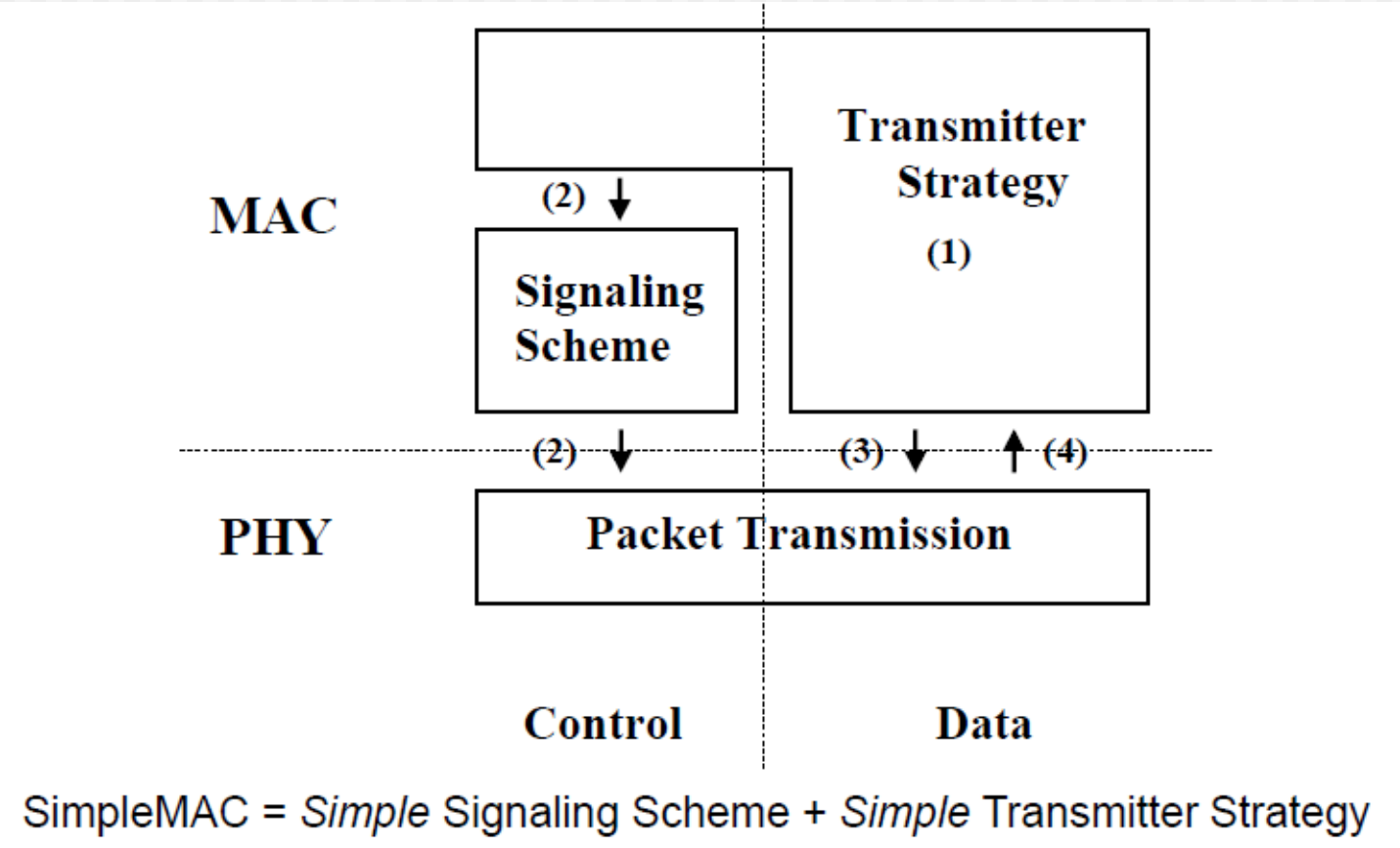
- We embrace the coexistence between collaborative users and protocol-deviating users
- (e.g., malicious users)



System Model and Assumptions

- No online central authority
- Pre-distributed keys for Sybil avoidance
- Unicast communication for data packets
- Performance metric is monotonic in SINR

MAC Framework and Terms

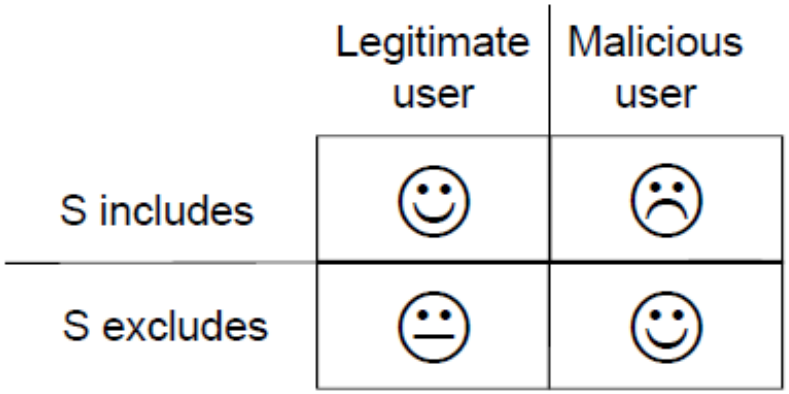


SimpleMAC = *Simple* Signaling Scheme + *Simple* Transmitter Strategy

Recipient List (S)

Recipient list is a subset of network users

Transmitter shares its channel usage information with the recipient list



Ideal recipient list includes all benign users but excludes all attackers

Simple Signaling Scheme (SSS)

- Shares control message with recipient list
- Confidentiality through symmetric cryptography
- Multicasts using repeated unicasts
- Unicast availability using direct sequence spread
- spectrum (DSSS)
- (Could also use any secure broadcast technique)

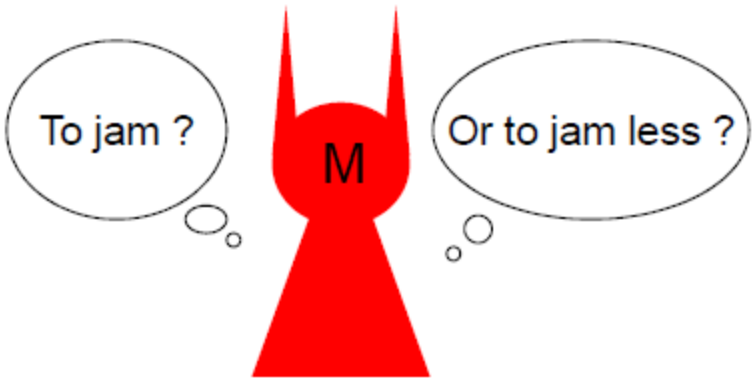
Simple Transmitter Strategy (STS)

- Decides the recipient list based on feedback:
- Each packet uses one of three possible strategies:
- Best so far, Randomly explore, Empty set
- Continued random exploration assures that ideal
- recipient list will be found
- (Standard multi-armed bandit problem)

STS Rounds

- STS operates in discrete intervals called ***rounds***
- Each round contains:
  - § A fixed number of Empty Set
  - § A decreasing number of Randomly Explore
  - § An increasing number of Best-so-Far
  - § Rate of growth in Exploration/Exploitation controlled by δ

STS Game-Theoretical Analysis



Assume:

Infinite horizon game

All bits have equal value

Then:

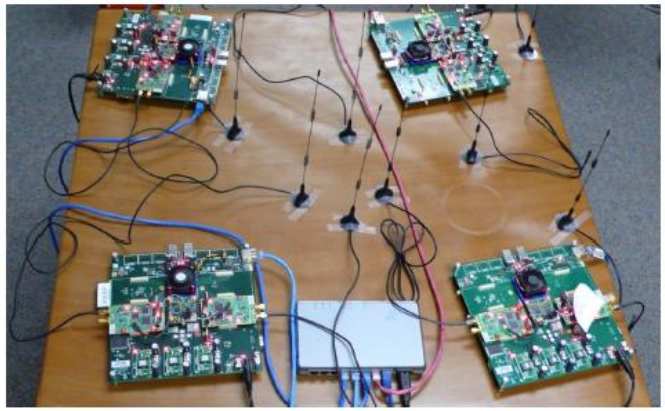
Attacker's dilemma: tradeoff between jamming and getting detected

Under STS, optimal attacker jams with full power and all available information (proof in the paper)

STS eventually finds and converges to ideal recipient list

SimpleMAC Testbed Setup

- WARP platform
- DQPSK modulation with preamble
- Antenna locations chosen for equal power
  - 1 source
  - 1 receiver
  - 4 other transmitters (incl. 1 jammer)
  - 5 channels (using OFDM)



SimpleMAC Testbed Results

- 1. Recipient list performance is consistent with our expectations
  - § More benign nodes ⇒ better performance
  - § Jammer ⇒ much worse performance
- 2. STS quickly outperforms baseline strategy
- … and gets even better with time

- In the paper: SSS helps ensure availability
  - § Spread spectrum for noise resistance

SimpleMAC Conclusion

- Studied effective MAC-aware jamming when network participants are compromised
- Devised effective countermeasure schemes that force attackers to give up the advantage of information they gain from compromised nodes
- Nearly immediate performance gain over Nash
- Equilibrium and converges to optimal performance

Challenged Networks

- Assumptions in the TCP/IP Model are Violated
  - Limited End-to-End Connectivity
    - Due to mobility, power saving, or unreliable networks
  - DTN
    - Delay-Tolerant Networks
    - Disruption-Tolerant Networks
  - Activities
    - IRTF's DTRNRG (Delay Tolerant Net. Research Group)
    - EU's Haggle project

Delay Tolerant Networks (DTNs)

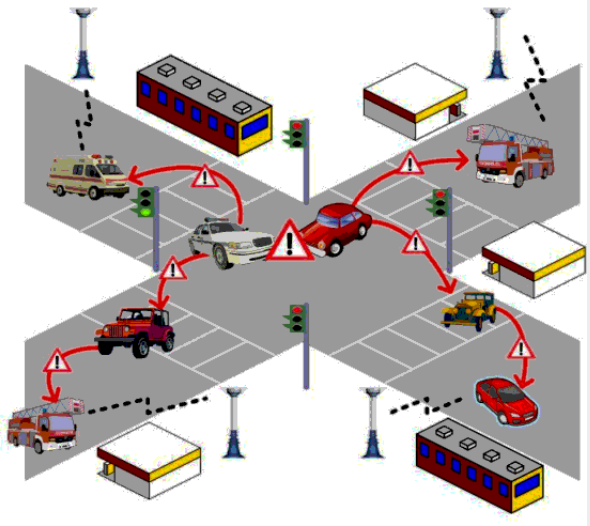
- Operates above transport layer for various network architectures and provides store and forward functionality for dissimilar and special networks.
- DTNs required to store messages in non-volatile memory when reliable delivery is required.

Why a Delay Tolerant Network

- The Internet's underlying assumptions
  - Continuous, bidirectional end-to-end path
  - Short round-trips
  - Symmetric data rates
  - Low error rates
- The characteristics of evolving and potential networks
  - Intermittent connectivity
  - Long or variable delay
  - Asymmetric data rates
  - High error rates
- New architectural concept is needed!

Examples of DTNs

- Deep space communication
- Wildlife monitoring
- Vehicular communication
- Social contact networks

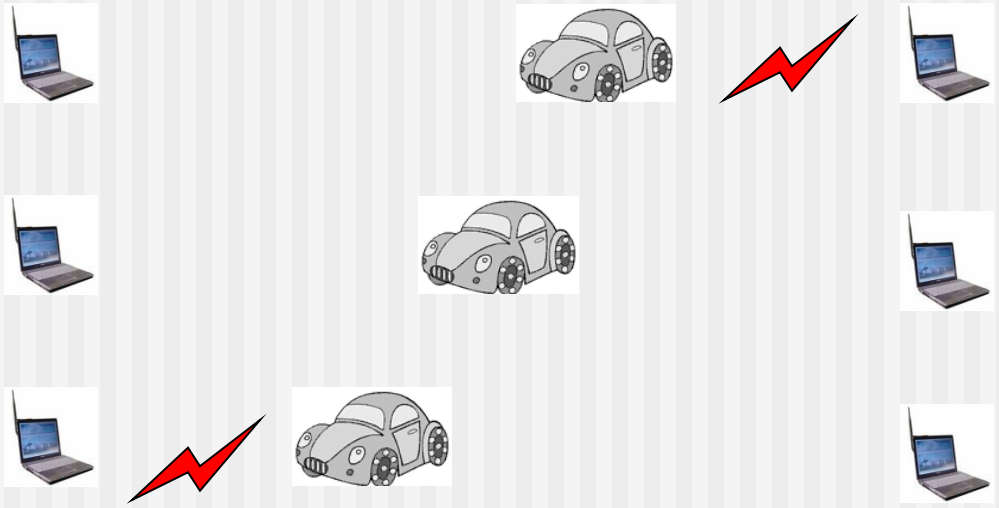


Store-Carry-Forward

- The problems of DTNs can be overcome by store-and-forward massage switching
- DTN routers need persistent storage for their queues because
  - A communication link may not be available for a long time
  - One node may send or receive data much faster or more reliably than the other node
  - A message, once transmitted, may need to be retransmitted for some reasons

Mobility as a Friend

- Movement-Assisted Routing
  - Views node movement as a desirable feature
  - Store
  - Carry
  - Forward



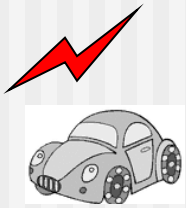
Two Paradigms

- Random Mobility
  - E.g., epidemic routing
  - Sightseeing cars (random movement)
- Controlled Mobility
  - E.g., message ferrying
  - Taxi (destination-oriented)
  - Public transportation (fixed route)

Mobility pattern affects the spread of information

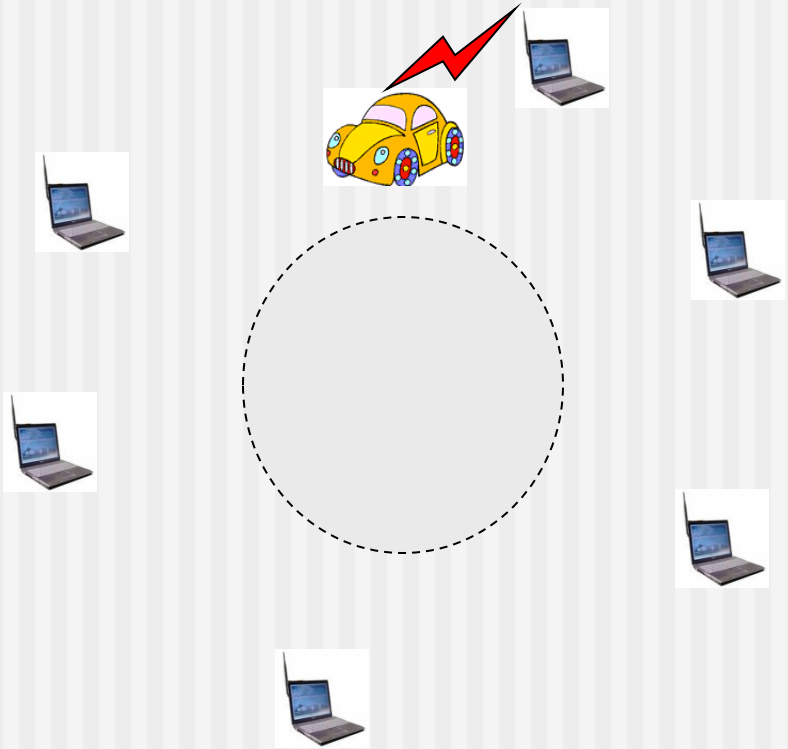
Epidemic Routing (Vahdat & Becker 00)

- Nodes store data and exchange them when they meet
- Data is replicated throughout the network through a random talk



Message Ferrying (Zhao & Ammar 03)

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



Mobility-Assisted Routing

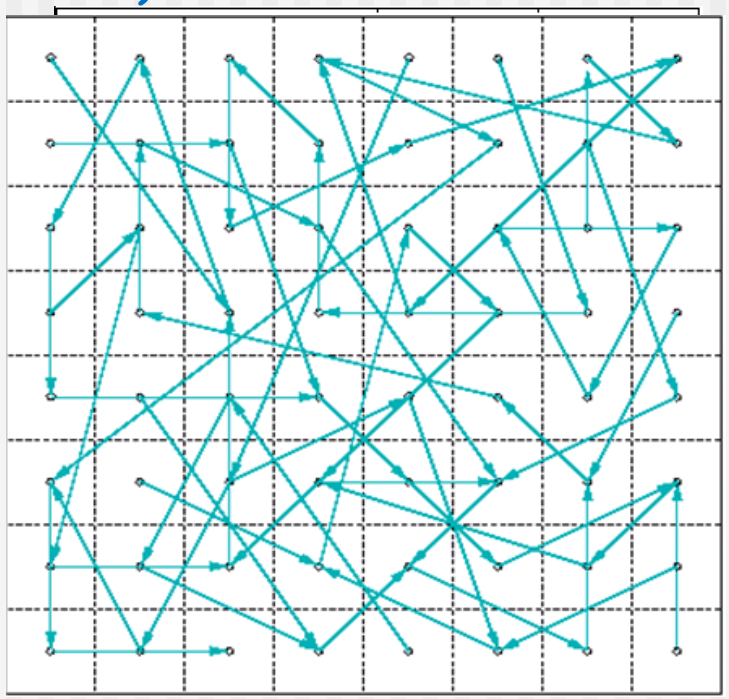
- Replication
  - Single copy vs. multiple copy
  - E.g., spray-and-wait and spray-and-focus
- Knowledge
  - Global vs. local (or zero) information
  - Deterministic vs. probabilistic information
  - E.g., MaxProp

(Predict-and-relay: Quan, Cardei, and Wu, ACM MobiHoc 2009)

Mobility-Assisted Routing (cont'd)

- Closeness (to dest.)
  - Location information (of contacts and dest.)
  - Similarity (between intermediate nodes and dest.)
  - E.g., logarithmic (and polylogarithmic) contacts
- Mobility
  - Random vs. control
  - Predictable
    - E.g., cyclic MobiSpace

(More information: Wu and Yang: IEEE MASS 2007 and IEEE TPDS 2007; Liu and Wu: ACM MobiHoc 2007 and 2008)

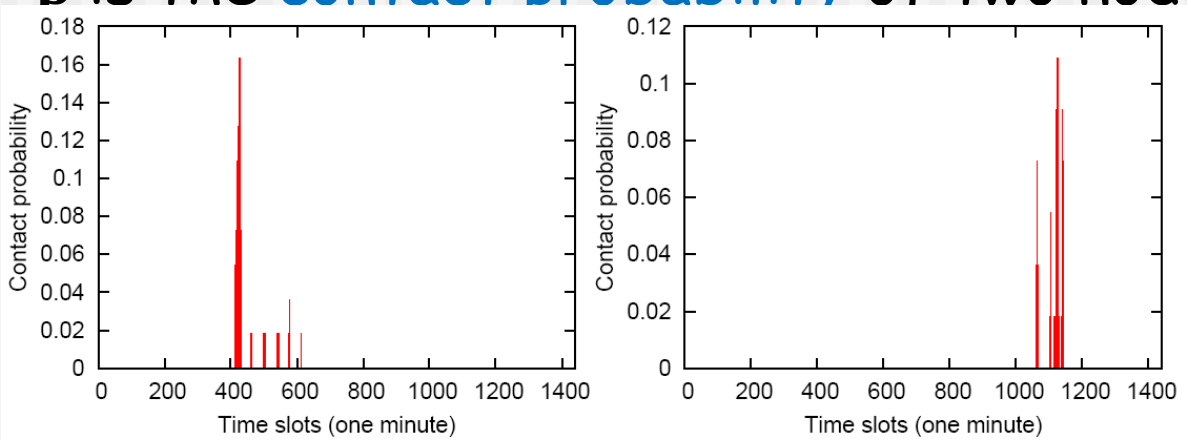


Delegation Forwarding Unicast (Erramilli 2008)

- The message holder forwards the message to an encountered node that has a higher quality than all previous nodes seen so far.
- In an N-node network, the expected cost of the algorithms
  - Flooding: $O(N)$
  - Delegation forwarding: $O(\sqrt{N})$
- Extensions
  - Probabilistic delegation forwarding
  - Delegation forwarding in multicasting

Routing in a Cyclic MobiSpace (Liu & Wu 08)

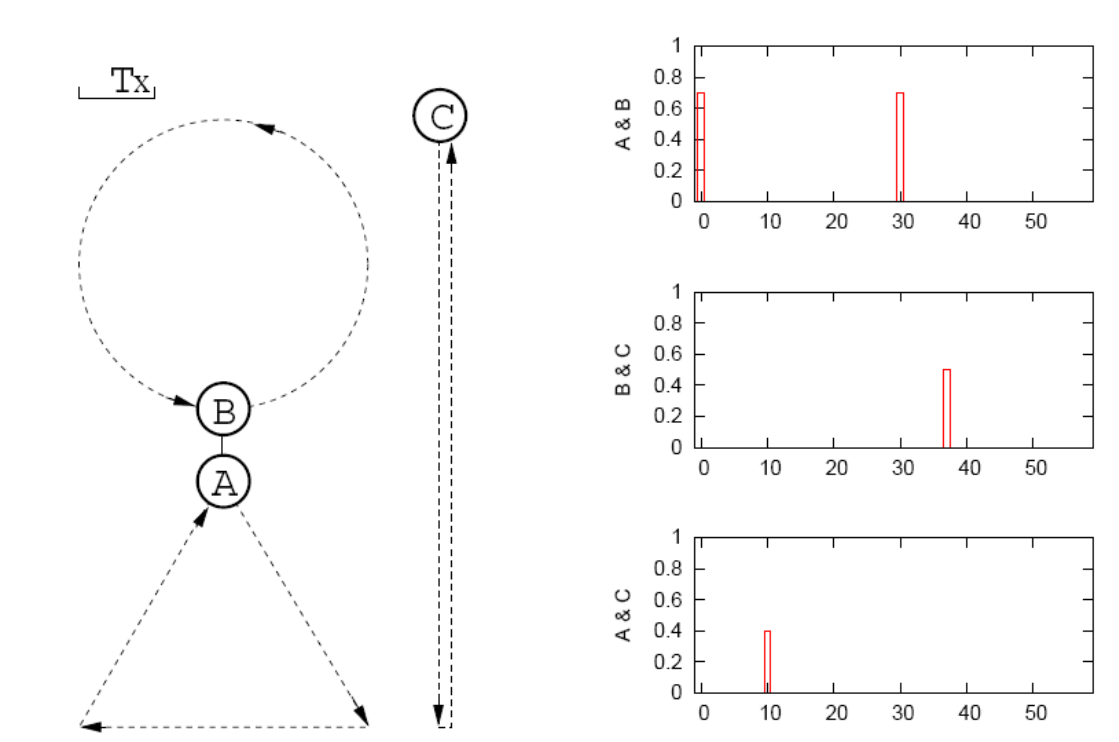
- Challenges
  - How to perform efficient routing in probabilistic time-space graphs
- Definition (t_i, p)
  - p is the contact probability of two nodes in t_i.

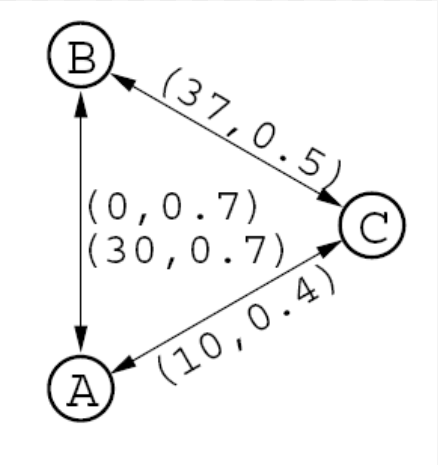


(a) Shifts 01/AM & 03/AM (b) Shifts 32/PM & 21/EVE

Probabilistic Time-Space Graph

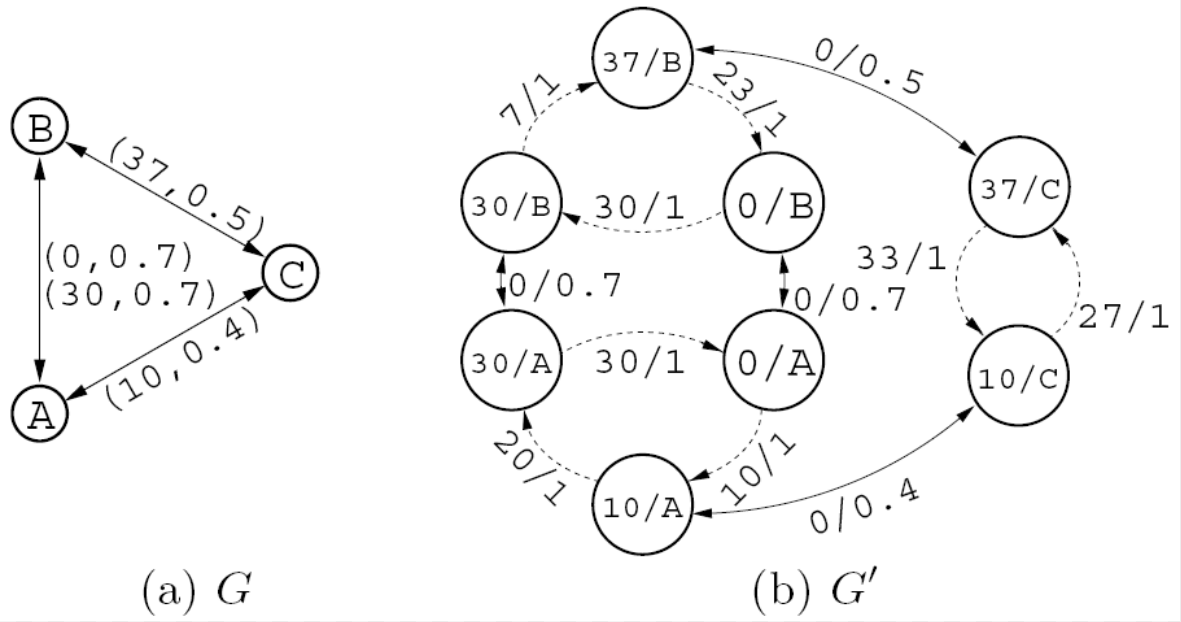
- A common motion cycle **T** (=60)





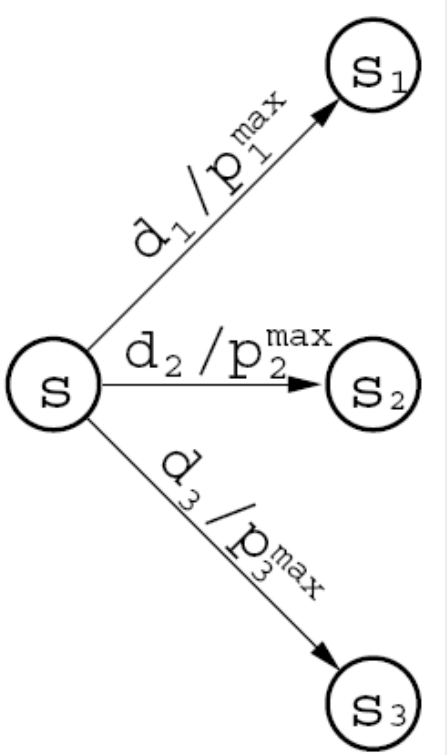
Probabilistic state-space graph

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



Iterative Process

- Iterative steps
  - Step t+1 based on step t
  - Ordering of neighbors

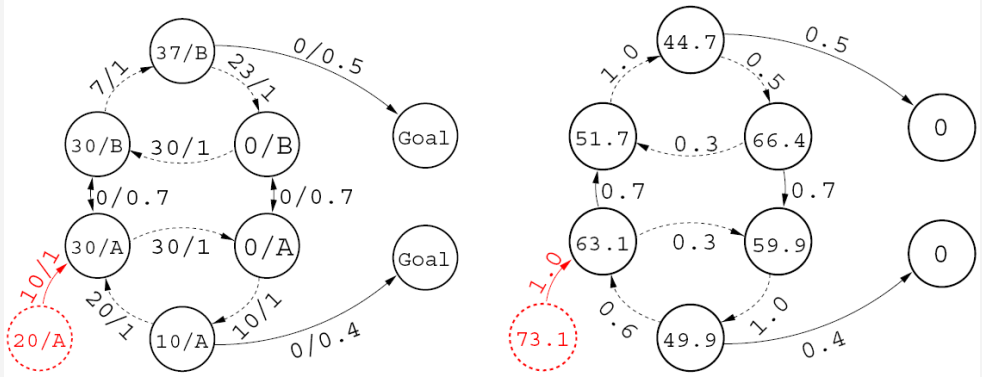


$p_i \le p_i^{max}$ and $\sum_i p_i = 1$

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

Expected Minimum Delay (EMD)

- Using EMD as the delivery probability metrics
  - Optimal single-copy forwarding: Liu and Wu MobiHoc 2008



- Optimal prob. forwarding with hop constraints
  - Single copy: Liu and Wu MobiHoc 2009
  - Multiple copy: Liu and Wu MASS 2009

Simulation

- Real traces
  - NUS student contact trace
  - UMassDieselNet trace (sub-shift based)
- Synthetic bus trace
  - Miami
  - Madrid

Social-based Routing: Feature
(Wu & Wang 12)

- Existing DTN routing:
  - *contact history*
  - *mobility pattern*
- Collecting such information is costly
- Using internal ***social feature information*** for routing without overhead.

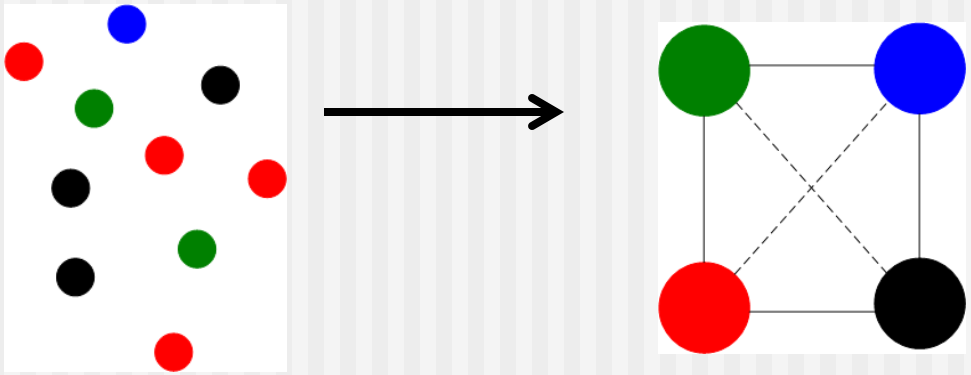
Motivation

- Increase delivery rate: *multiple copy*
  - Flooding (*O(N)*)
  - Delegation forwarding (*O(√N)*)
- It is hard to control the efficiency

- Our approach:
  - Feature-based (race, gender, language, …)
  - *logN** copies in *node-disjoint* paths
    (N * is feature space size)

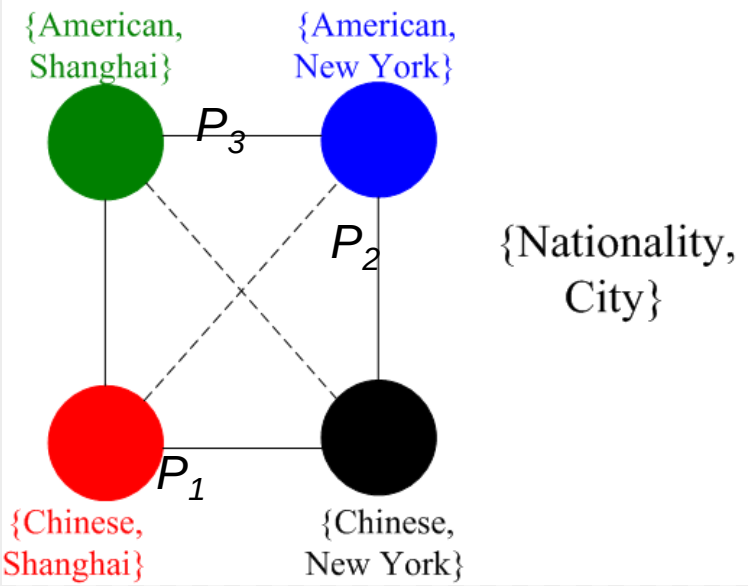
Social Feature

- Mobile & unstructured contact space (M-space) ⟶
- Static & structured feature space (F-space)



- Each individual with a social feature profile $\{F_1, F_2, \ldots\}$
- Individuals with the same features mapped to a

Example



- People come in contact with each other more frequently if they have more social features in common ($P_1 > P_3$, $P_2 > P_3$)

Feature Extraction

- *Extract m most important features*
- Extract most informative subset (**MIS**) with *m*(from *m'*) key features based on entropy

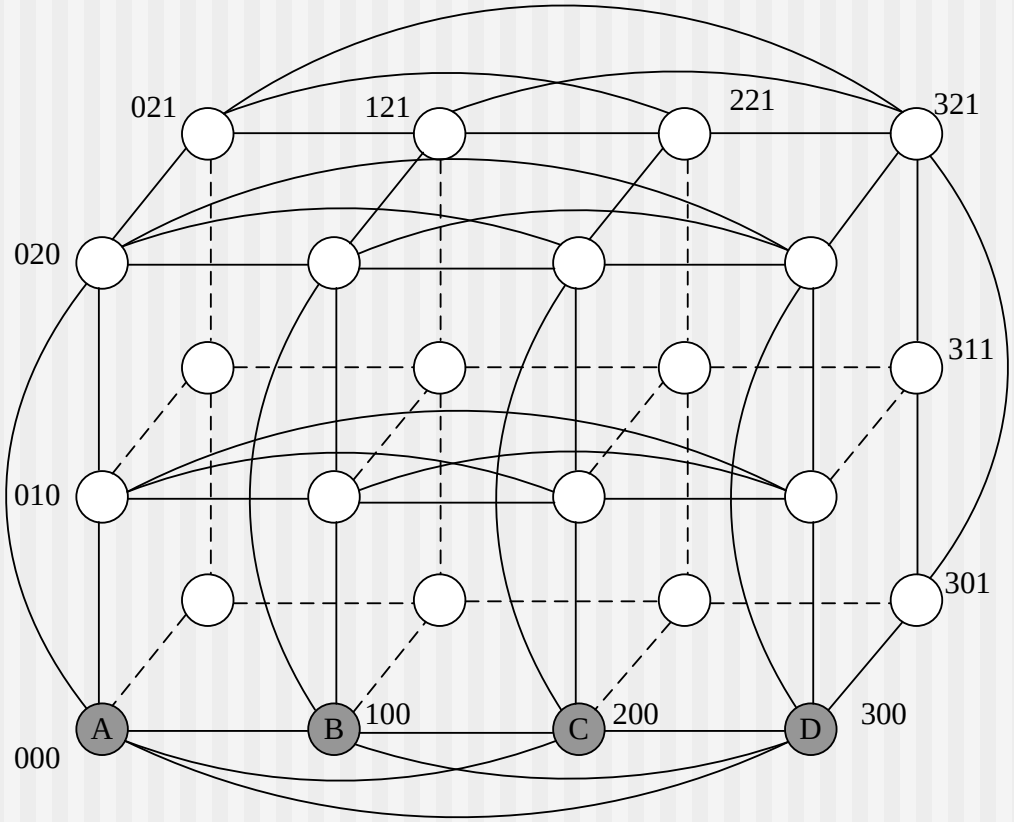
$$E(F_j) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i), \qquad (j = 1, 2, \ldots, m')$$

- $E(F_j)$: entropy of F_j
- *p:* probability mass function of F_j

The entropy of each social feature (Infocom 2006 trace)

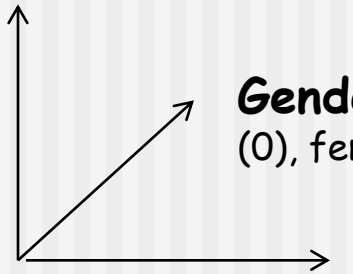
Social Feature	Entropy
Affiliation	*4.64*
City	*4.45*
Nationality	*4.11*
Language	*4.11*
Country	*3.59*
Position	*1.37*

A 3-dimensional (3-D) hypercube



- *Dimension 1: city*
- *Dimension 2: position*
- *Dimension 3: gender*

Position: professor (0), researcher (1), student (2)



Gender: male (0), female (1)

City: New York(0), London(1), Paris (2), Shanghai (3)

Example: "*311*" : a female researcher lives in Shanghai

Property of Hypercube

Efficient copy management in an m-d binary cube

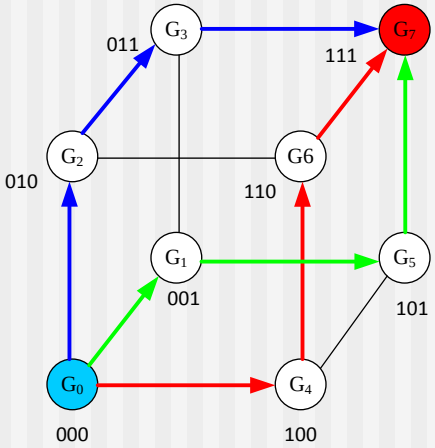
(S and D differ in k features)

- m node-disjoint paths from S to D
  - k shortest paths of length k
  - $m-k$ non-shortest paths of length $k+2$.

Hypercube Routing

- The relative address of the current group and destination group (a small string in the header)
  - calculated through XOR on S and D
  - sent, along with the packet, to the next node
- Any node in the group can forward to any node in the adjacent group
- Special treatment is needed at the destination group

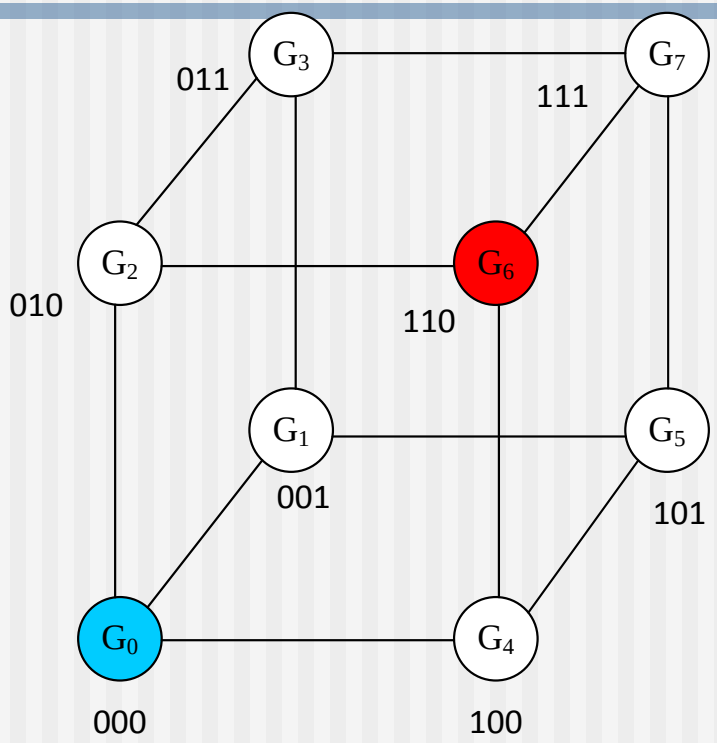
Coordinate sequence



- Path 1: <1, 2, 3> (red directed line)
- Path 2: <2, 3, 1> (blue directed line)
- Path 3: <3, 1, 2> (green directed line)

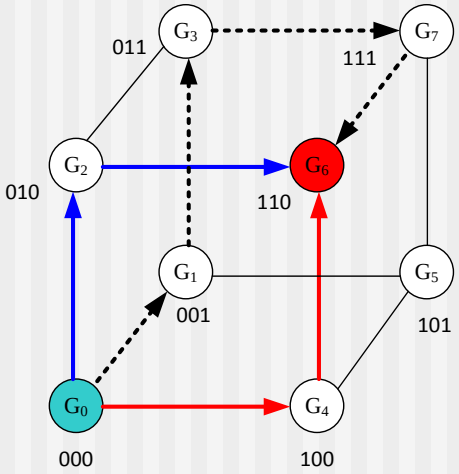
- Node-disjointness

Example: Node-Disjoint-based Routing



- $S = G_0$ and $D = G_6$

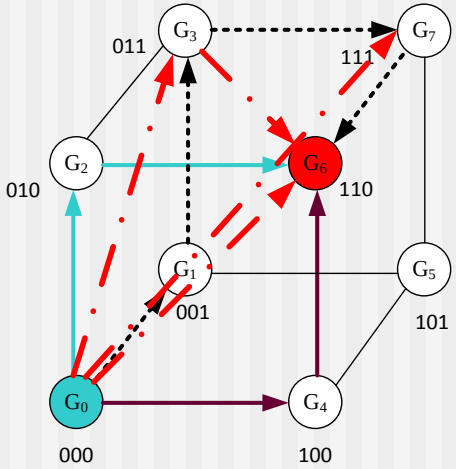
Example: Node-Disjoint-based Routing

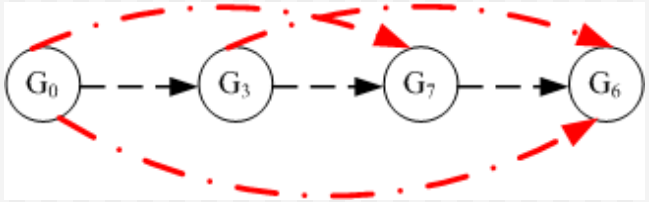


- *2* shortest node-disjoint paths:
  - <1, 2>: red directed line
  - <2, 1>: blue directed line
- *1* non-shortest node-disjoint path:
  - <3, 2, 1, 3>: dashed directed line

Shortcuts

- *Feature matching shortcut* can resolve the feature distance more than one at a time.

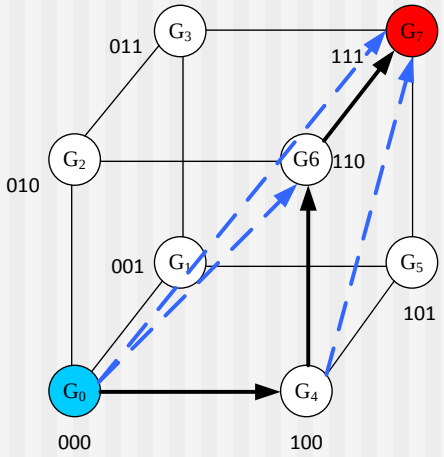




- Shortcut reduces the number of forwardings while ensuring the path disjointness property.

Composite Path

- An example of the composite path from 0000 to 1111.



Composite path: all possible paths from one node to another in *a given dimension sequence*, including the *direct path (solid directed line)* and all possible *shortcut paths (*dashed directed lines*)*.

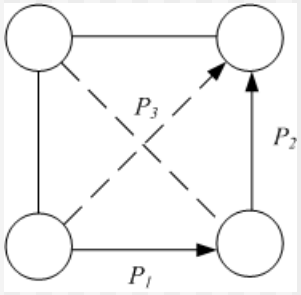
Comparison of contact frequency with different feature distance in the Infocom 2006 trace

(0000, 1000, 1100, 1110)	$p_1p_2p_3 \approx 0.007$	$P'_{1..3}$
(0000, 1100, 1110)	$p_{12}p_3 \approx 0.008$	
(0000, 1000, 1110)	$p_1p_{23} \approx 0.008$	
(0000, 1110)	$p_{123} = 0.019$	
(1000, 1100, 1110, 1111)	$p_2p_3p_4 \approx 0.007$	$P'_{2..4}$
(1000, 1110, 1111)	$p_{23}p_4 \approx 0.007$	
(1000, 1100, 1111)	$p_2p_{34} \approx 0.008$	
(1000, 1111)	$p_{234} = 0.018$	
(0000, 1000, 1100, 1110, 1111)	$p_1p_2p_3p_4 \approx 0.0013$	$P'_{1..4}$
(0000, 1000, 1100, 1111)	$p_1p_2p_{34} \approx 0.0015$	
(0000, 1000, 1110, 1111)	$p_1p_{23}p_4 \approx 0.0014$	
(0000, 1100, 1110, 1111)	$p_{12}p_3p_4 \approx 0.0014$	
(0000, 1000, 1111)	$p_1p_{234} \approx 0.0035$	
(0000, 1110, 1111)	$p_{123}p_4 \approx 0.0036$	
(0000, 1100, 1111)	$p_{12}p_{34} \approx 0.0016$	
(0000, 1111)	$p_{1234} = 0.01$	
(1000, 1100, 1110, 1111)	$p_2p_3p_4 \approx 0.007$	$P_{2..4}$
(0000, 1000, 1100, 1110, 1111)	$p_1p_2p_3p_4 \approx 0.0013$	$P_{1..4}$

- *Direct frequency: $P_{i..j}$*
- *Shortcut frequency: P_{ij}*
- *Composite frequency: $P'_{i..j}$*

Two Observations

- ***Observation 1***: the **shortcut frequency** (Diagonal line) (P_3)
  - smaller than each side ($P_3 < P_1$ or P_2)
  - Larger than the product of two sides ($P_3 > P_1 * P_2$)
- ***Observation 2***: the **composite frequency** (Diagonal line) ($P'_{1..2} = P_3 + P_1 * P_2$)
  - smaller than each side ($P'_{1..2} < P_1$ or P_2)
  - Larger than the product of two sides ($P'_{1..2} > P_1 * P_2$)



Simulation

- Synthetic trace
  - A node *A* has *m* contact frequencies, $p_1, p_2, \ldots, p_m$, with its *m* neighbors in the *m*-D F-space
  - N = 128 individuals
  - *m* is 4, 5, 7, 8, and 11 ($M = 2^m$, group number)
- Real trace
  - Infocom 2006 trace (and MIT reality mining trace)
  - N = 61 individuals
  - 6 social features, *m* is 3, 4, 5, and 6

Comparison

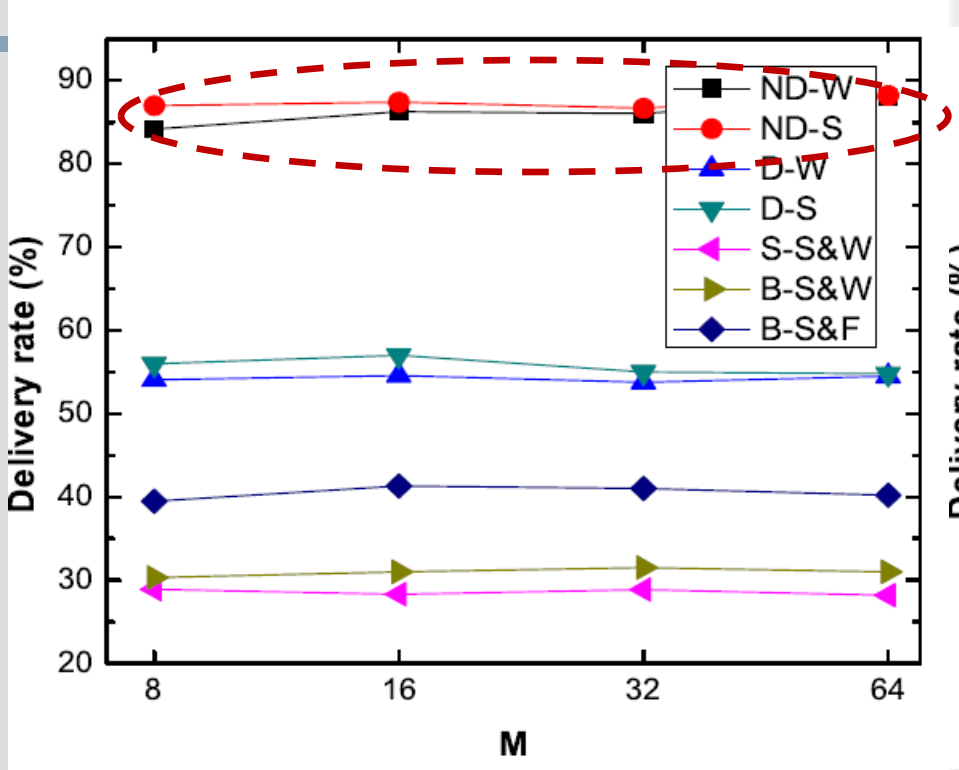
- Delegation-based with wait (**D-W**)
- Delegation-based with spray (**D-S**)
  - The copies of a packet are only forwarded to the individual with a smallest feature distance to the dest. it has met so far
- Source spray-and-wait (**S-S&W**)
  - The source forwards copies to the first m distinct nodes it encounters. If the dest. is not found, the copy carriers wait for the dest.

(Note: spray is needed at the dest. group to increase the chances to meet the actual destination!)

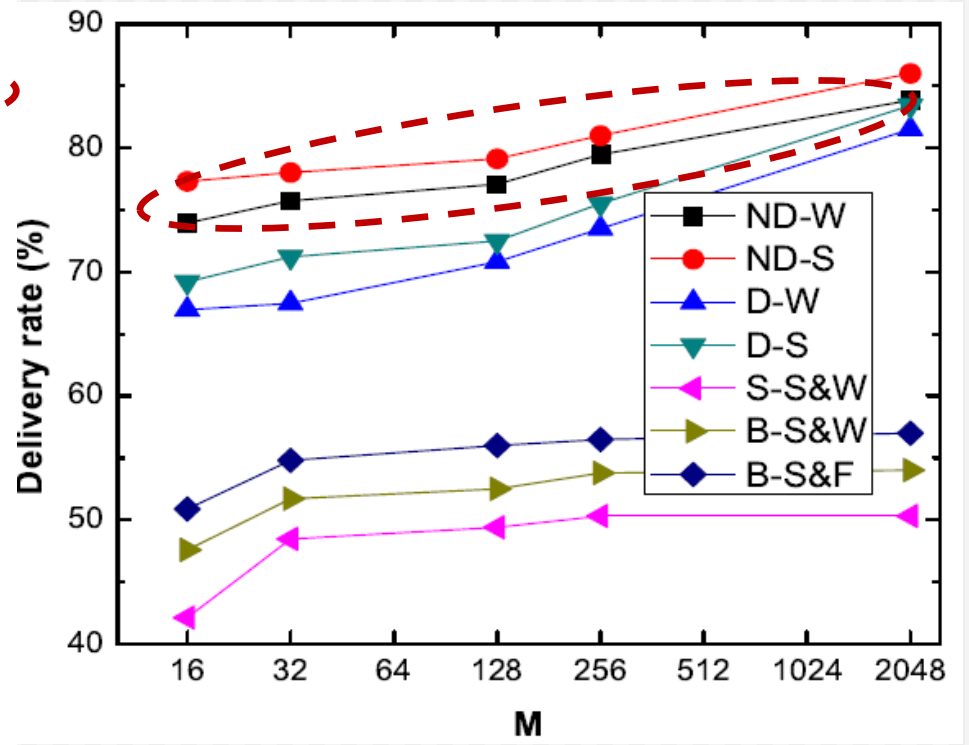
Comparison (cont'd)

- Binary spray-and-wait (B-S&W)
  - Any node with copies will forward half of the copies to the encountered node with no copy.
- Binary spray-and-focus (B-S&F)
  - The copy carriers forward the copy to the encountered node with a smaller feature distance to the dest.
- Node-disjoint-based with wait-at-dest. (ND-W)
  - Waiting for the destination after the packet enters the dest. group
- Node-disjoint-based with spray-at-dest. (ND-S)
  - Spraying N/(2M) copies into the dest. group after the packet enters

Delivery Rate

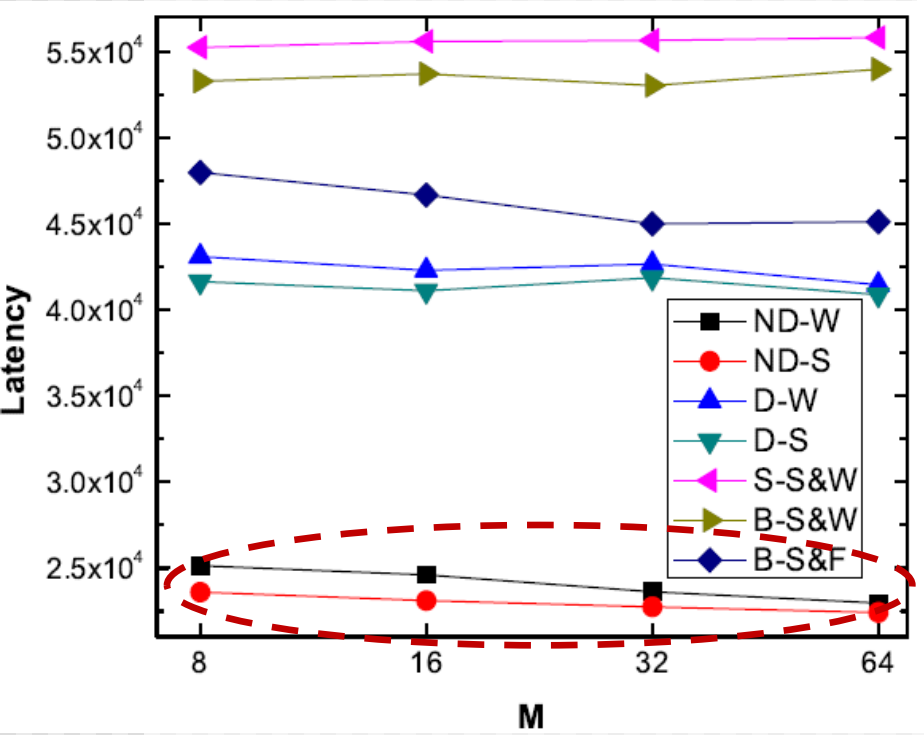


Real trace

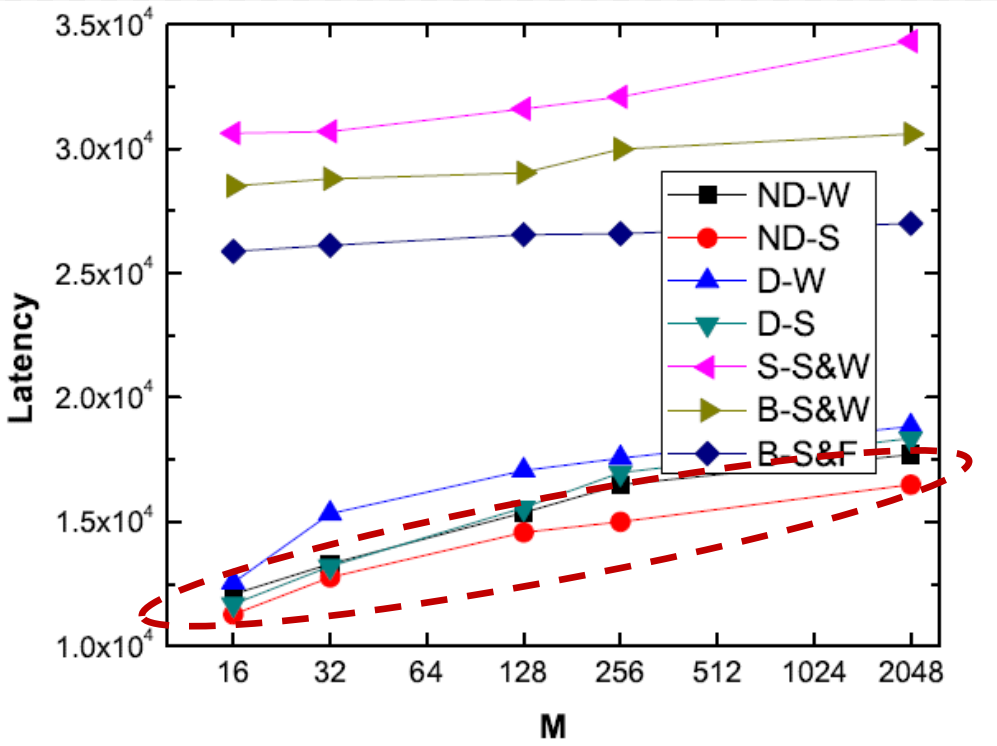


Synthetic trace

Latency

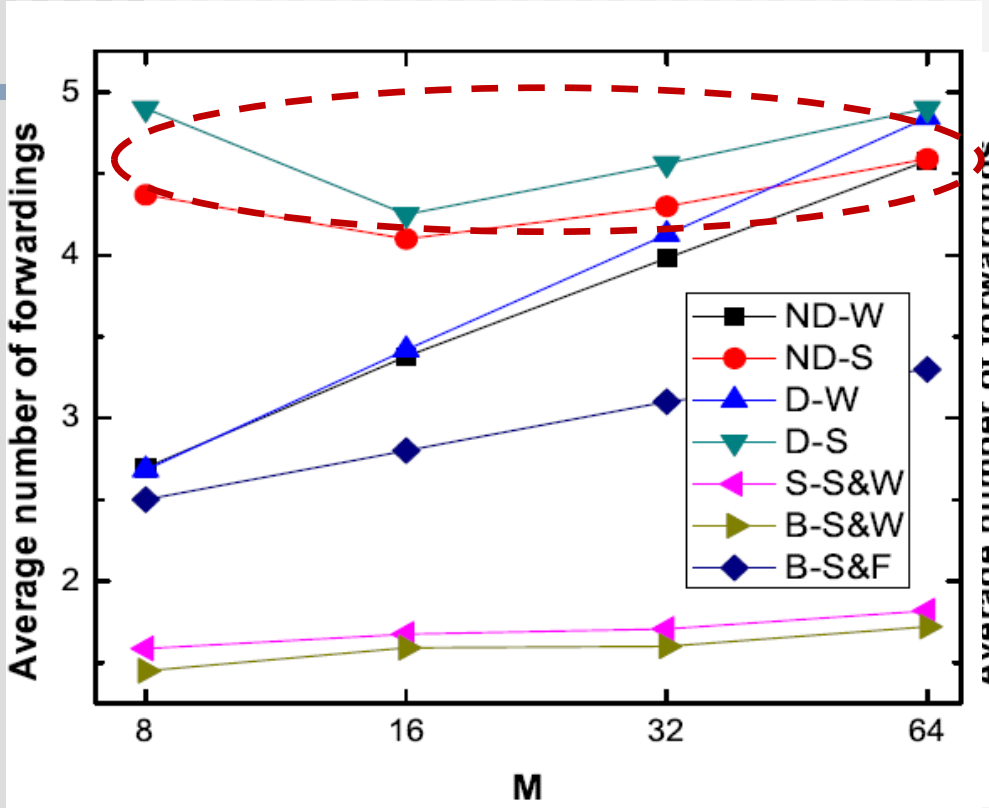


Real trace

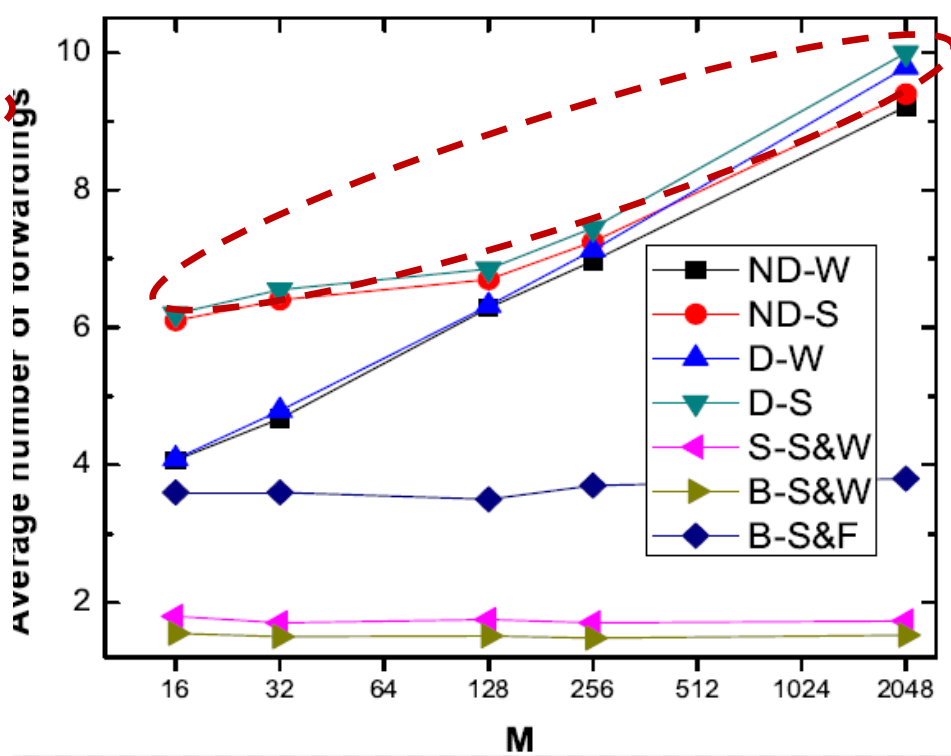


Synthetic trace

Number of Forwardings



Real trace



Synthetic trace

Recap and Extensions

- Multi-path routing in hypercubes
  - Feature-based, efficient copy management, and node-disjoint-based
- Extensions
  - General hypercubes
  - Shortest single-path routing (based on predicted contact freq.)
  - Analytical results: delivery rate and latency

Social-based: Community Home (Xiao, Wu, & Huang 13)

- Social characteristics

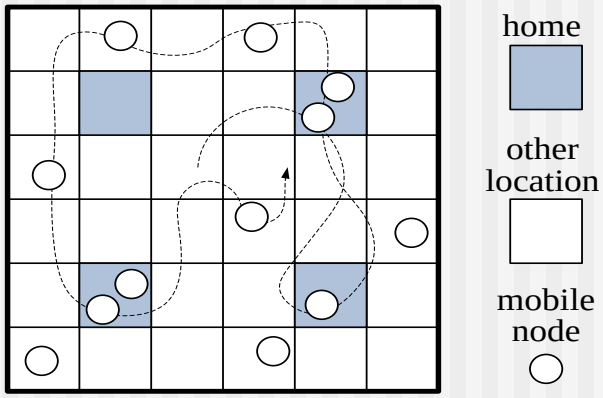
  Nodes visit some locations (community homes) frequently, while the other locations are visited less frequently.

- Real or virtual "throwbox"

  Each community home is equipped with a real or virtual "throwbox" so that it can store and forward messages

Community Home-Based Routing

- Network Model
  - V={1,2,…,n}, H={n+1, n+2,…, n+h}



- Problem
  - Given a fixed number of message copies C
  - Minimize the expected delivery delay

Homing Spread (HS)

Three phases

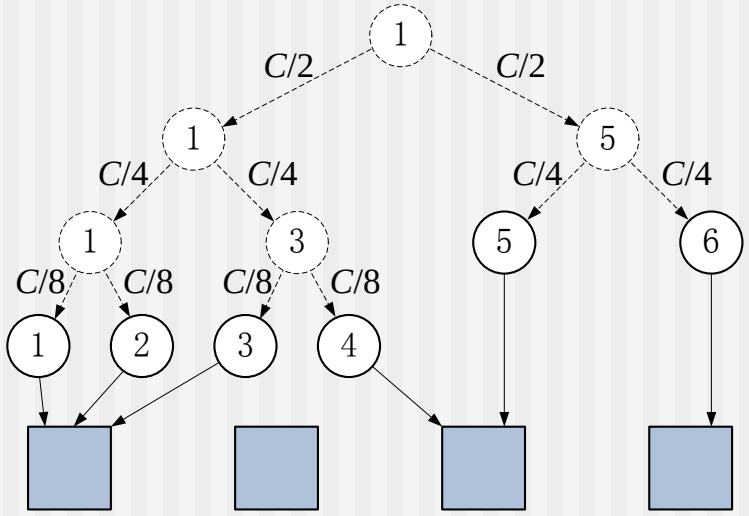
- Homing phase
  - The source sends message copies to homes quickly
- Spreading phase
  - Homes with multiple message copies spread them to other homes and mobile nodes
- Fetching phase
  - The destination fetches the message when it meets any message holder

Challenges

- Given a fixed number of message copies C
  - What is the optimal way for a message holder to spread copies during homing and roaming?
  - Once a home receives some message copies, how should it further spread these copies?
  - What is a general way for a mobile destination to obtain a copy?

Homing Phase

- Binary Homing Scheme
  - Each message holder sends all of its copies to the first (visited) home.
  - If the message holder meets another node before it visits a home, it binary splits the copies between them.



Homing Phase

- Assume:
  - Inter-meeting time between any two nodes follows the exponential distribution (λ)
  - Inter-meeting time between a node and a home follows the exponential distribution (Λ)
- Lemma 1:
  - The binary homing scheme can spread the C message copies to the maximum umber of nodes before they reach the homes.

Homing Phase

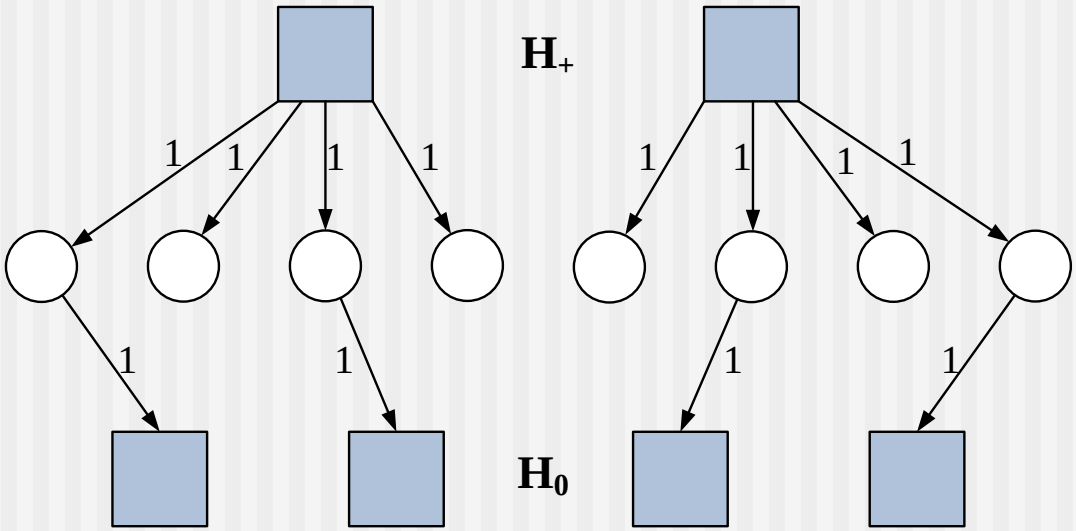
- Analysis
  - The expected delay of each message copy is always $1/h\Lambda$ no matter which splitting scheme is adopted
  - the maximum umber of nodes received the message copies
  - the maximum umber of homes received the message copies
- The binary homing scheme is the optimal homing scheme

Spreading Phase

- 1-Spreading Scheme
  - Each home with more than one message copies spreads a copy to each visiting node until only one copy remains
  - If a node with one copy later visits another home, the node sends the copy to that home
- Result
  - Each home has at most one copy.
  - If $C > h$, there are $C-h$ nodes outside the homes that have a copy.

Spreading Phase

- 1-Spreading Scheme

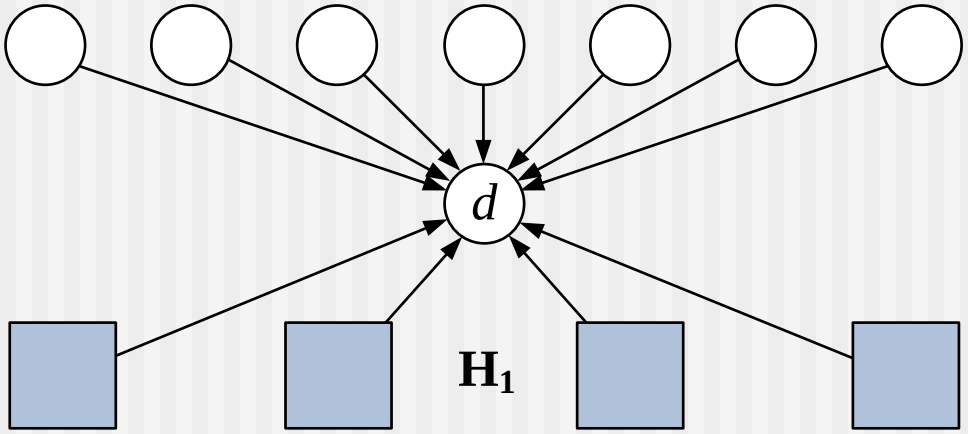


Spreading Phase

- Lemma 2:
  - The 1-spreading scheme can spread message copies from a home to the maximum number of nodes with the fastest speed.

Fetching Phase

- Fetching Scheme
  - The destination fetches the message once it meets a message holder



Distributed HS Algorithm

Algorithm 1 The Homing Spread (HS) algorithm

1: **for** each mobile node i **do**
2: **if** node i encounters another node j **then**
3: **if** node j is the destination **then**
4: node i sends the message to j;
5: **if** nodes i and j have r_i and r_j message copies **then**
6: node i holds $\lceil r_i/2 \rceil + \lfloor r_j/2 \rfloor$ copies through exchange with node j;
7: **if** node i visits a home h **then**
8: node i sends all its copies to h;
9: **if** $h \in H_+$ or i is the destination **then**
10: h sends a copy to node i.

Continuous Markov Chain

- Expected delivery delay
- Number of copies needed for a given delay bound

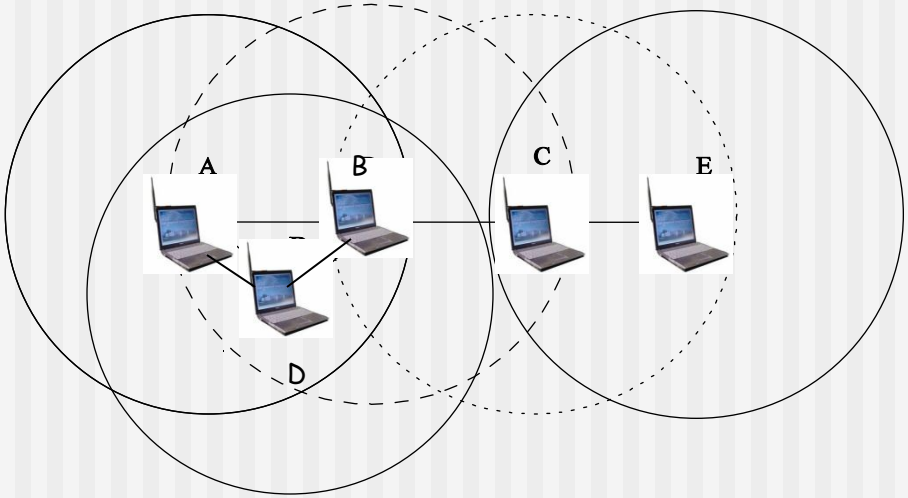
Recap and Extensions

- Homing-spread
  - Optimal multi-copy routing in three phases: homing, spreading, and fetching
- Extensions
  - Heterogeneous homes: nodes having different homes
  - Heterogeneous visiting: nodes having different visit rates
  - Utility-based routing: messages with different rewards

Other Challenges

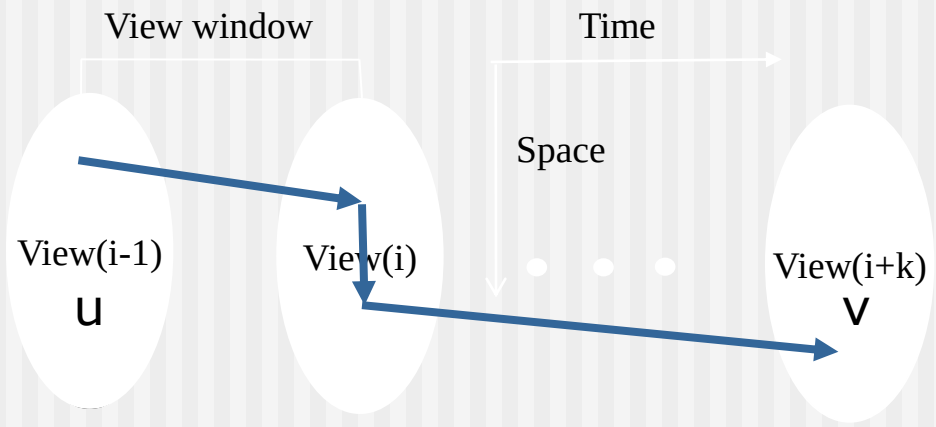
- Mobility
- Connectivity
- Complexity
- Bandwidth
- Latency
- Robustness
- Storage
- Security

- Intermittent connectivity
  - Node mobility
  - Unstable wireless links
  - Scheduled on/off sensor nodes



Connectivity

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



Complexity

Managing complexity of time-space graphs

- Lossless translation method
  - Time-space to state-space (state explosion issue)
- Lossy comprehension method
  - Removing time using averaging in hierarchical routing
  - E.g. contact information compression

(Liu & Wu: Scalable Routing in Delay Tolerant Networks, ACM MobiHoc 2007)

Evolving Graph and Its Extensions

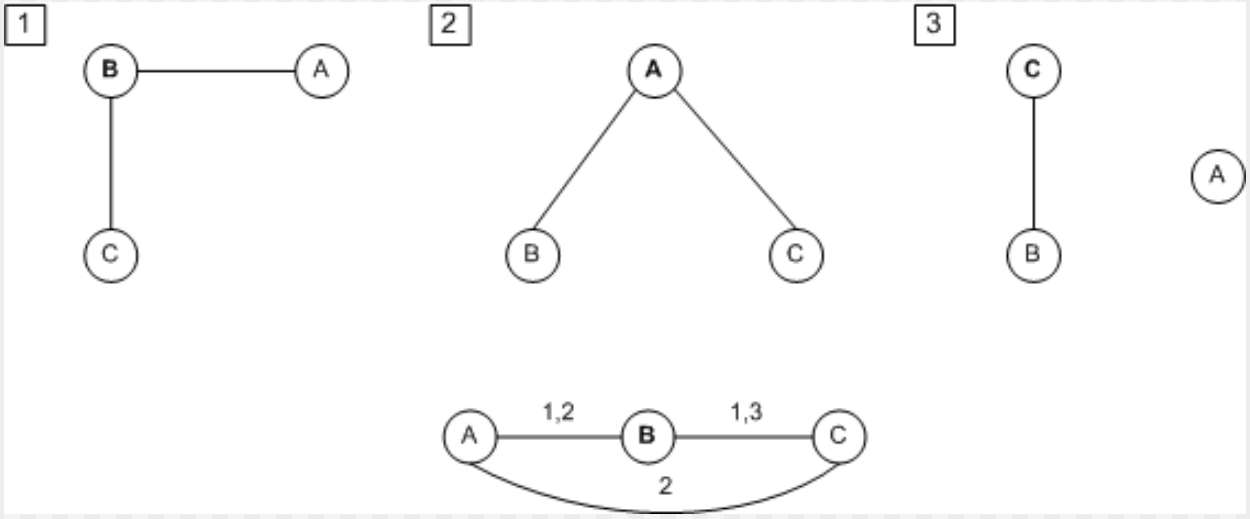
- Time sequnence: $t_1, t_2, \ldots, t_L$
- $Gi = (V_i, E_i)$: subgraph during $[t_i, t_{i-1})$
- Evolving graph

  (V, E), where $(u,v) = \{i \mid (u, v) \epsilon E_i\}$.

- Weighted evolving graph

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

  where w_i can be bandwidth, reliability, or latency



Several Optimization Problems

- Optimization
  - Earliest-completion
  - Fastest
  - Minimum-hop
  - Maximum-bandwidth
  - Maximum-reliability

Dijkstra's Shortest Path Algorithm

- Dijkstra's algorithm (Dijk) on (s, d)
  - Initially s is black and all others are white.
  - White nodes are colored gray if it has a black neighbor.
  - Select "best" gray node (w.r.t to s) and color it black (i.e., relax: adjust its "best" metric).
  - Repeat the above steps until d becomes black.

Challenges

- Optimal greedy – optimal prefix principle

- Proposed solutions
  - Slicing
    - Partition G into $G_1, G_2, \ldots, G_i$
    - Select the best among i solutions for G_i
  - Virtualization
    - Enlarge G to G' through virtualization
    - Solve G' which includes a solution for G

Journey

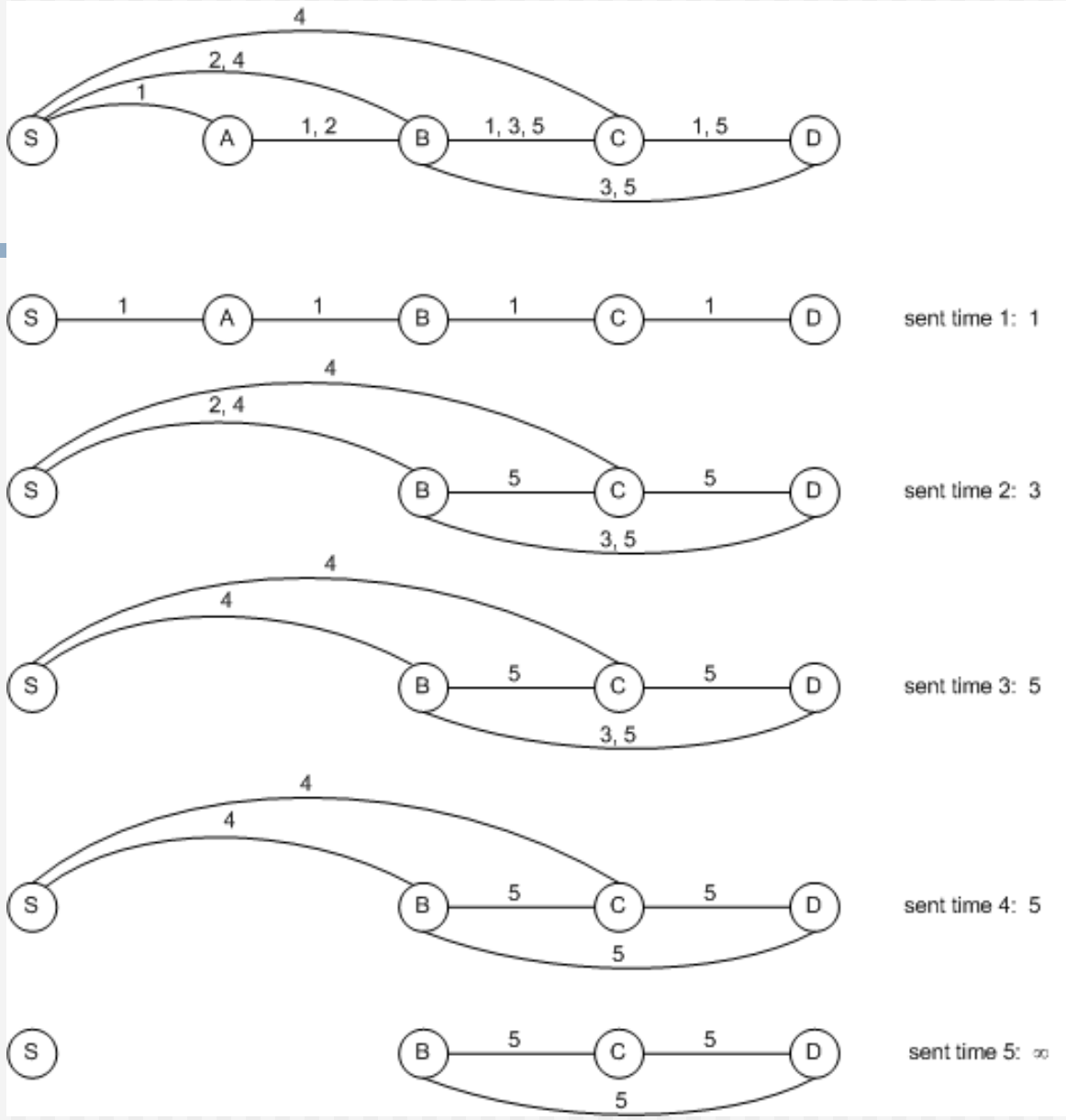
- Journey
  - Selection of non-decreasing link labels along a path.
  - E.g. (2, 4), (2, 5), (4,5)
- Earliest journey
  - A journal with the smallest last label.

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

- Start time is i at s
- Apply Dijk(G(i)) for earliest completion time
- Suppose completion time for d is f_i, then time span is $s_i = f_i - i$
- Fastest: min{s_i}
- Complexity: L times of Dijk



Minimum Hops

G(l ≤ i): a subgraph with labels ≤ i

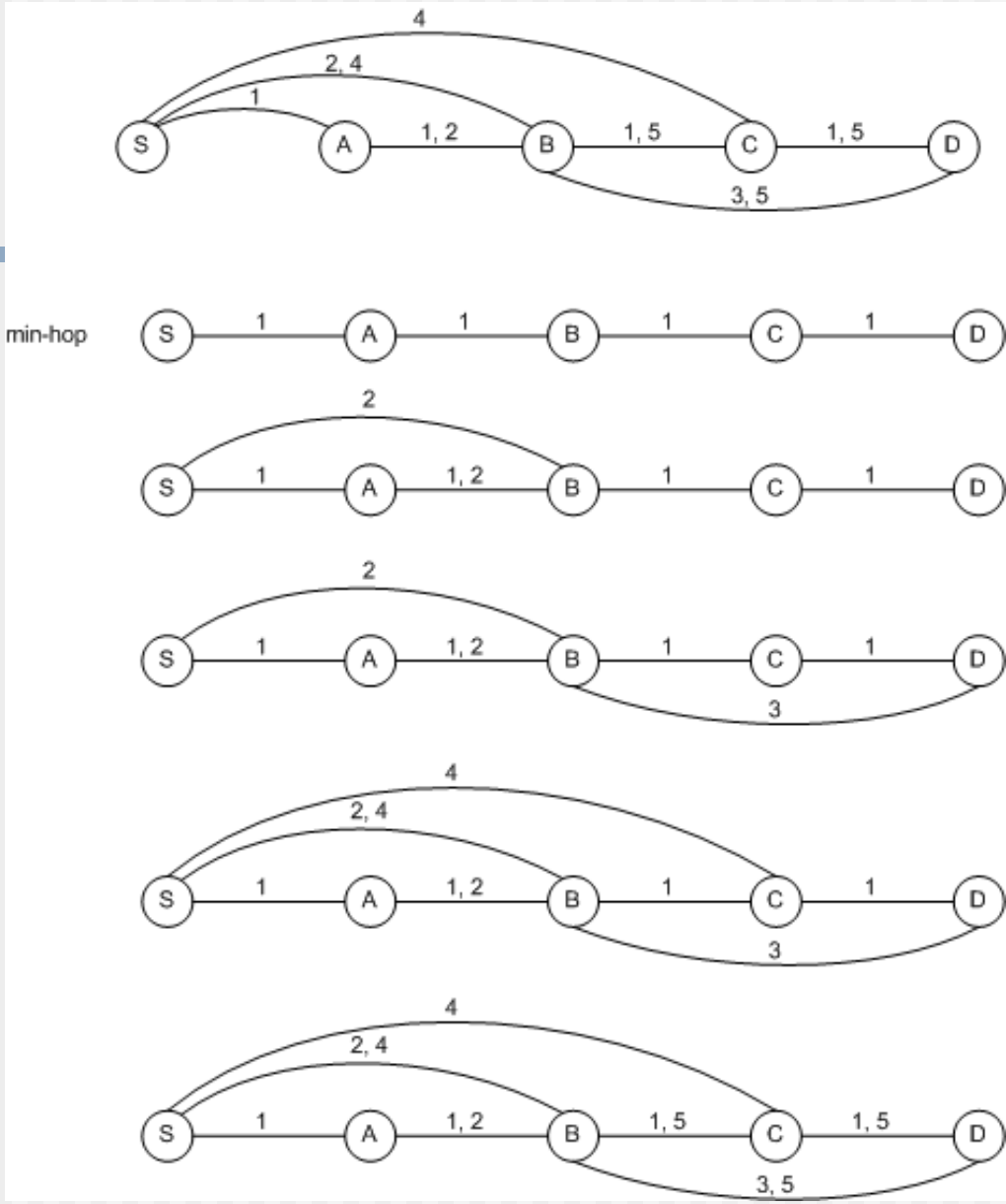
- Dijk(G(l ≤ 1))
- Dijk(G(l ≤ 2)) on above results by relaxing only links with label 2

…

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

Result is minimum hop count to d after Dijk(G(l ≤ L))

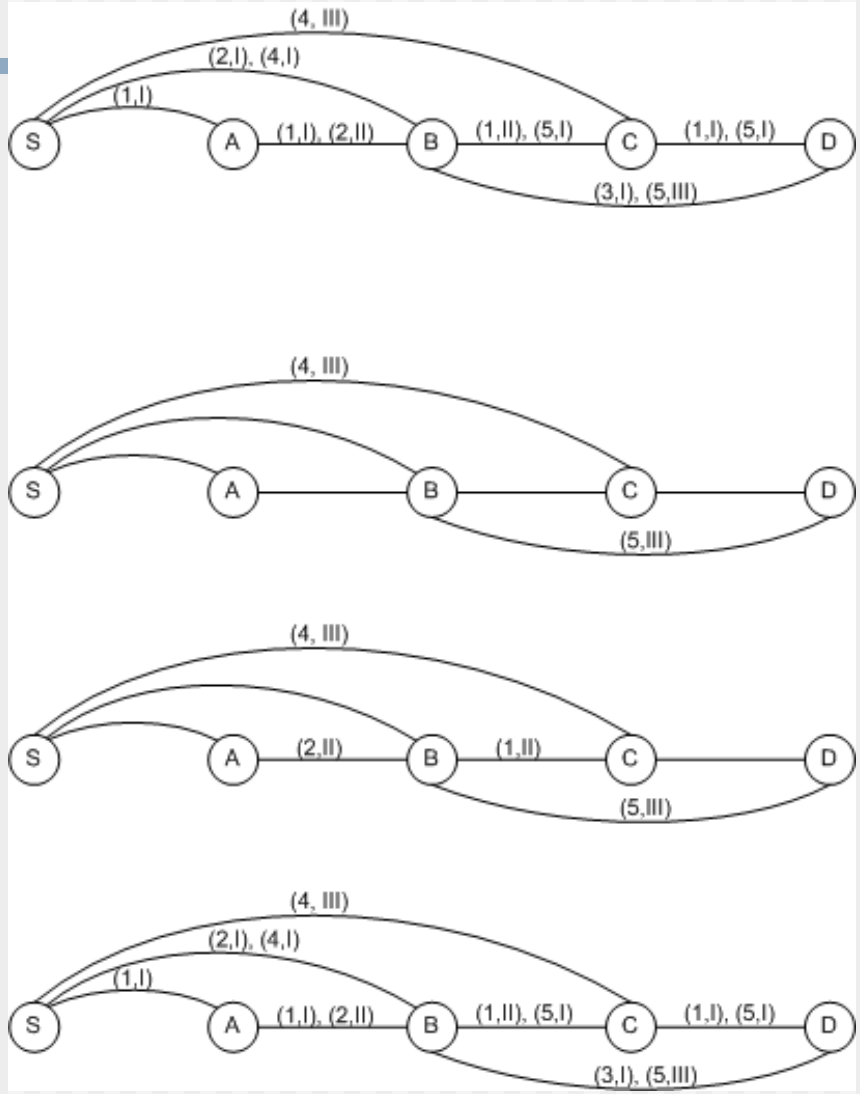
Complexity: L times of Dijk*



Maximum Bandwidth

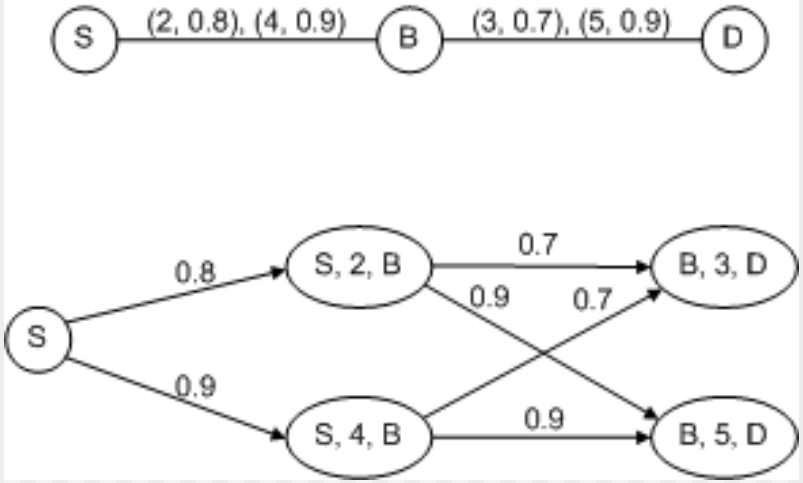
Round i (starting i = largest)

- Dijk(*G*(B≥i)) /* subgraph of labels with bandwidth ≥ i, but exact bandwidth is removed */
- Stop if d is reachable and bandwidth is i
- Otherwise, repeat the above for i = i-1
- Complexity: log L times of Dijk



Maximum Reliability

Virtual Graph (G')



- For a node v in (u, v) with labels $\{l_1, l_2, \ldots, l_{L'}\}$

- L' virtual nodes are used (u, l_i, v) for each v.

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

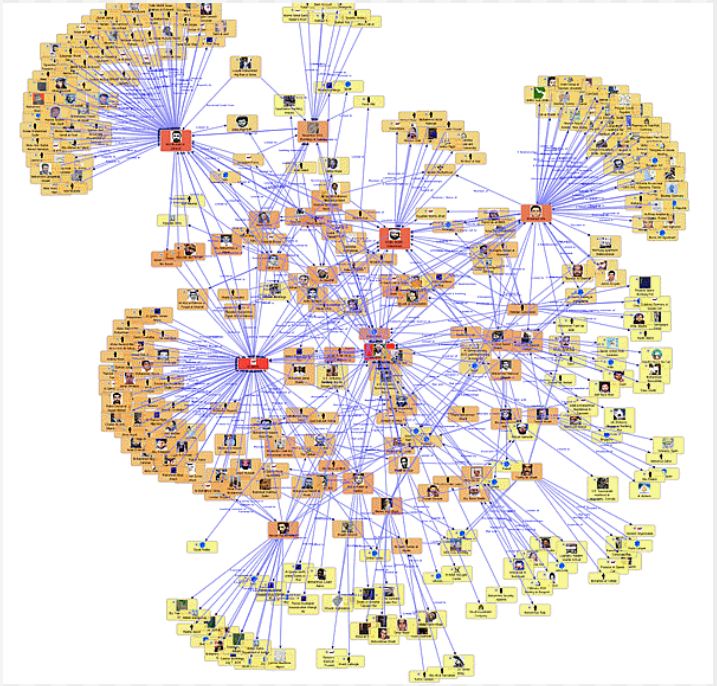
Opportunities

- Increasing system performance
  - Routing capability
  - Network capacity
  - Security
  - Sensor coverage
  - Information dissemination (mobile pub/sub)
  - Reducing uncertainty in reputation systems

    (Li and Wu, IEEE INFOCOM 2007)

Social Networks

- A social network is a theoretical construct useful in the social sciences to study relationships between
  - individuals
  - groups
  - organizations
  - or even entire societies.



Social Network Structure --- *Centrality*

- With graph theory and network analysis, there are various measures of the centrality of a vertex within a graph that determine the relative importance of a vertex within the graph.
  - degree centrality,
  - betweenness centrality
  - closeness centrality

Degree Centrality

- Degree centrality is defined as the number of links incident upon a node.
- A node with high degree centrality maintains contacts with numerous other network nodes.

Betweenness Centrality

- Betweenness is a centrality measure of a vertex within a graph.
- Vertices that occur on many shortest paths between other vertices have higher betweenness than those that do not.

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}},$$

where σ_{st} is the number of shortest paths from *s* to *t*, and $\sigma_{st}(v)$ is the number of shortest paths from *s* to *t* that pass through a vertex *v*.

Closeness Centrality

- Closeness centrality measures the reciprocal of the mean geodesic distance $d(p_i, p_k)$, which is the shortest path between a node p_i and all other reachable nodes:

$$C_C(p_i) = \frac{N-1}{\sum_{k=1}^{N} d(p_i, p_k)},$$

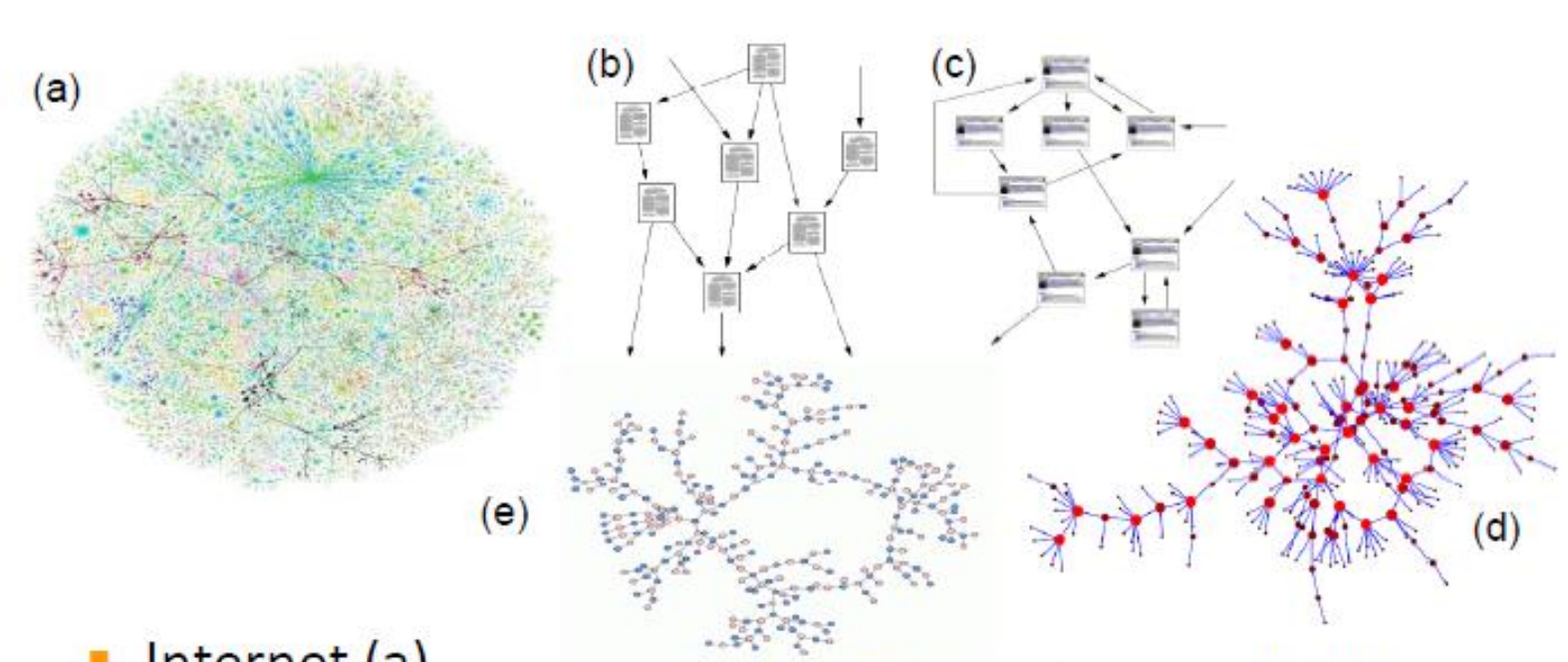
  where N is the number of nodes in the network and $i \neq k$.
- Closeness centrality can be regarded as a measure of how long it will take information to spread from a given node to other nodes in the network.

Information Propagation

- Information Propagation in Social Networks
  - Influence Maximization (KDD 03)
  - Efficient Influence Maximization in Social Networks (KDD09)
  - Cascading Behavior in Large Blog Graphs (SDM 07)

*Some slides are borrowed from
http://www.cs.cmu.edu/~jure/talks/www08tutorial/
And corresponding authors

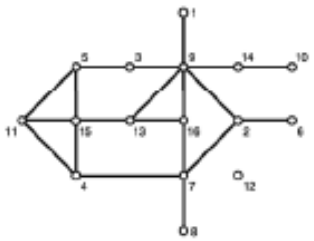
Examples of Networks



- Internet (a)
- Citation network (b)
- World Wide Web (c)
- Sexual network (d)
- Dating network(e)

Examples of Networks

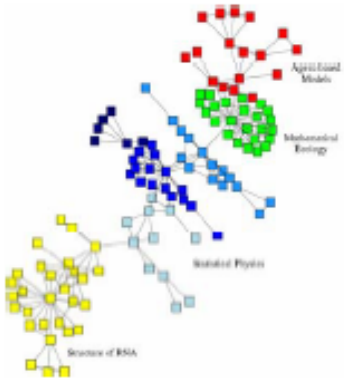
- Information networks:
  - World Wide Web: hyperlinks
  - Citation networks
  - Blog networks
- Social networks: people + interactions
  - Organizational networks
  - Communication networks
  - Collaboration networks
  - Sexual networks
  - Collaboration networks
- Technological networks:
  - Power grid
  - Airline, road, river networks
  - Telephone networks
  - Internet
  - Autonomous systems



Florence families

Karate club network

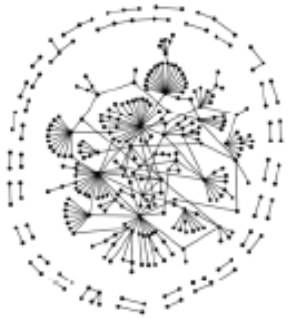
Friendship network



Collaboration network

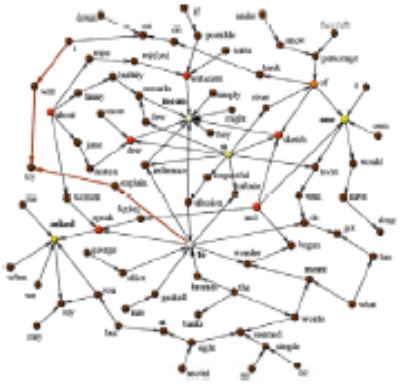
Examples of Networks

- Biological networks
  - metabolic networks
  - food web
  - neural networks
  - gene regulatory networks
- Language networks
  - Semantic networks
- Software networks
- ...

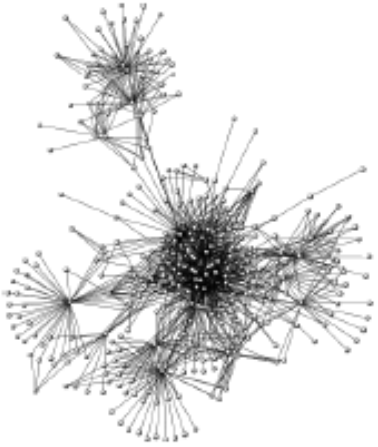


Yeast protein interactions

Semantic network

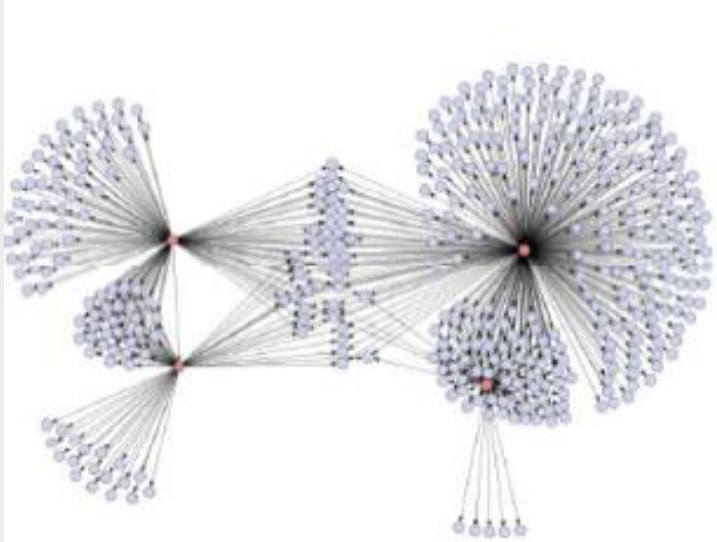


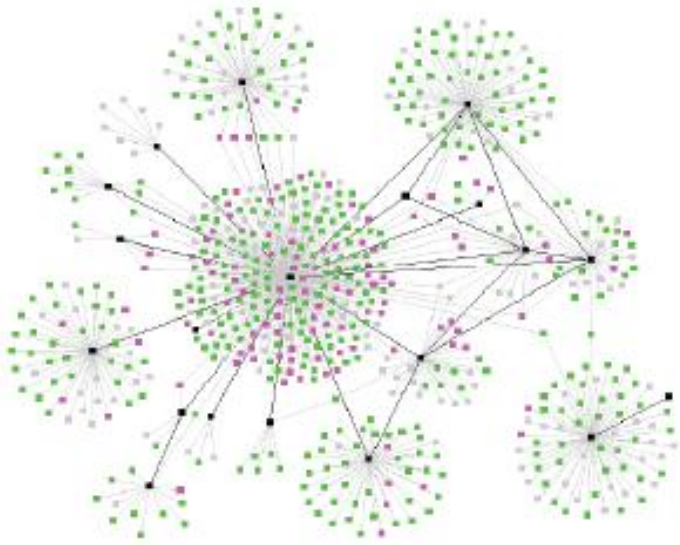
Language network



Software network

Diffusion in Networks





- One of the networks is a spread of a disease, the other one is product recommendations
- Which is which? ☺

Diffusion Curves

- Basis for models:
- – Probability Probability of of adopting adopting new new behavior behavior depends depends on the number of friends who have adopted
- [Bass ’69, Granovetter ‘78, Shelling ’78]

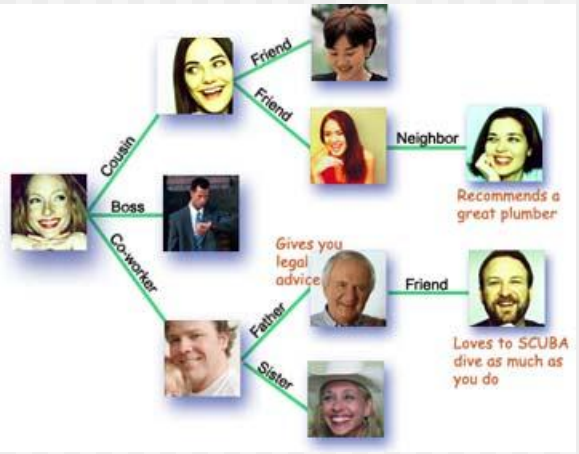
- Diminishing returns? Critical mass?

Maximizing the Spread of Influence through a Social Network

Authors: David Kempe, Jon Kleinberg, Éva Tardos

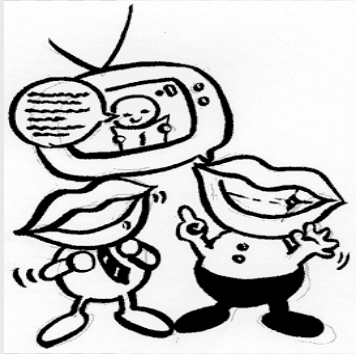
KDD 2003

Social Network and Spread of Influence



- Social network plays a fundamental role as a medium for the spread of INFLUENCE among its members
  - Opinions, ideas, information, innovation…

- Direct Marketing takes the "word-of-mouth" effects to significantly increase profits (Gmail, Tupperware popularization, Microsoft Origami …)



Problem Setting

- Given
  - a limited budget B for initial advertising (e.g. give away free samples of product)
  - estimates for influence between individuals
- Goal
  - trigger a large cascade of influence (e.g. further adoptions of a product)
- Question
  - Which set of individuals should B target at?
- Application besides product marketing
  - spread an innovation
  - detect stories in blogs

What we need

- Form models of influence in social networks.
- Obtain data about particular network (to estimate inter-personal influence).
- Devise algorithm to maximize spread of influence.

Models of Influence

- First mathematical models
  - [Schelling '70/'78, Granovetter '78]
- Large body of subsequent work:
  - [Rogers '95, Valente '95, Wasserman/Faust '94]
- Two basic classes of diffusion models: threshold and cascade
- General operational view:
  - A social network is represented as a directed graph, with each person (customer) as a node
  - Nodes start either active or inactive
  - An active node may trigger activation of neighboring nodes
  - Monotonicity assumption: active nodes never deactivate

Outline

- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Linear Threshold Model

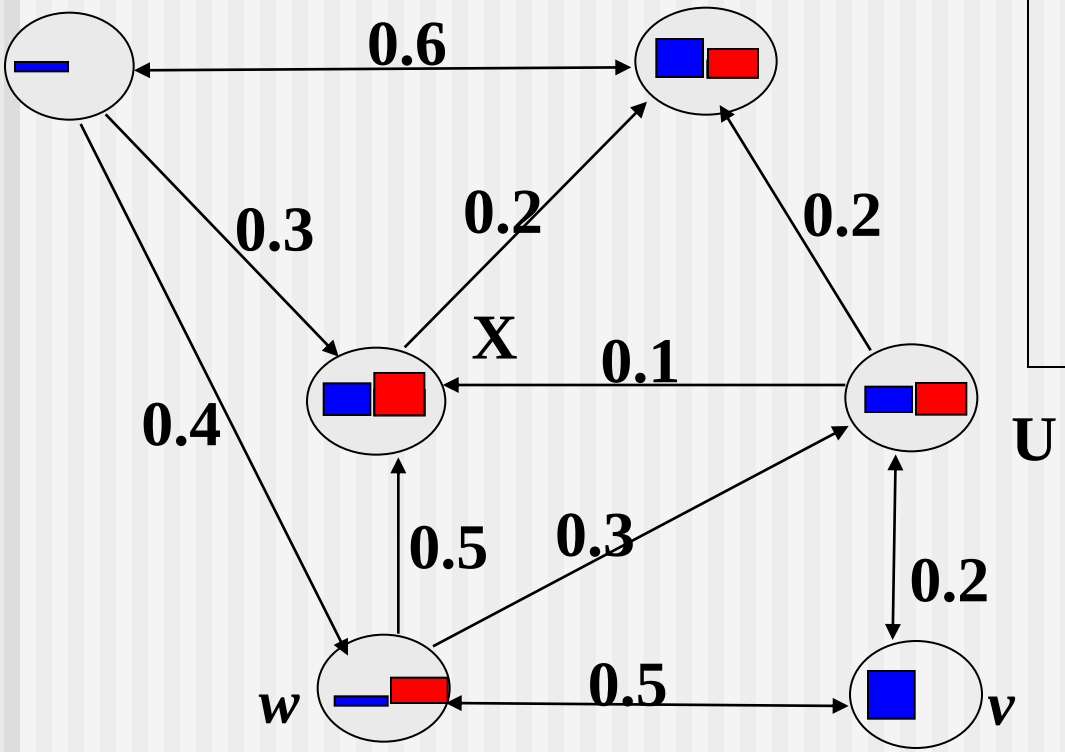
- A node v has random threshold $\theta_v \sim U[0,1]$
- A node v is influenced by each neighbor w according to a *weight* b_{vw} such that

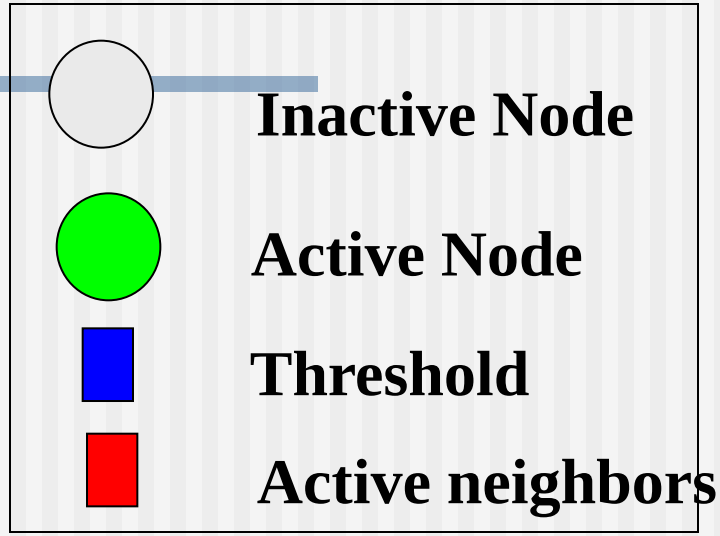
$$\sum_{w \text{ neighbor of } v} b_{v,w} \leq 1$$

- A node v becomes active when at least (weighted) θ_v fraction of its neighbors are active

$$\sum_{w \text{ active neighbor of } v} b_{v,w} \geq \theta_v$$

Example



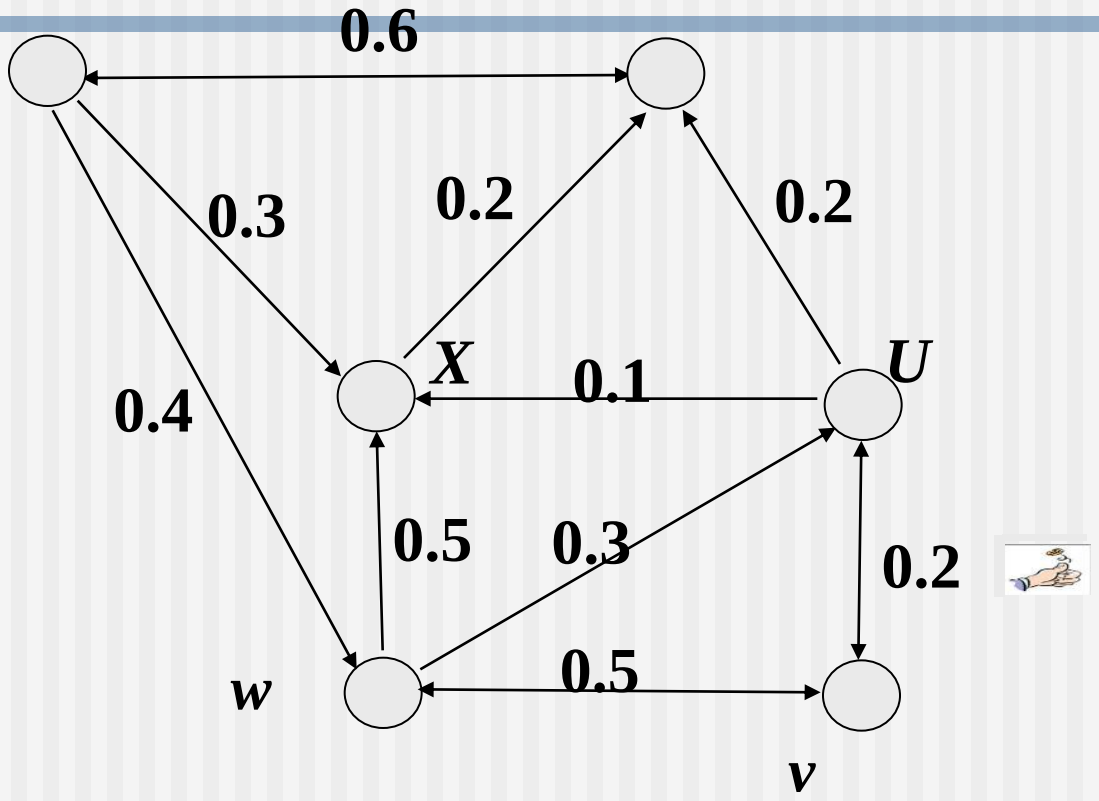


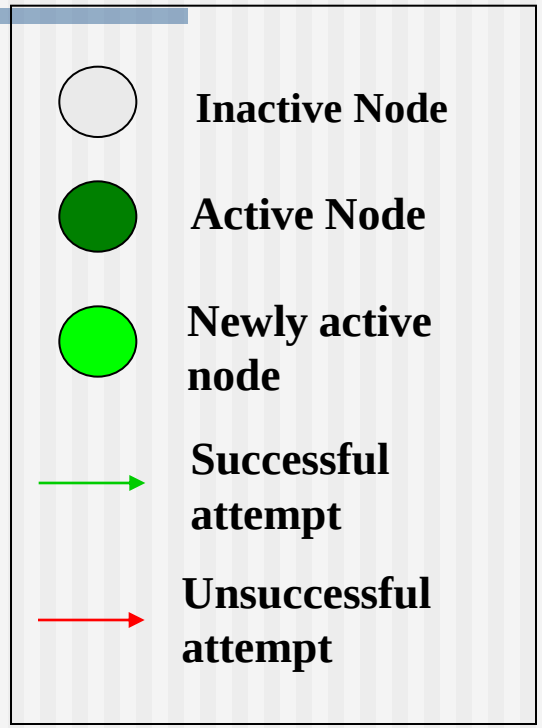
Stop!

Independent Cascade Model

- When node *v* becomes active, it has a single chance of activating each currently inactive neighbor *w.*
- The activation attempt succeeds with probability p_{vw} .

Example





Stop!

Influence Maximization Problem

- Influence of node set S: f(S)
  - expected number of active nodes at the end, if set S is the initial active set
- Problem:
  - Given a parameter *k* (budget), find a *k*-node set *S* to maximize f(S)
  - Constrained optimization problem with f(S) as the objective function

f(S): properties (to be demonstrated)

- Non-negative (obviously)
- Monotone: $f(S+v) \geq f(S)$
- Submodular:
  - Let *N* be a finite set
  - A set function $f : 2^N \mapsto \Re$ is submodular *iff*

$$\forall S \subset T \subset N, \forall v \in N \setminus T,$$

$$f(S+v) - f(S) \geq f(T+v) - f(T)$$

(diminishing returns)

Bad News

- For a submodular function *f*, if *f* only takes non-negative value, and is monotone, finding a *k*-element set *S* for which *f(S)* is maximized is an NP-hard optimization problem[GFN77, NWF78].
- It is NP-hard to determine the optimum for influence maximization for both independent cascade model and linear threshold model.

Good News

- We can use Greedy Algorithm!
  - Start with an empty set S
  - For k iterations:

    Add node v to S that maximizes $f(S+v) - f(S)$.
- How good (bad) it is?
  - Theorem: The greedy algorithm is a $(1 - 1/e)$ approximation.
  - The resulting set S activates at least $(1- 1/e) > 63\%$ of the number of nodes that any size-k set S could activate.

Outline

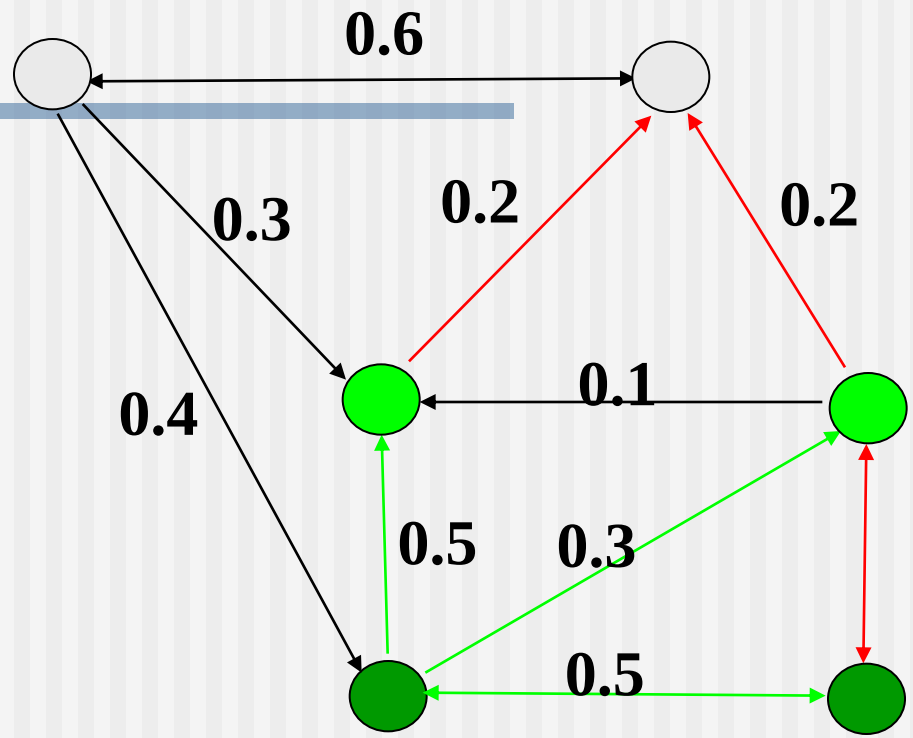
- Models of influence
  - Linear Threshold
  - Independent Cascade
- Influence maximization problem
  - Algorithm
  - Proof of performance bound
  - Compute objective function
- Experiments
  - Data and setting
  - Results

Key 1: Prove submodularity

$$\forall S \subset T \subset N, \forall v \in N \setminus T,$$
$$f(S+v) - f(S) \geq f(T+v) - f(T)$$

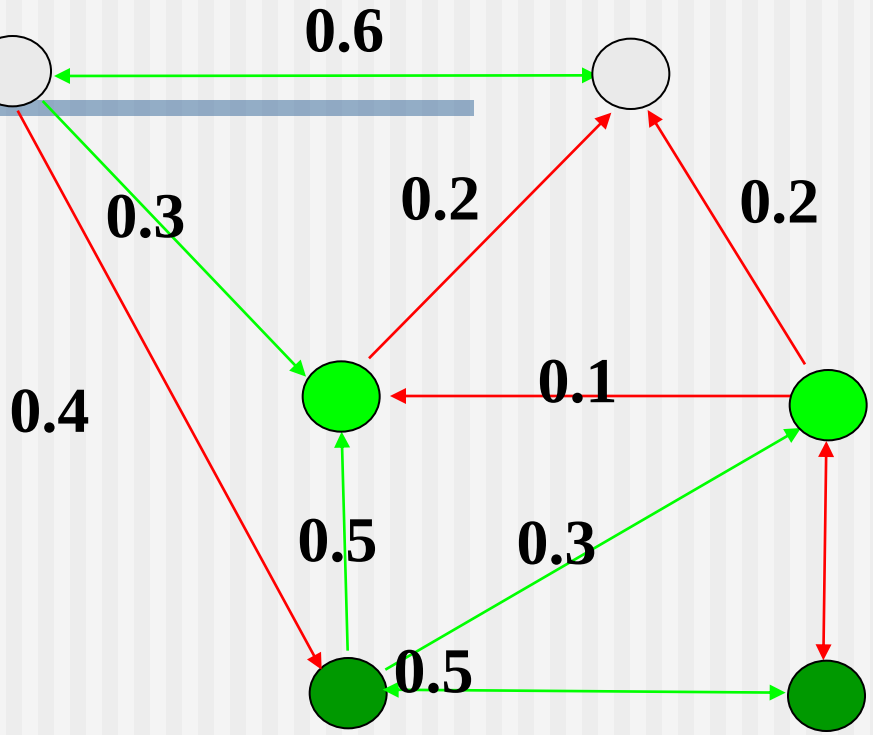
Submodularity for Independent Cascade

- Coins for edges are flipped during activation attempts.



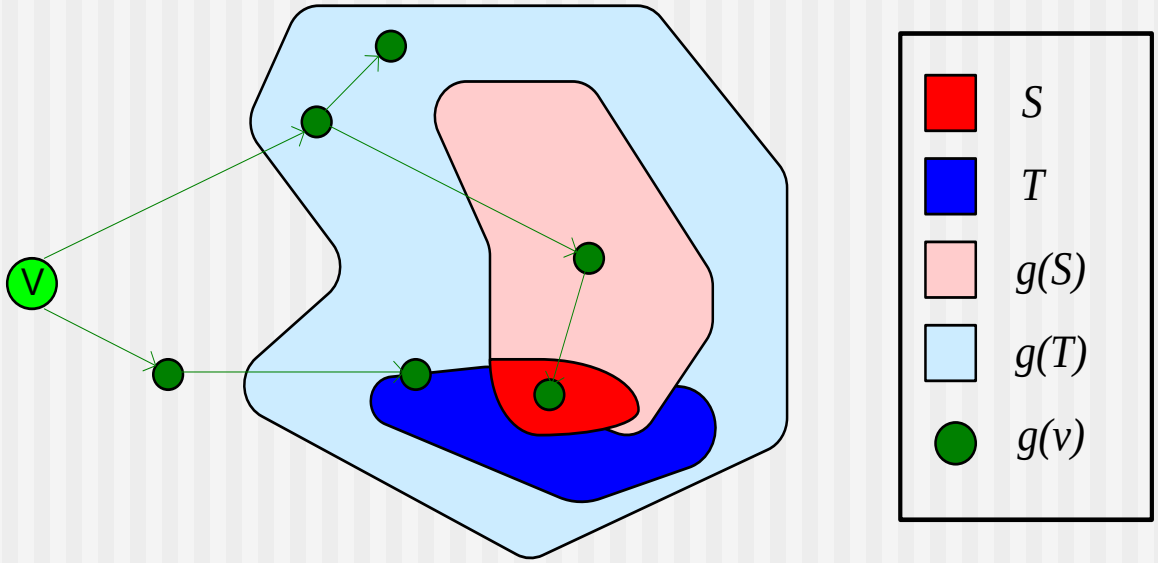
Submodularity for Independent Cascade

- Coins for edges are flipped during activation attempts.
- Can pre-flip all coins and reveal results immediately.



- Active nodes in the end are reachable via green paths from initially targeted nodes.
- Study reachability in green graphs

Submodularity, Fixed Graph



Submodularity of the Function

Fact: A non-negative linear combination of submodular functions is submodular

$$f(S) = \sum_{G} \text{Prob}(G \textit{ is green graph}) \cdot g_G(S)$$

- $g_G(S)$: nodes reachable from S in G.
- Each $g_G(S)$: is submodular (previous slide).
- Probabilities are non-negative.

Submodularity for Linear Threshold

- Use similar “green graph” idea.
- Once a graph is fixed, “reachability” argument is identical.
- How do we fix a green graph now?
- Each node picks at most one incoming edge, with probabilities proportional to edge weights.
- Equivalent to linear threshold model (trickier proof).

Key 2: Evaluating *f(S)*

Evaluating $f(S)$

- How to evaluate $f(S)$?
- Still an open question of how to compute efficiently
- But: very good estimates by simulation
  - repeating the diffusion process often enough (polynomial in n; $1/\varepsilon$)
  - Achieve $(1 \pm \varepsilon)$-approximation to $f(S)$.
- Generalization of Nemhauser/Wolsey proof shows: Greedy algorithm is now a $(1\text{-}1/e\text{-}\varepsilon')$-approximation.

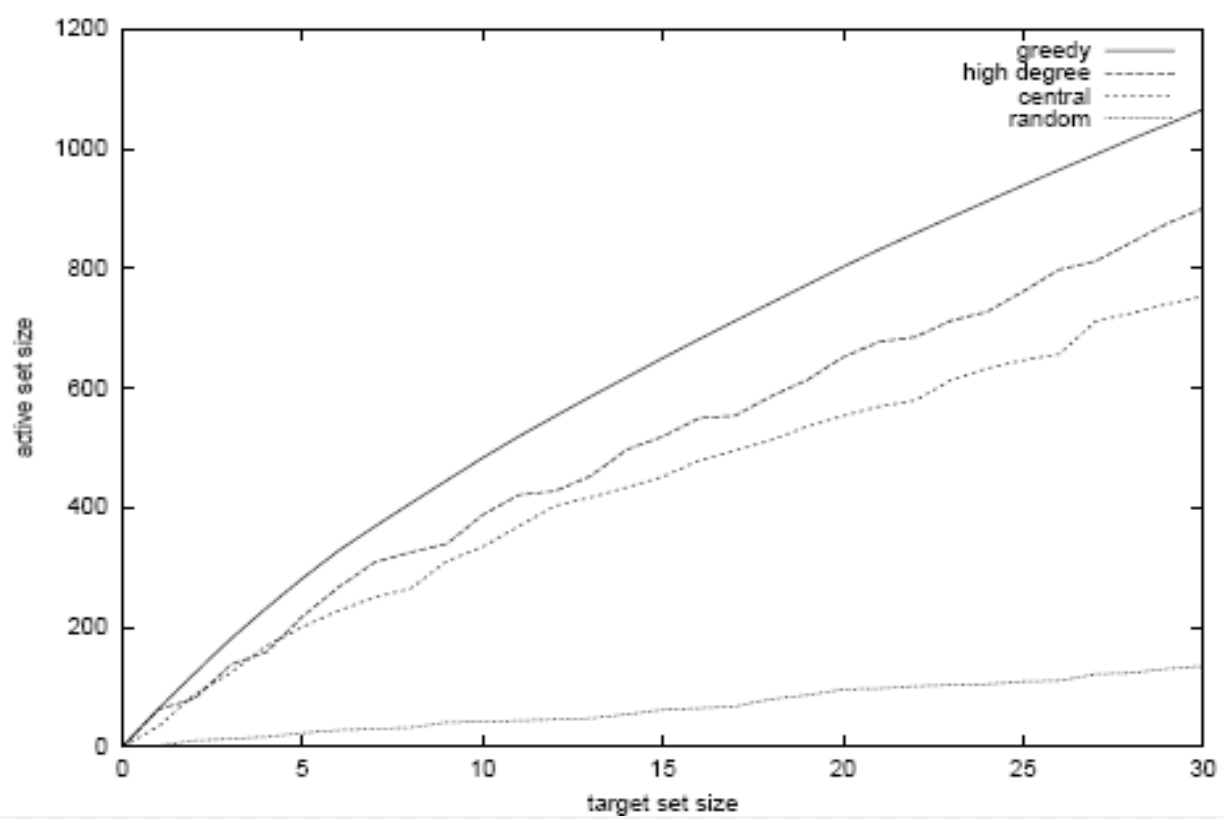
Experiment Data

- A collaboration graph obtained from co-authorships in papers of the arXiv high-energy physics theory section
- co-authorship networks arguably capture many of the key features of social networks more generally
- Resulting graph: 10748 nodes, 53000 distinct edges

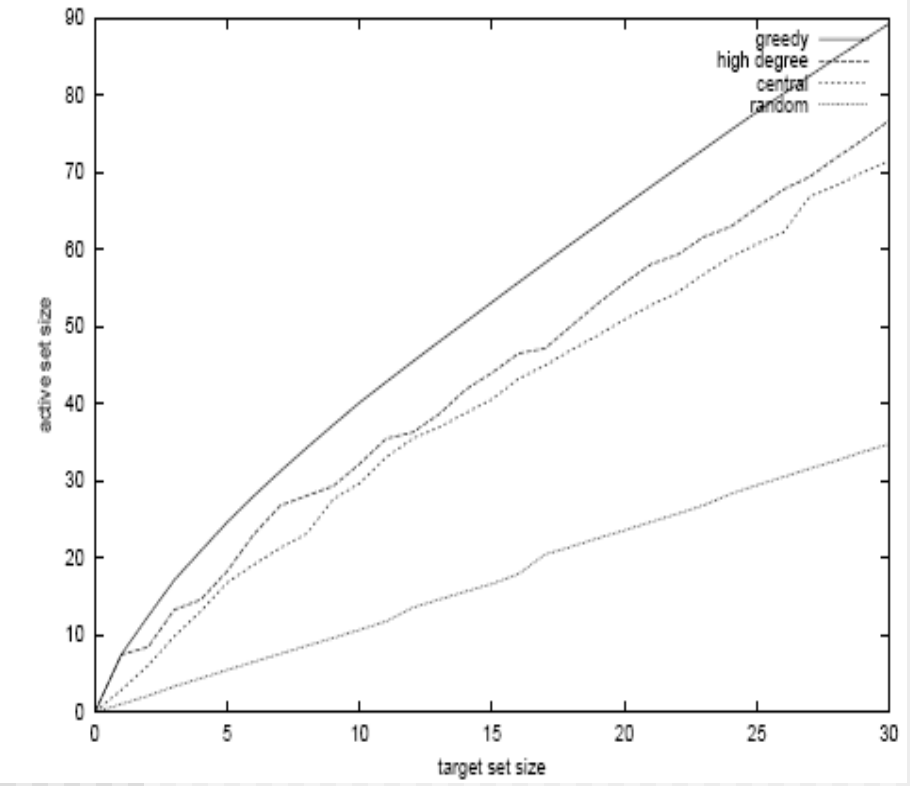
Experiment Settings

- Linear Threshold Model: multiplicity of edges as weights
  - weight(v→ω) = C_{vw} / dv, weight(ω→v) = C_{wv} / dw
- Independent Cascade Model:
  - Case 1: uniform probabilities p on each edge
  - Case 2: edge from v to ω has probability $1/ d\omega$ of activating ω.
- Simulate the process 10000 times for each targeted set, re-choosing thresholds or edge outcomes pseudo-randomly from [0, 1] every time
- Compare with other 3 common heuristics
  - (in)degree centrality, distance centrality, random nodes.

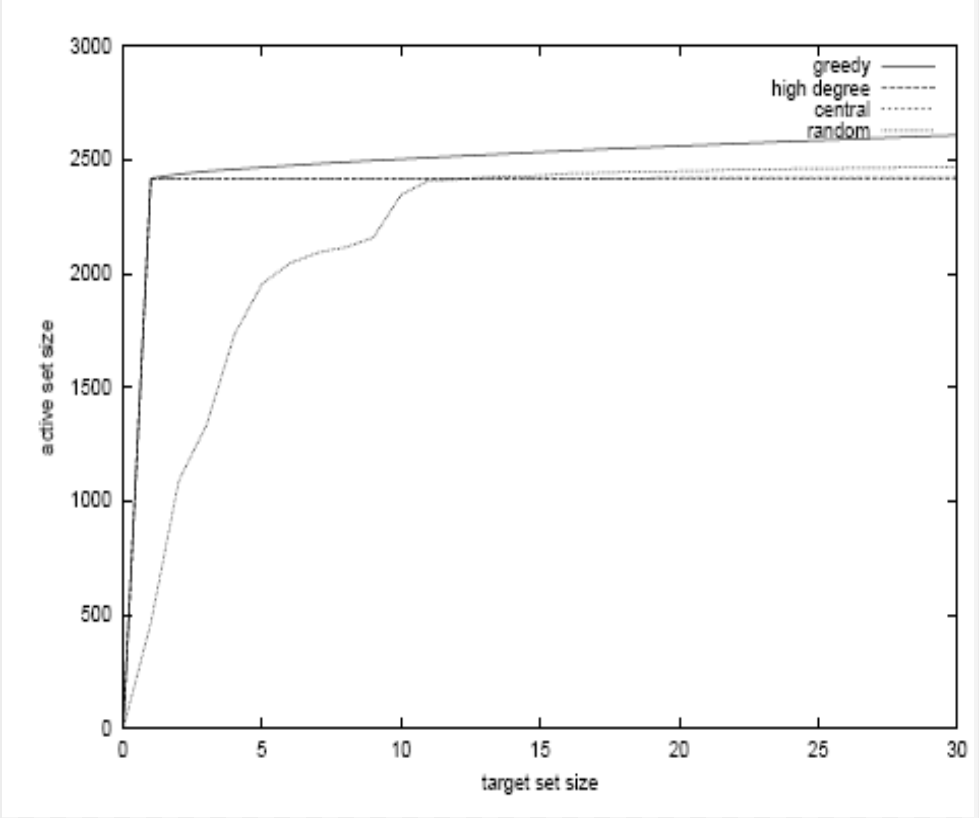
Results: linear threshold model



Independent Cascade Model – Case 1

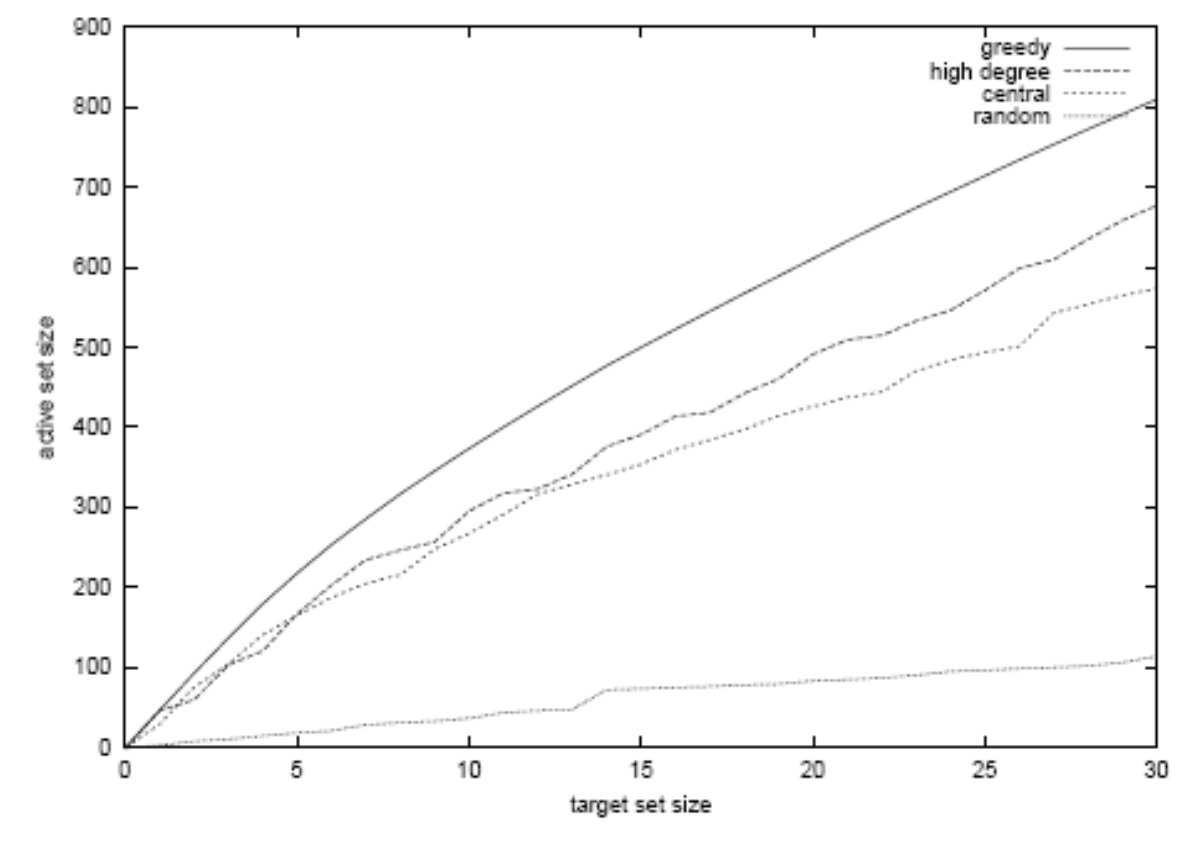


P = 1%

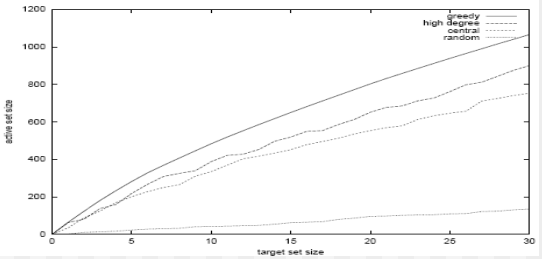


P = 10%

Independent Cascade Model – Case 2



Reminder: linear threshold model



More in the Paper

- A broader framework that simultaneously generalizes the two models
- Non-progressive process: active nodes CAN deactivate.
- More realistic marketing:
  - different marketing actions increase **likelihood** of initial activation, for **several** nodes at once.

Open Questions

- Study more general influence models. Find

  trade-offs between generality and feasibility.
- Deal with negative influences.
- Model competing ideas.
- Obtain more data about how activations occur in real social networks.

Efficient Influence Maximization in Social Networks

Wei Chen, Yajun Wang, Siyu Yang

Problem

- Influence maximization – finding a small subset of nodes in a social network that could maximize the spread of influence
- In other words, find a core group of people that can influence the most amount of people

Outline

- Situation
- Definitions
- Contents of this paper
- Original method
- Improved method
- Degree Discount Heuristics
- Results

Situation

- Small start-up
- Create a new cool app
- Want to get a lot of people to use it
- Convince a few people yourself to use it and hope that it spreads by word of mouth

Definitions

- Social Network is a graph G
- Vertex:person, Edge:Connection/Relationship
- k is the seed set size
- Cascade models are how we decide whether people are influenced
- Given graph G, int k, and a cascade model, output a seed set that would maximize influence (Called the influence spread)

This Paper

- Look at a greedy algorithm that has been implemented
- Look to improve on that algorithm
- Create heuristics to improve run time while not sacrificing to much accuracy

Cascade Models

- How we decide if people are influenced
- Ex) independent, weight, linear threshold
- Could be random (as with original greedy)
- Probability p is the propagation probability
- If a vertex has l neighbors in the seed set then it has a 1 – (1 – p)^l chance of being included in the next round

Original Greedy

- Kempe, Kleinberg, Tardos
- In each round add a vertex into S such that this vertex and S maximize the influence spread
- Influence spread is estimated with R repeated simulations of RanCas(S union v)
- How we decide which vertex to add
- Guarantees influence spread with (1 – 1/e) of the optimal
- Drawback is efficiency-need to compute influence spreads multiple times
- 15,000 nodes takes a few days to compete

Original Greedy

Algorithm 1 GeneralGreedy(G, k)

initialize $S = \emptyset$ and $R = 20000$
for $i = 1$ to k **do**
  for each vertex $v \in V \setminus S$ **do**
    $s_v = 0$.
    for $i = 1$ to R **do**
      $s_v \mathrel{+}= |RanCas(S \cup \{v\})|$
    end for
    $s_v = s_v / R$
  end for
  $S = S \cup \{\arg\max_{v \in V \setminus S}\{s_v\}\}$
end for
output S.

CELF

- An improvement to the greedy algorithm already implemented
- Uses the submodularity property – when adding v into S, the incremental influence spread as the result of adding v is larger if S is smaller
- Same influence spread, 700 times faster

Improved Greedy

- Uses different cascade models
- For Independent Cascade construct a graph G’
- Obtain G’ by removing all edges not for propagation from G with probability 1-p
- Use DFS/BFS to find out the set of vertices reachable from S in G’ (also for all vEV)

- Same influence spread with 15-34% shorter run time

Improved Greedy IC

Algorithm 2 NewGreedyIC(G, k)

initialize $S = \emptyset$ and $R = 20000$
for $i = 1$ to k **do**
  set $s_v = 0$ for all $v \in V \setminus S$
  for $i = 1$ to R **do**
    compute G' by removing each edge from G with probability $1 - p$
    compute $R_{G'}(S)$
    compute $|R_{G'}(\{v\})|$ for all $v \in V$
    for each vertex $v \in V \setminus S$ **do**
      if $v \notin R_{G'}(S)$ **then**
        $s_v \mathrel{+}= |R_{G'}(\{v\})|$
      end if
    end for
  end for
  set $s_v = s_v / R$ for all $v \in V \setminus S$
  $S = S \cup \{\arg\max_{v \in V \setminus S}\{s_v\}\}$
end for
output S

Improved Mix

- In CELF we must consider all the vertices to be added in the first round, but then we can decrease in future rounds
- New greedy we traverse the graph R times, thus we combine them
- First use new greedy and then switch to CELF optimization

Improved Weighted Model

- For the weighted model, we use the degrees of each vertex
- Here u activating v is not the same probability of v activating u
- Use the same idea as the IC model but now with different probabilities
- Here G’ is directed so we use Cohen’s algorithm to find the reachability

Cohen's Algorithm

- Traverse G' and create a DAG of strongly connected components
- Let node weight in DAG be how many nodes of G' it represents
- In T iterations take a random same according to an exponential distribution with the probability density function

$$w(v^*)e^{-w(v^*)x},\ x \geq 0$$

Improved Greedy WC

Algorithm 3 NewGreedyWC(G, k)

initialize $S = \emptyset$, $R = 20000$, $T = 5$.
for $i = 0$ to k **do**
  initialize $s_v = 0$ for all vertices.
  for $j = 1$ to R **do**
    obtain $G' = RanWC(G)$
    compute DAG G'^* and weights $w(v^*)$ for all $v^* \in V^*$
    for $\ell = 1$ to T **do**
      for each $v^* \in V^*$, $s^\ell_{v^*} = 0$
      for each $v^* \in V^*$, generate random value $X^\ell_{v^*}$ from the exponential distribution with mean $1/w(v^*)$
      for each $v^* \in V^*$, compute $Y^\ell_{v^*} = \min_{u^* \in R_{G'^*}(S^* \cup \{v^*\})} X^\ell_{u^*}$
      for each $v^* \in V^*$, $s^\ell_{v^*} \mathrel{+}= Y^\ell_{v^*}$
    end for
    for each $v \in V \setminus S$, $s_v \mathrel{+}= (T-1)/s^\ell_{v^*}$
  end for
  set $s_v = s_v/R$ for all $v \in V \setminus S$
  $S = S \cup \{\arg\max_{v \in V \setminus S}\{s_v\}\}$
end for
output S

Degree Discount

- Consider edge uv, with u in the seed set and v being considered
- Since u is in the seed set, that neighbor should not be counted towards v's degree
- Same for all of v's neighbors in the seed set
- Let Star(v) be the subgraph with v and all of its neighbors, only with edges from v to its neighbors
- t_v is number of neighbors of v already in seed set

Degree Discount

THEOREM 2. *In the IC model with propagation probability p, suppose that $d_v = O(1/p)$ and $t_v = o(1/p)$ for a vertex v.* [4] *The expected number of additional vertices in $Star(v)$ influenced by selecting v into the seed set is:*

$$1 + (d_v - 2t_v - (d_v - t_v)t_v p + o(t_v)) \cdot p. \quad (1)$$

PROOF. Let S_v be the set of t_v neighbors of v that have been selected as seeds. The probability that v is influenced by its immediate neighbors is $1 - (1-p)^{t_v}$. In this case, selecting v as a seed does not contribute additional influence in the graph.

If v is not influenced by any of the already selected seeds, which occurs with probability $(1-p)^{t_v}$, then the additional vertices in $Star(v)$ influenced by selecting v into the seed set include: (a) v itself with probability 1; and (b) each u in the remaining $d_v - t_v$ neighbors with probability p. Thus the additional vertices in $Star(v)$ influenced by v is $1 + (d_v - t_v) \cdot p$. Hence the overall expected number of additional vertices in $Star(v)$ influenced by v is

$$\begin{aligned}
&(1-p)^{t_v} \cdot (1 + (d_v - t_v) \cdot p) \\
&= (1 - t_v p + o(t_v p)) \cdot (1 + (d_v - t_v) \cdot p) \\
&\qquad \{\text{since } t_v p = o(1)\} \\
&= 1 + (d_v - 2t_v)p - (d_v - t_v)t_v p^2 + o(t_v p) \\
&\qquad \{\text{since } (d_v - t_v)p = O(d_v p) = O(1)\} \\
&= 1 + (d_v - 2t_v - (d_v - t_v)t_v p + o(t_v))p.
\end{aligned}$$

□

Degree Discount

Algorithm 4 DegreeDiscountIC(G, k)

initialize $S = \emptyset$
for each vertex v **do**
    compute its degree d_v
    $dd_v = d_v$
    initialize t_v to 0
end for
for $i = 1$ to k **do**
    select $u = \arg\max_v \{dd_v \mid v \in V \setminus S\}$
    $S = S \cup \{u\}$
    for each neighbor v of u and $v \in V \setminus S$ **do**
        $t_v = t_v + 1$
        $dd_v = d_v - 2t_v - (d_v - t_v)t_v p$
    end for
end for
output S

Results

- Improved greedy algorithm achieves better run time than original with matching influence spread
- Heuristics run in milliseconds and achieve nearly same influence spreads

Results

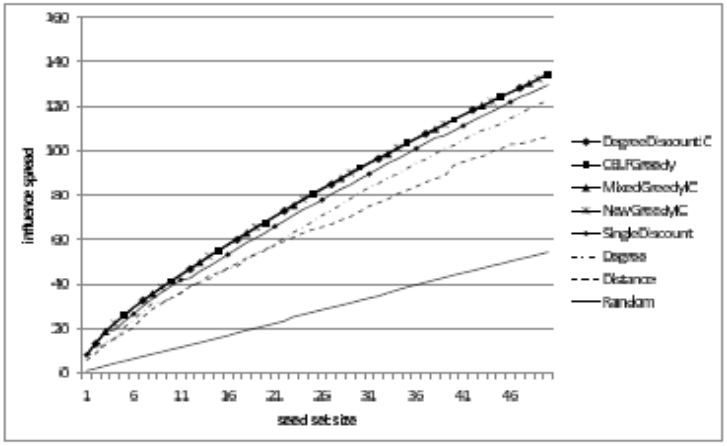


Figure 1: Influence spreads of different algorithms on the collaboration graph NetHEPT under the independent cascade model ($n = 15,233$, $m = 58,891$, and $p = 0.01$).

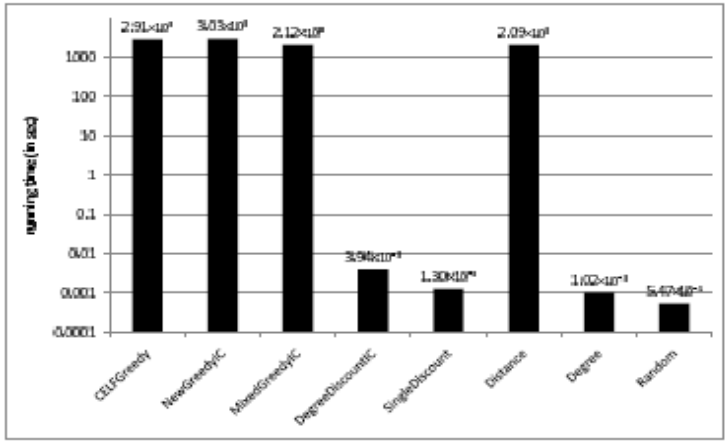


Figure 3: Running times of different algorithms on the collaboration graph NetHEPT under the independent cascade model ($n = 15,233$, $m = 58,891$, $p = 0.01$, and $k = 50$).

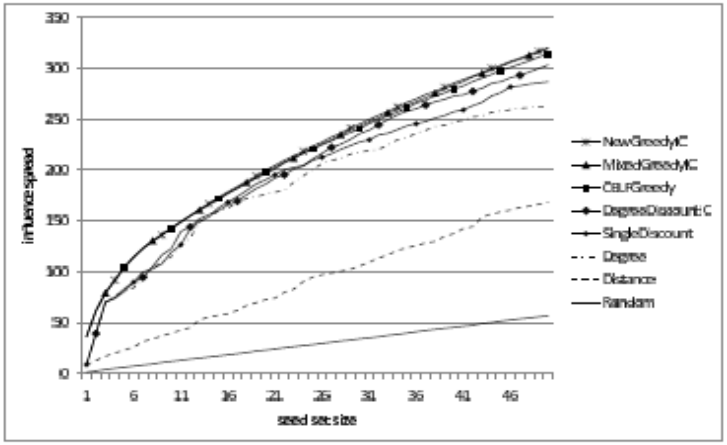


Figure 2: Influence spreads of different algorithms on the collaboration graph NetPHY under the independent cascade model ($n = 37,154$, $m = 231,584$, and $p = 0.01$).

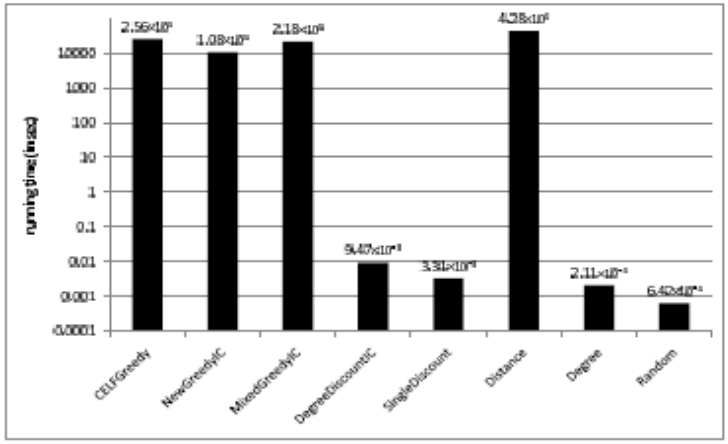


Figure 4: Running times of different algorithms on the collaboration graph NetPHY under the independent cascade model ($n = 37,154$, $m = 231,584$, $p = 0.01$, and $k = 50$).

Results

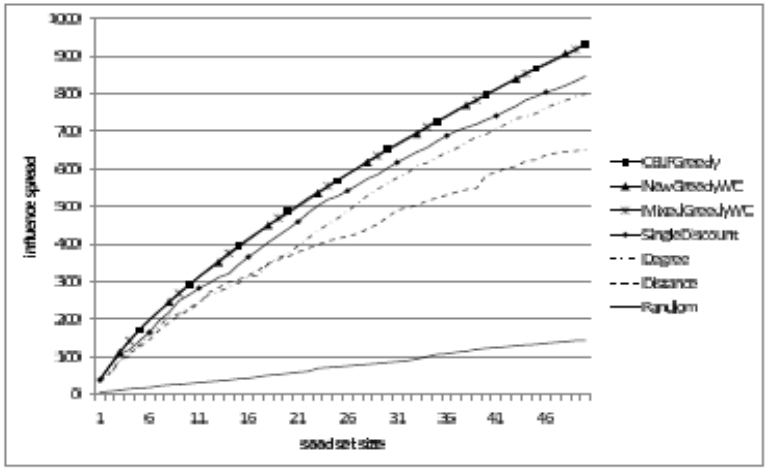


Figure 5: Influence spreads of different algorithms on the collaboration graph NetHEPT under the weighted cascade model ($n = 15,233$ and $m = 58,891$).

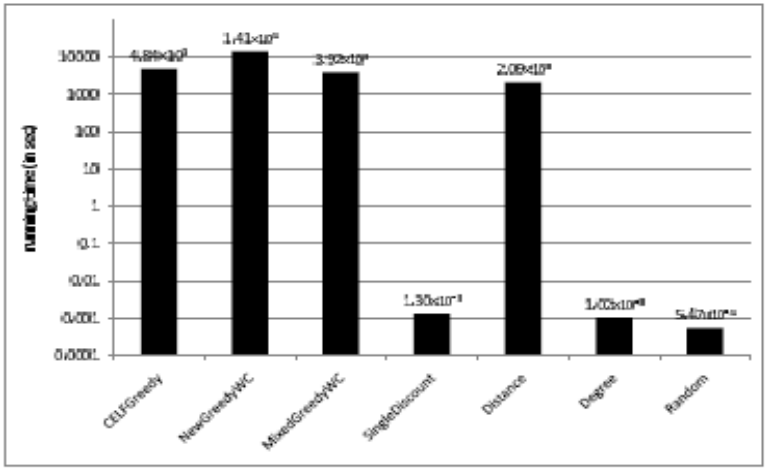


Figure 7: Running times of different algorithms on the collaboration graph NetHEPT under the weighted cascade model ($n = 15,233$, $m = 58,891$, and $k = 50$).

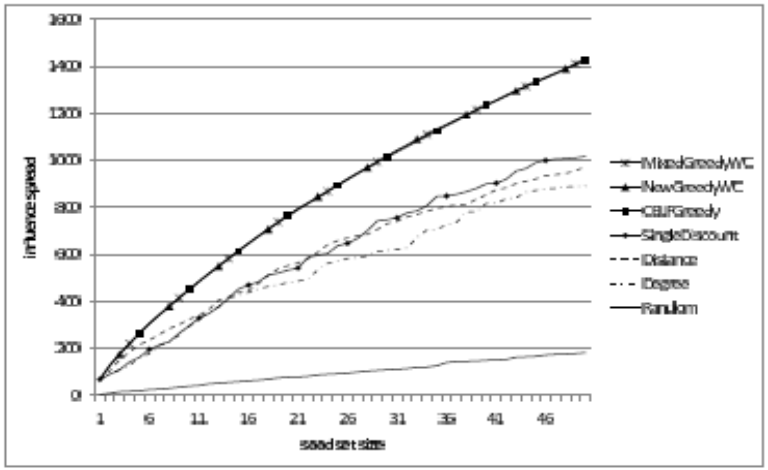


Figure 6: Influence spreads of different algorithms on the collaboration graph NetPHY under the weighted cascade model ($n = 37,154$ and $m = 231,584$).

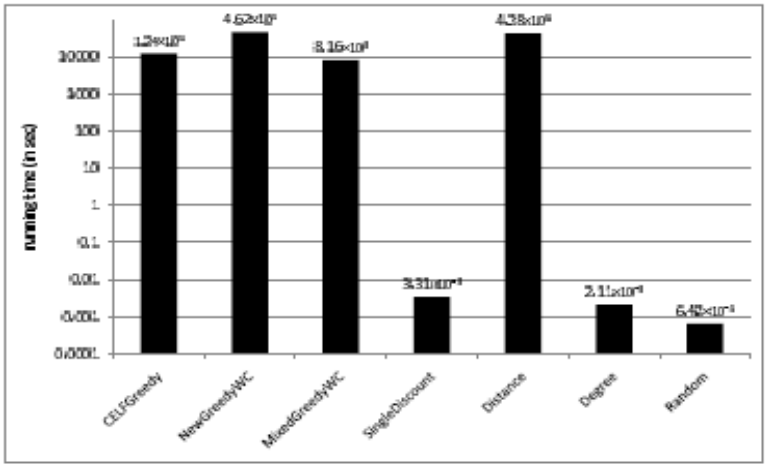


Figure 8: Running times of different algorithms on the collaboration graph NetPHY under the weighted cascade model ($n = 37,154$, $m = 231,584$, and $k = 50$).

Results

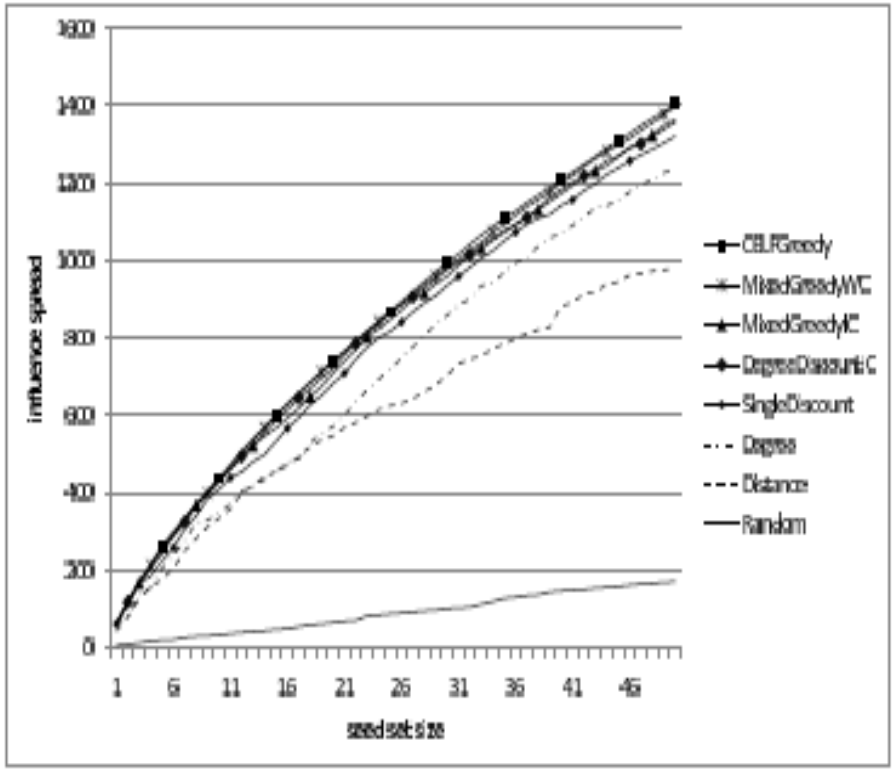


Figure 9: Influence spreads of different algorithms on the collaboration graph NetHEPT under the linear threshold model ($n = 15,233$ and $m = 58,891$).

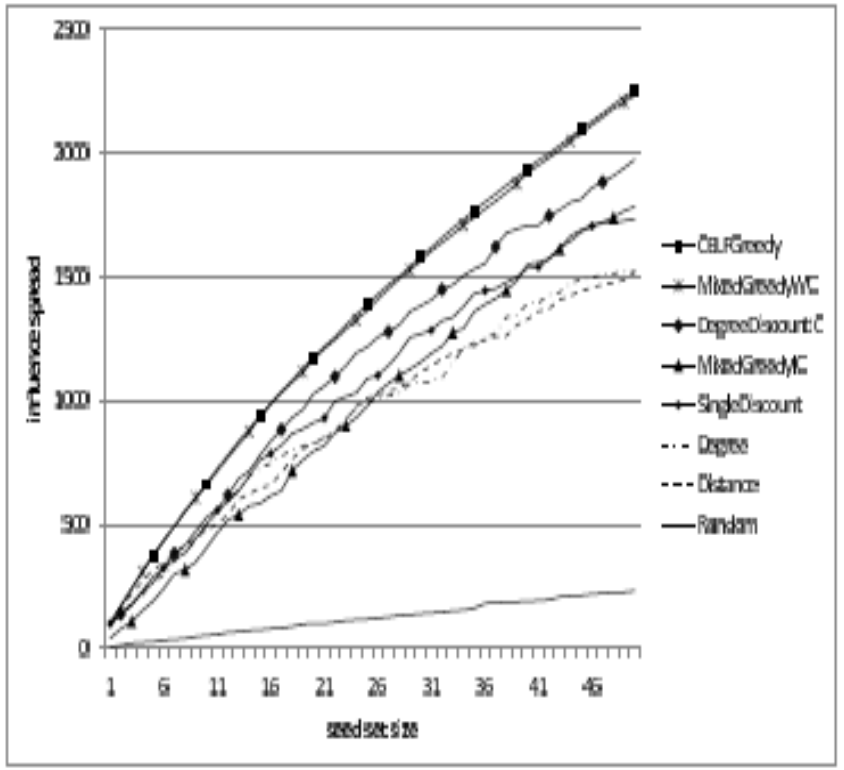


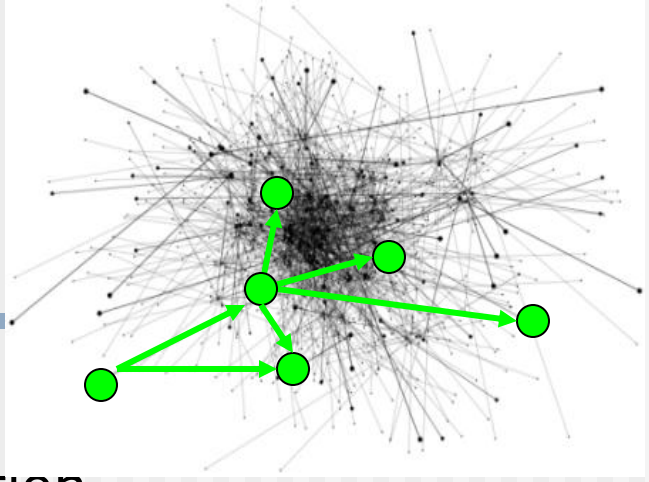
Figure 10: Influence spreads of different algorithms on the collaboration graph NetPHY under the linear threshold model ($n = 37,154$ and $m = 231,584$).

Cascading Behavior in Large Blog Graphs

--Patterns and a model

Authors: Jure Leskovec, Mary McGlohon, Christos Faloutsos Natalie Glance, Matthew Hurst

Introduction

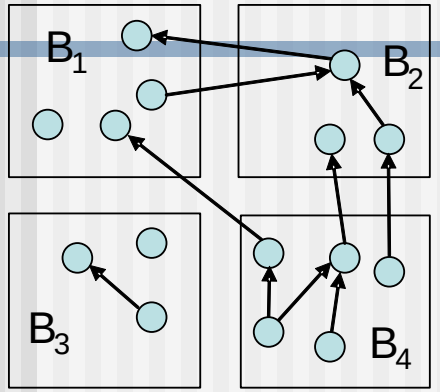


- Blog / 博客/ 部落格...
  - an important medium of information
  - a publicly available record of how information and influence spreads through a social network
- Blogosphere: the collective term encompassing all blogs linked together forming as a community or social network.
- Information Cascade: phenomena in which an idea becomes adopted due to influence by others

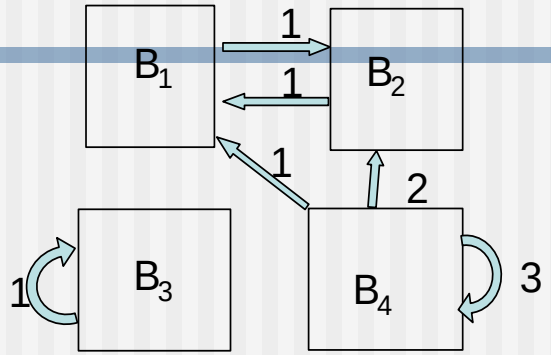
Research Questions

- Temporal questions: How does popularity die off? Is there burstiness/periodicity?
- Topological questions: What topological patterns do posts and blogs follow? What are the characteristic (size, shape, etc.) of a cascade?
- Generative model: Can we build model that generate realistic cascades?

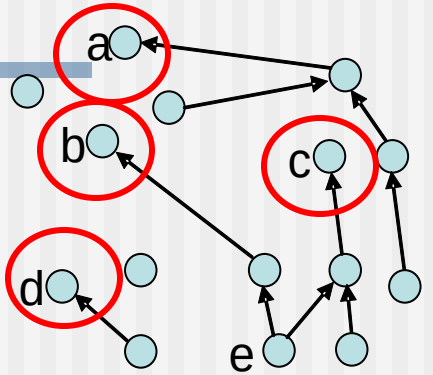
Preliminaries



(a) Blogsphere
blogs + posts

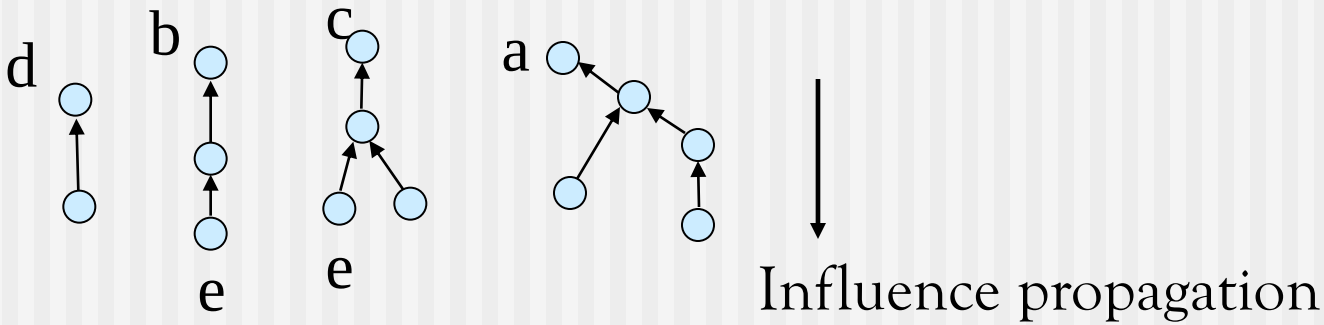


(b) Weighted Blog Network
Links among blogs



(c) Post Network
Links among posts

Extracted (Nontrivial) Cascades: sub-graph induced by a time ordered propagation of information (edges)

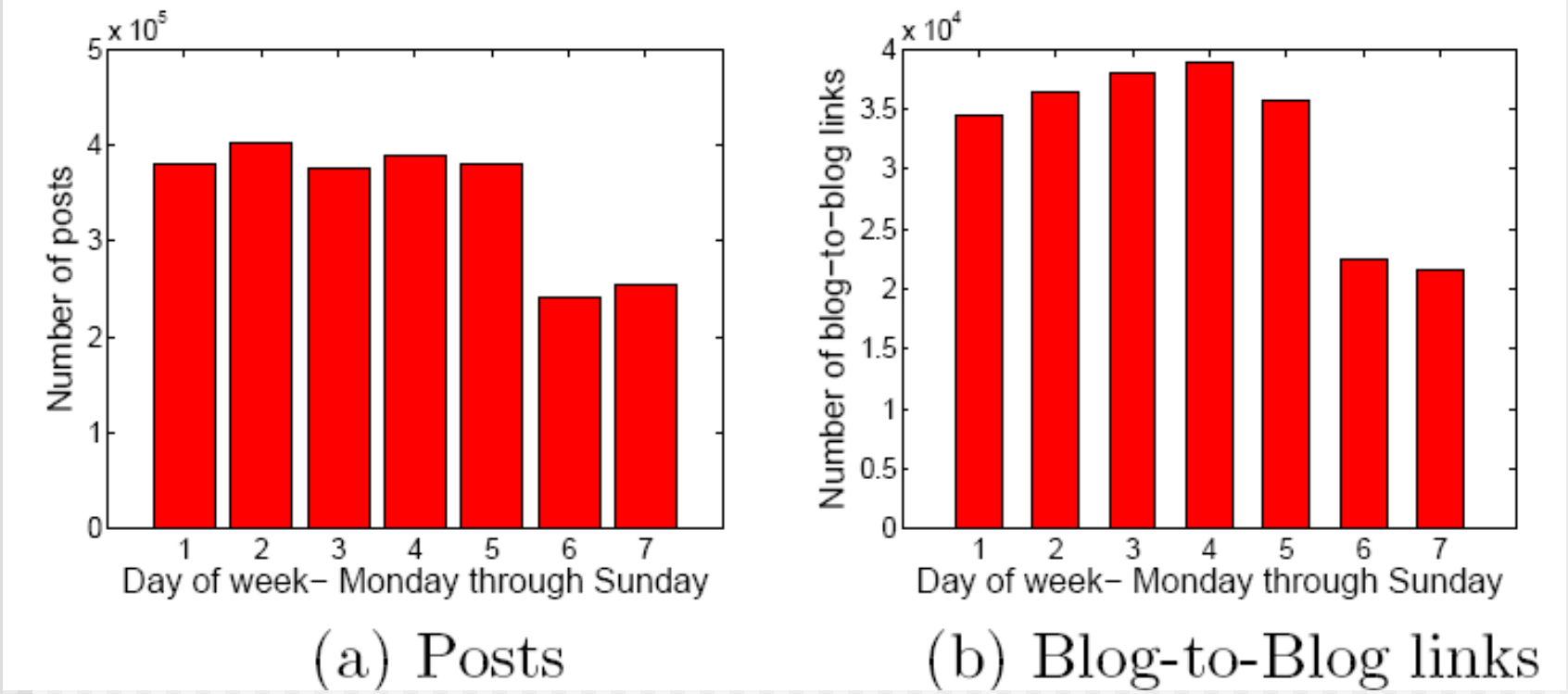


Blog Dataset

- Constructed from another larger dataset *
- 45,000 blogs participating in cascades (biased towards the active part of the blogospher)
- All their posts for 3 months (Aug-Sept ‘05)
- 2.4 million posts
- ~5 million links (245,404 inside the dataset)

Temporal Observations

- Is there periodicity in blog traffic?
  - Yes. A week-end effect in both number of posts and number of links.

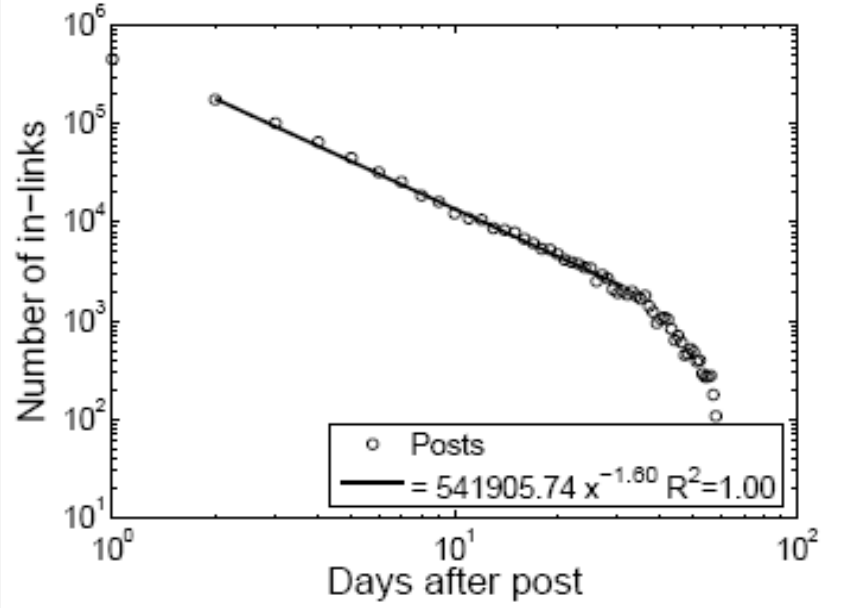


Temporal Observations

- How does a post's popularity grow over time?
  - Post popularity drop-off follows a power law

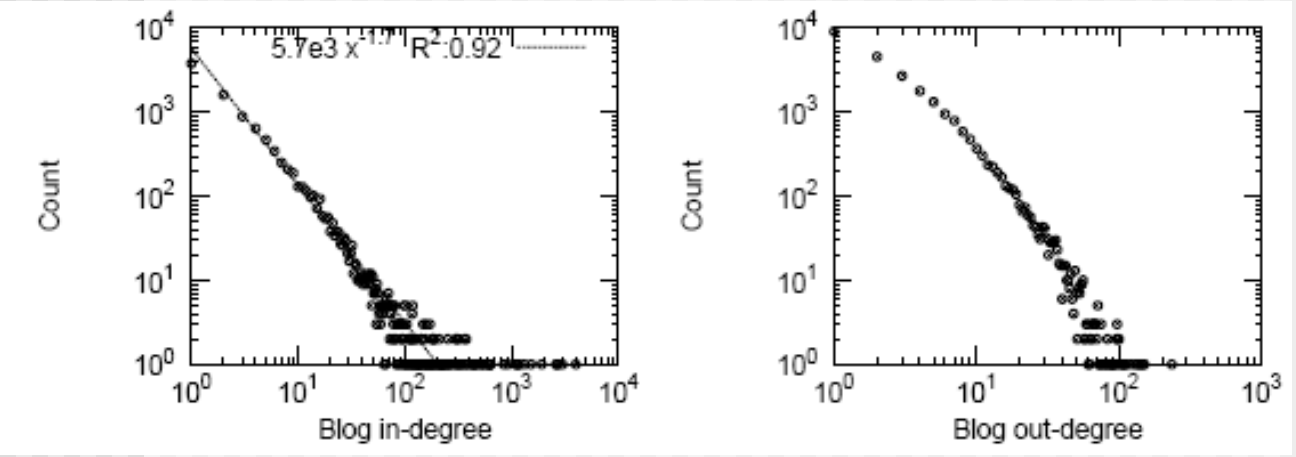
The probability that a post written at time t_p acquires a link at time $t_p + \Delta$ is:

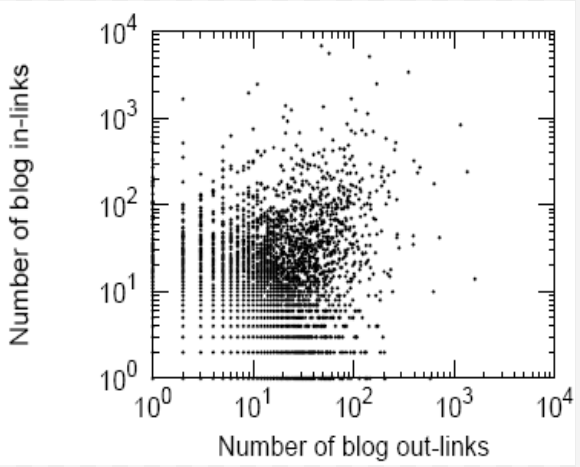
$$p(t_p+\Delta) \propto \Delta^{-1.5}$$



Topological Observations—Blog Network

- Half of blogs belong to largest connected component
- the other half are isolated

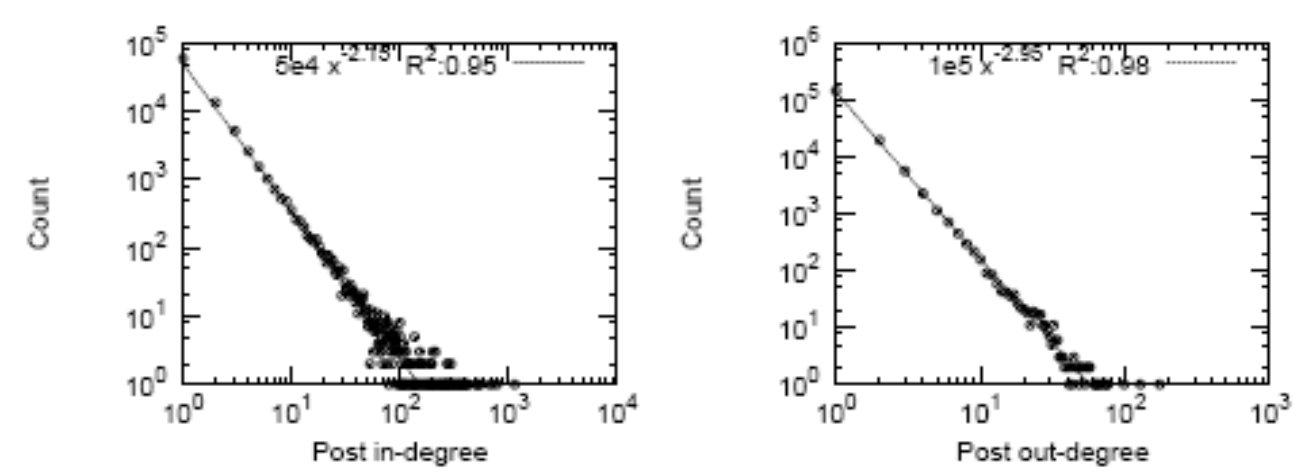




- Both In- and out-degree follow (heavy tailed) power law distribution. In-degree exponent 1.7, out 3 (but they are NOT correlated– $\rho = 0.16$).
- Strong rich-get-richer phenomena

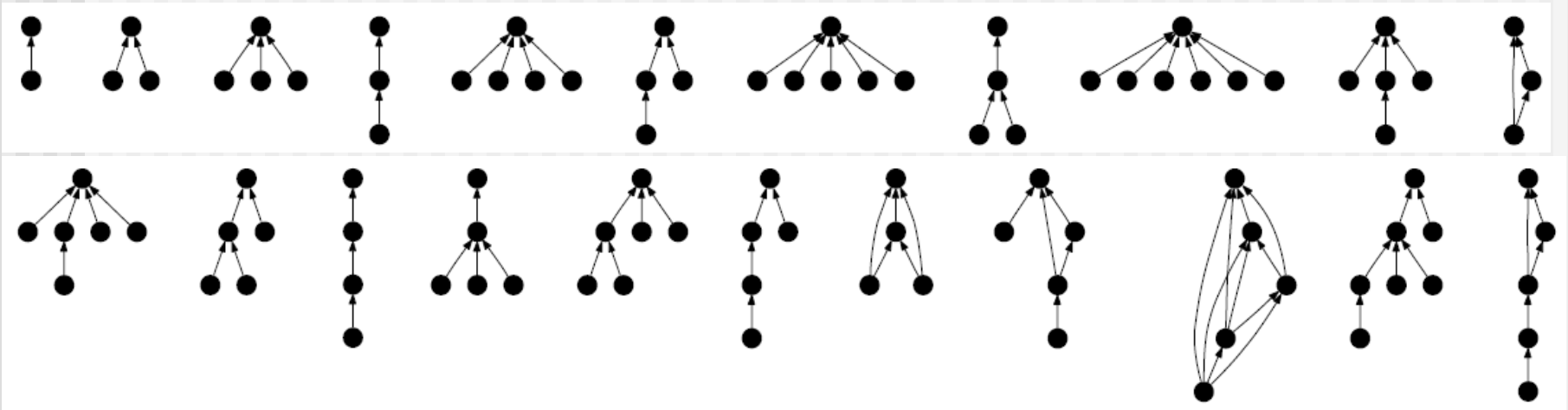
Topological Observations—Post Network

- Very sparsely connected:2.2 million nodes and only 205, 000 edges
- 98% of the posts are isolated
- In-degree and Out-degree follow power law with exponents −2.1 (In) and −2.9 (Out)



Topological Observations—Cascades

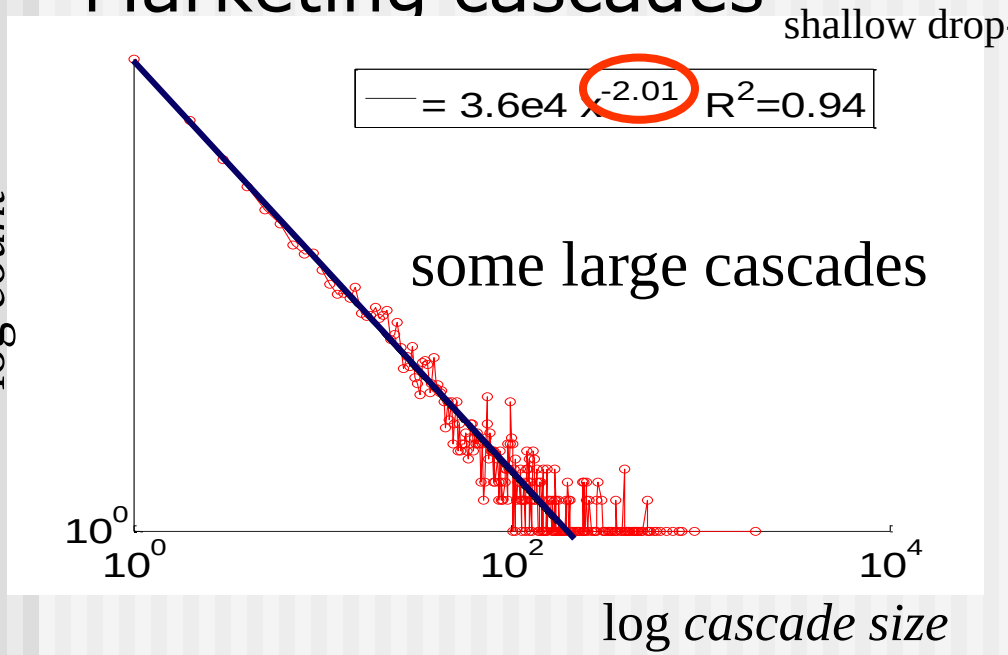
- Cascade shapes (ordered by frequency)



- Cascades are mostly tree-like, esp. stars
- Interesting relation between the cascade frequency and structure

Topological Observations—Cascades

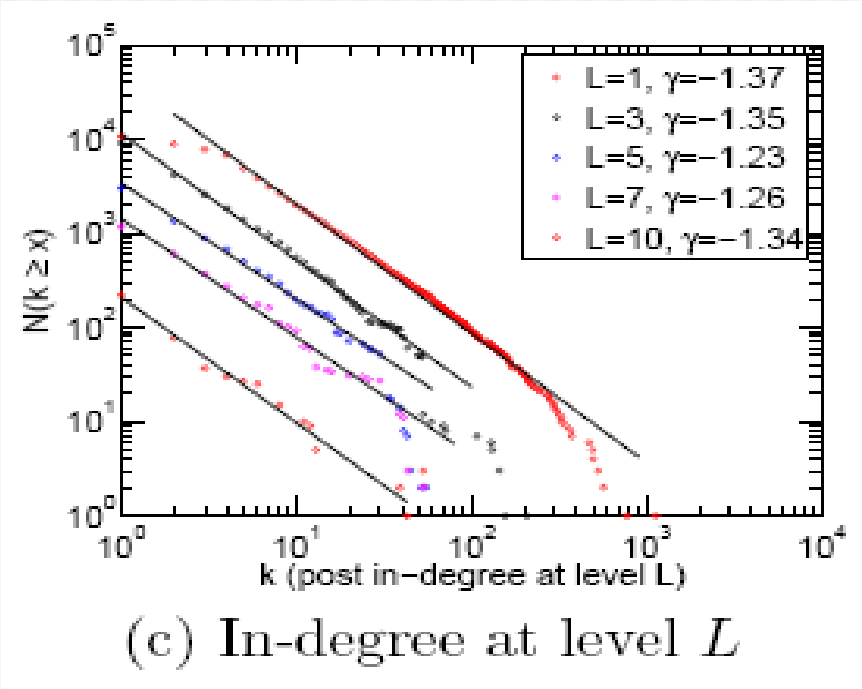
- Cascade size: how many posts participate in cascades
- Blog cascades tend to be larger than Viral Marketing cascades



The probability of observing a cascade on n nodes follows a Zipf distribution: $p(n) \propto n^{-2}$

Topological Observations—Cascades

- Also power laws in in/out-degree, size of different cascades (chains, stars) and degree per level.



(c) In-degree at level L

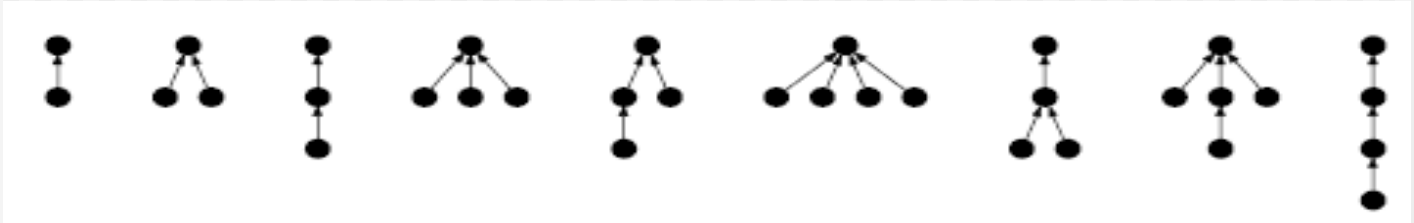
A Generative Model

- Model cascade generation as an epidemic
- Use Simple virus propagation type of model (SIS)
  - At any time, an entity is in one of two states: susceptible or infected.
  - One parameter β determines how infectious the virus is.
- Process
  - Randomly pick blog *u* to be infected, and add it to cascade
  - *u* infects each in-linked neighbor with probability $\beta(*)$
  - Add infected neighbors to cascade and link them to node *u*
  - Set *u* to be not infected. Continue step $(*)$ until no nodes are infected.

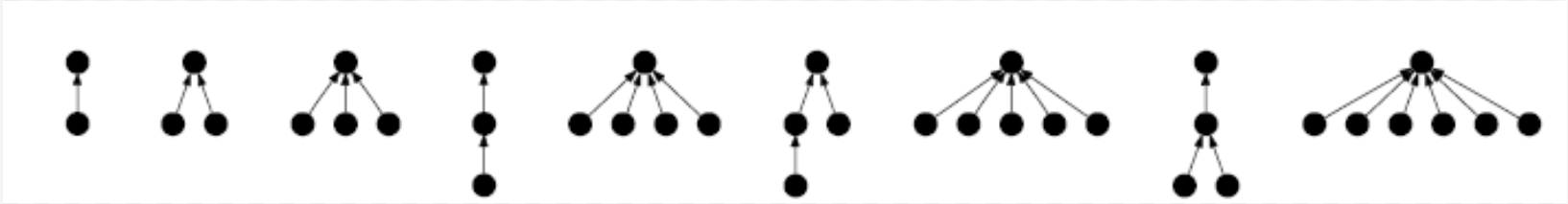
A Generative Model—Validation

- 10 simulations, 2 million cascades each time (β=.025)
- Top 10 (9?) most frequent cascades: 7 are matched exactly

Model generated:

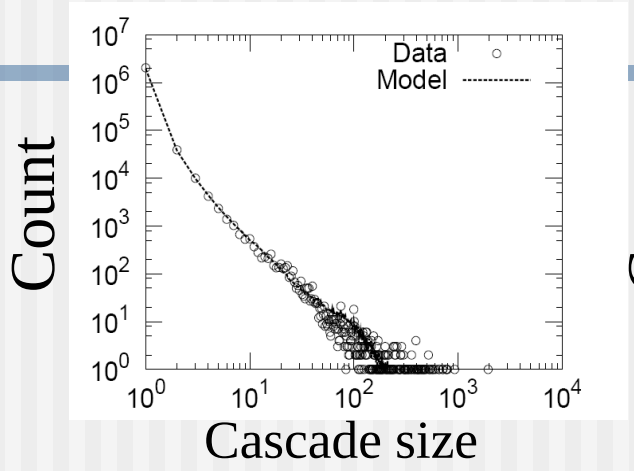


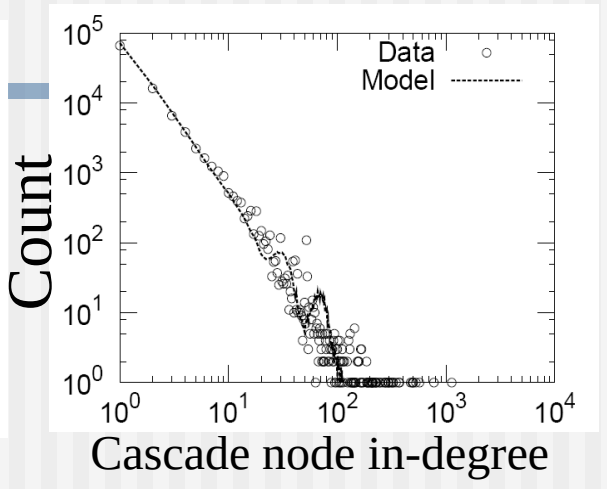
Real:

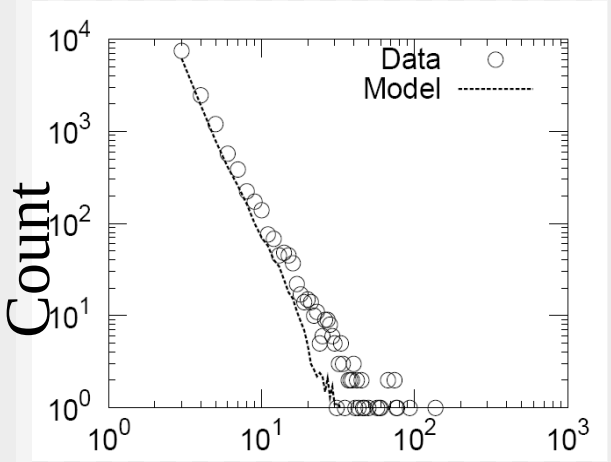


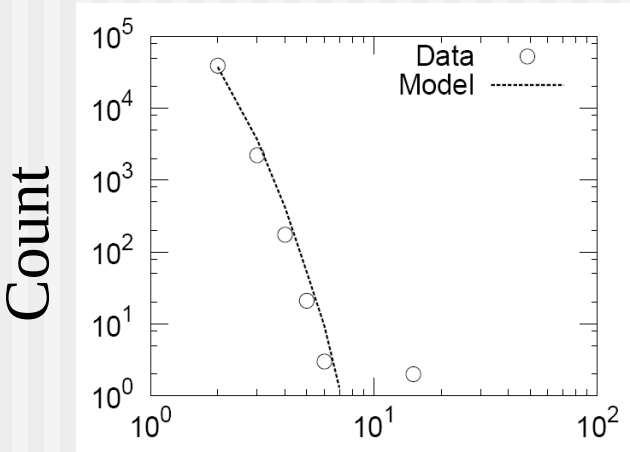
A Generative Model—Validation

- matching cascade size and in-degree distributions (out-degree ≡ 1)
- Generally good agreement









Conclusions

- *Temporal Properties*
  - Popularity drop-off follows power-law distribution exactly as found in other work about human response times.
  - Posts follow weekly periodicity.
- *Topological Properties*
  - Power law distributions in almost every topological property. Star cascades are more common than chains, and size of cascades follow a power law.
- *Generative Model*
  - Developed a generative model based on SIS model in epidemiology that matched properties of cascades.

Final Notes

- Different applications
  - Classic Dijkstra's algorithm
  - Using sliding and virtualization
- Other optimization problems
  - E.g., transmission delay
- Other solutions
  - E.g., min-hop by iteratively increased hop count and max-bandwidth by applying Kruskal's solution on $G(B \geq i)$
- Open problems
  - Problem complexity
  - Optimal solutions

Indoor Environments

- Three popular technologies
  - Wireless LANs (IEEE 802.11 standard)
  - HomeRF (http://www.homerf.org/tech/, Negus et al, IEEE Personal Comm. Feb. 2000)
  - Bluetooth (http://www.bluetooth.com/)

Indoor Environments

- Network topology
  - Straightforward for 802.11WLAN and HomeRF (e.g., In TDMA-based MAC protocol, a central entity is used to assign slots to the stations).
  - The Bluetooth topology poses interesting challenges.

Bluetooth

- Bluetooth Special Interest Group (formed in July 1997 with now 1200 companies).
- Major technology for short-range wireless networks and wireless personal area network.
- An enabling technology for multi-hop ad hoc networks.
- Low cost of Bluetooth chips (about $5 per chip).

Bluetooth

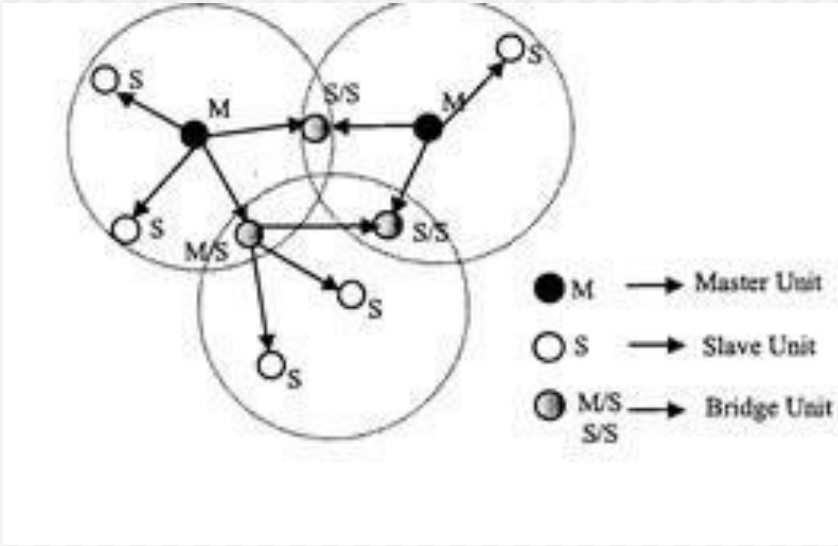
- Basic facts
  - Operates in the unlicensed Industrial-Science-Medical (ISM) band at 2.45 GHz.
  - Adopts frequency-hop transceivers to combat interference and fading.
  - The nominal radio range: 10 meters with a transmit power of 0 dBm.
  - The extended radio range: 100 meters with amplified transmit power of 20 dBm.

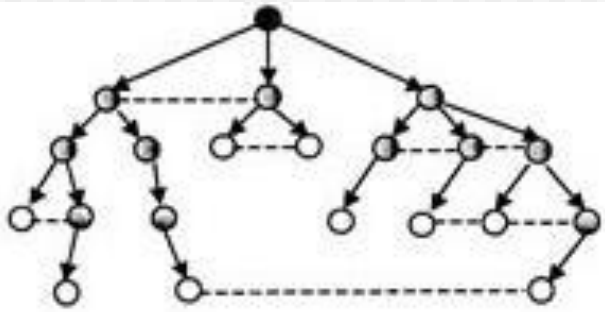
Bluetooth: Basic Structure

- *Piconet*
  - A simple on-hop star-like network
  - A master unit
  - Up to 7 active slave units
  - Unlimited number of passive slave units.
- *Scatternet*
  - A group of connected piconets
  - A unit serves as a *bridge* between the overlapping piconets in proximity.

Bluetooth: Basic Structure

- Open problem: a method for forming an efficient scatternet under a practical networking scenario.
- Two methods: *Bluetree* and *Bluenet*





Bluetooth: Basic Structure

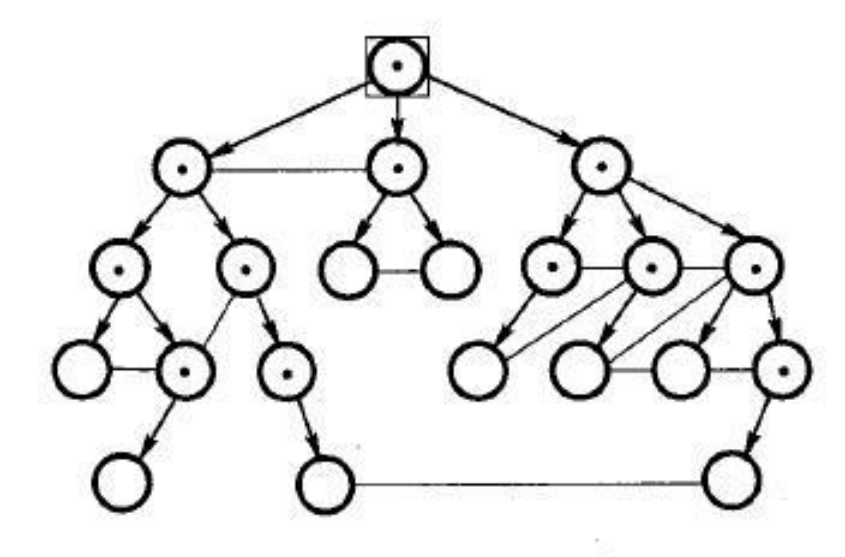
- Scatternet formation
  - Connected scatternet
  - Resilience to disconnections in the network
  - Routing robustness (multiple paths)
  - Limited route length
  - Selection of gateway slaves (a salve being a neighbor of two maters)
  - Small number of roles per node
  - Self-healing (converge to a new scatternet after a topology change)

Bluetree (Zaruba, ICC 2001)

- Blueroot Grown Bluetrees
  - The blueroot starts paging its neighbors one by one.
  - If a paged node is not part of any piconet, it accepts the page (thus becoming the slave of the paging node).
  - Once a node has been assigned the role of slave in a piconet, it initiates paging all its neighbors one by one, and so on.

Bluetree (Zaruba, ICC 2001)

- Blueroot Grown Bluetrees (sample)



Bluetree (Zaruba, ICC 2001)

- Limiting the number of slaves
  - Observations: if a node has more than five neighbors, then there are at least two nodes that are neighbors themselves.
  - The paging number obtains the neighbor set of each neighbor.
- Balanced Bluetree (Dong and Wu, 2003)
  - Using neighbors' neighbor sets.
  - Using neighbor locations.

Bluetree (Zaruba, ICC 2001)

- Distributed Bluetrees
  - Speed up the scatternet formation process by selecting more than one root (phase 1).
  - Then by merging the trees generated by each root (phase 2).

Bluetree (Zaruba, ICC 2001)

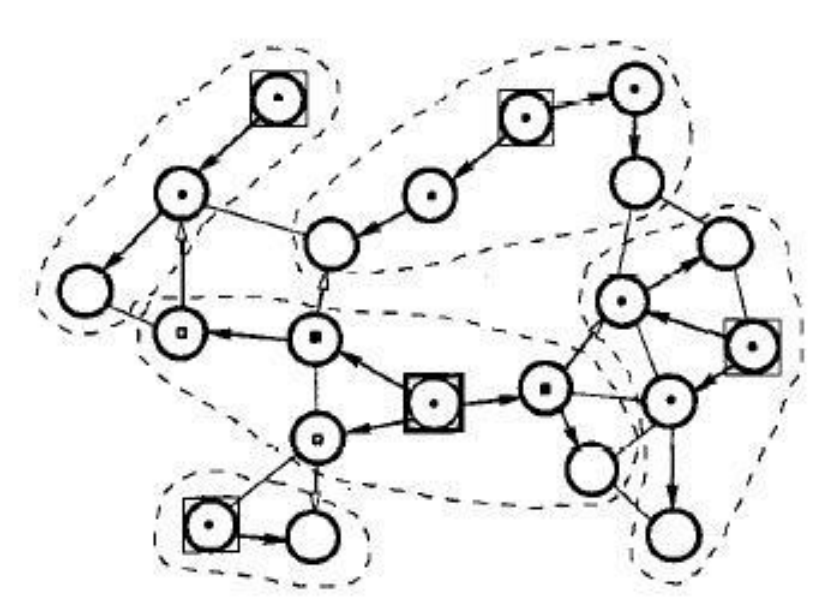
- Phase 1
  - Each slave will be informed about the root of the tree.
  - When paging nodes are in the tree, information of respective roots are exchanged.
  - Each node having roles from the set {M, S, (MS)}, where M for master and S for slave.

Bluetree (Zaruba, ICC 2001)

- Phase 2
  - Merge bluetrees (pairwise)
  - Each node can only receive at most one additional M, S, or MS.
  - Each node having roles from the set: {M, S, (MS), (SS), (MSS)} (note that (MM)=M).

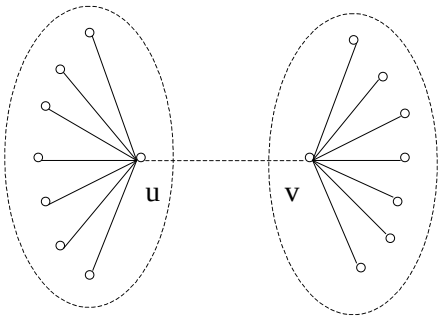
Bluetree (Zaruba, ICC 2001)

- Distributed bluetree (sample)



Bluetree (Zaruba, ICC 2001)

- Overflow problem (Wu)



- Solution: slot reservation (up to 6 slaves)

Bluenet (Wang et al, Hawaii Conf. 2002)

- Drawbacks of bluetrees
  - Lacks of reliability
  - Lacks of efficient routing
  - Parents nodes are likely to become communication “bottleneck”.
- Three types of nods in Bluenet
  - Master (M), Slave (S), Bridge (M/S or S/S)

Bluenet (Wang et al, Hawaii Conf. 2002)

- Rule 1: Avoid forming further piconets inside a piconet.
- Rule 2: For a bridge node, avoid setting up more than one connections to the same piconet.
- Rule 3: Inside a piconet, the master tries to acquire some number of slaves (not too many or too few).

Bluenet (Wang et al, Hawaii Conf. 2002)

- Phase 1: Initial piconets formed with some separate Bluetooth nodes left.
- Phase 2: Separate Bluetooth nodes get connected to initial piconets.
- Phase 3: Piconets get connected to form a scatternet (slaves set up outgoing links).
- Dominating-set-based bluenet?

BlueStars (Petrioli et al, IEEE TR 2003)

- BlueStars (i.e., piconet) formation phase
  - Clustering-based approach for master selection
  - The formation of disjoint piconets
  - Selection of gateway devices to connect multiple piconets
- Yao construction phase
  - Yao procedure is used to ensuring the max number of node degree by removing links without losing connectivity
- BlueStars over the “Yao” topology

NeuRFon (Motorola Research Lab., ICCCN 2002)

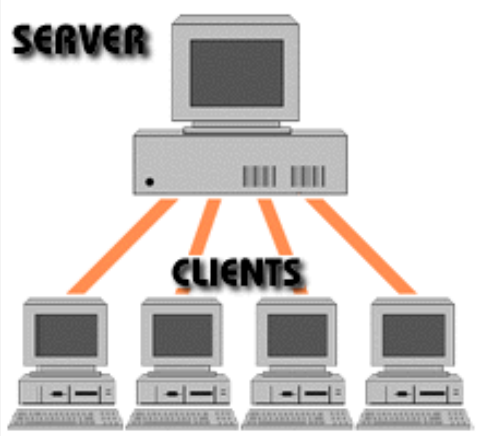
- Build a reverse shortest path tree (w.r.t. a given root) through paging.
- Self-healing: find a new parent with a lowest-level number (cloested to the root).

What are P2P networks?

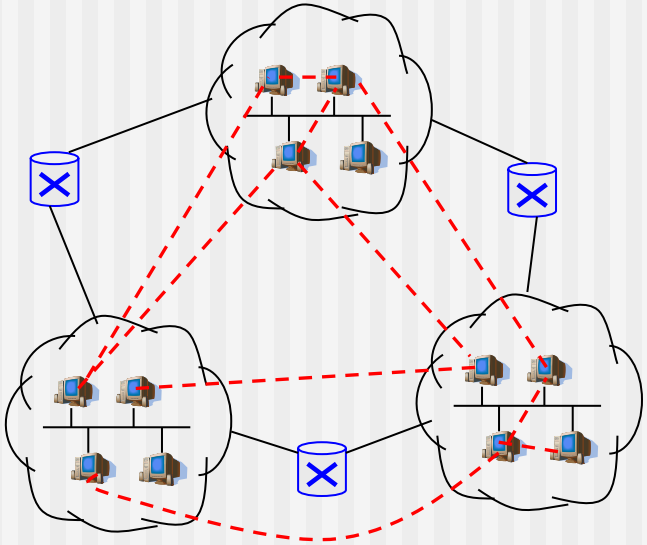
- Definition
  - A distributed system in which peers employ distributed resources to perform a critical function in a decentralized fashion
- Characteristics
  - Peer-to-Peer (P2P): equal node roles
  - Application-level overlay networks
  - Distributed and decentralized
  - Nodes join and leave freely

What are P2P networks?

Peer-to-peer network



Client-server network

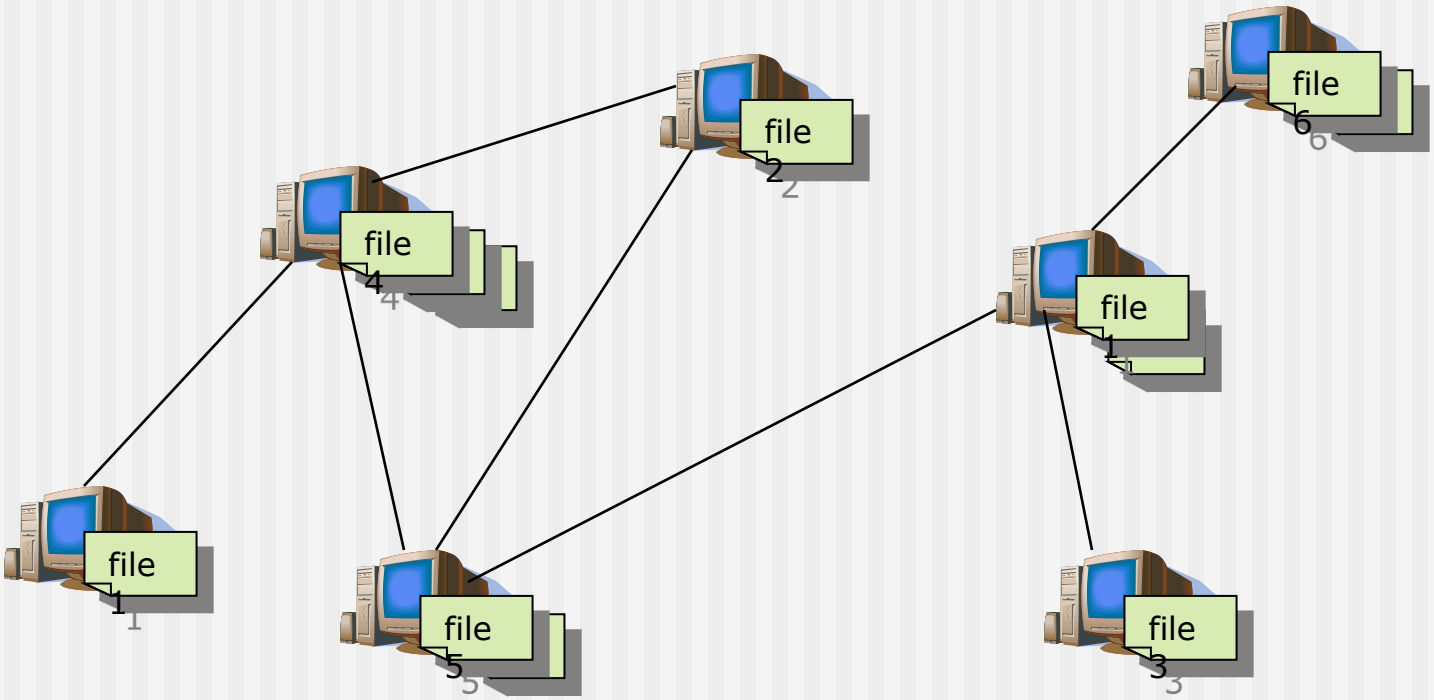


Peer-to-peer network: overlay network

What are P2P networks?

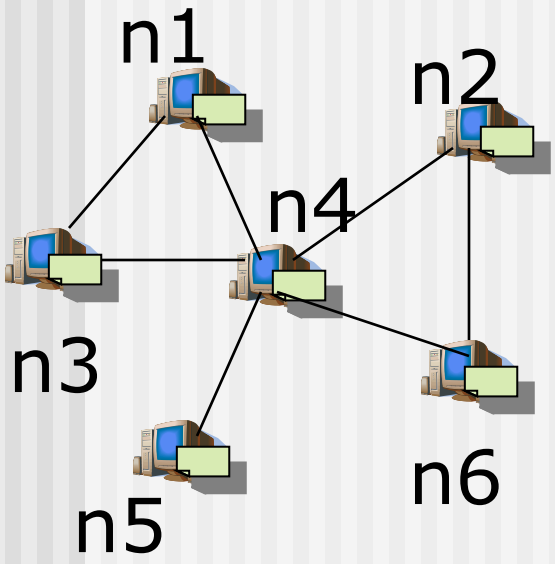
- Benefits of P2P networks
  - No special administration or financial arrangement
  - Can gather and harness computation and storage resources on the edge of the Internet
  - Self-organized and adaptive
- File-sharing P2P networks
  - *Commercial* - Napster, Gnutella, BitTorrent, Kazaa, eMule, iMesh, Morpheus, Freenet, etc.
  - *Research-oriented* - Chord, Pastry, Tapestry, CAN, Symphony, PlanetLab, etc.

What are P2P networks?

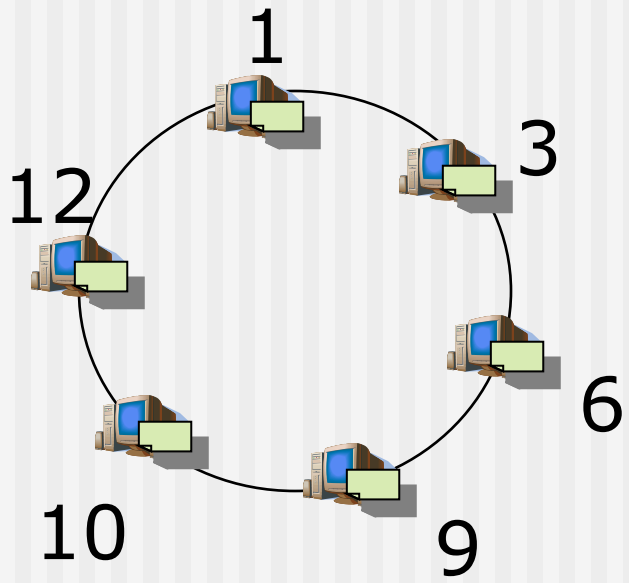


File-sharing peer-to-peer networks

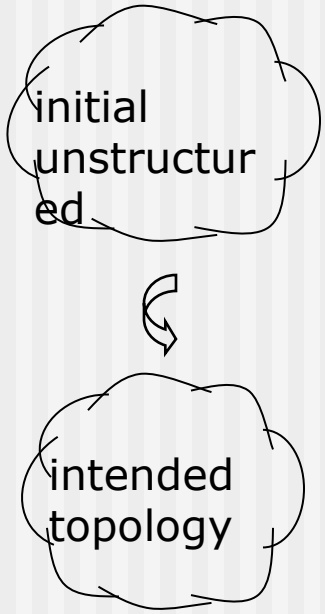
Classification of P2P networks



Unstructured
e.g. Gnutella
(arbitrary)



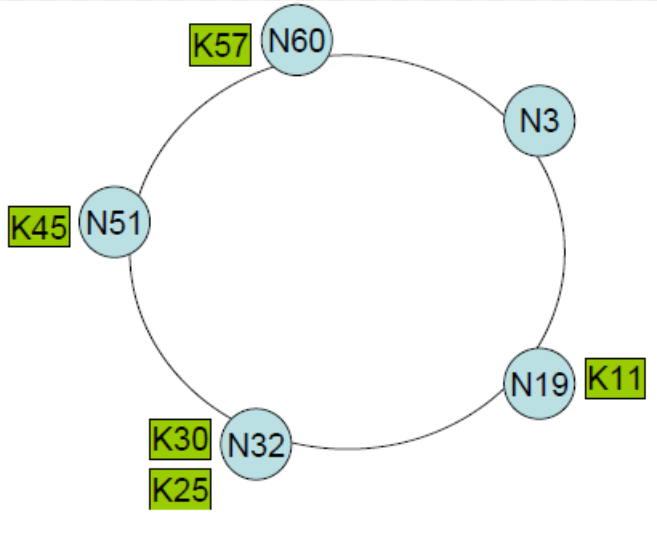
Structured
e.g. Chord
(well defined)



Loosely structured
e.g. Freenet
(based on hints)

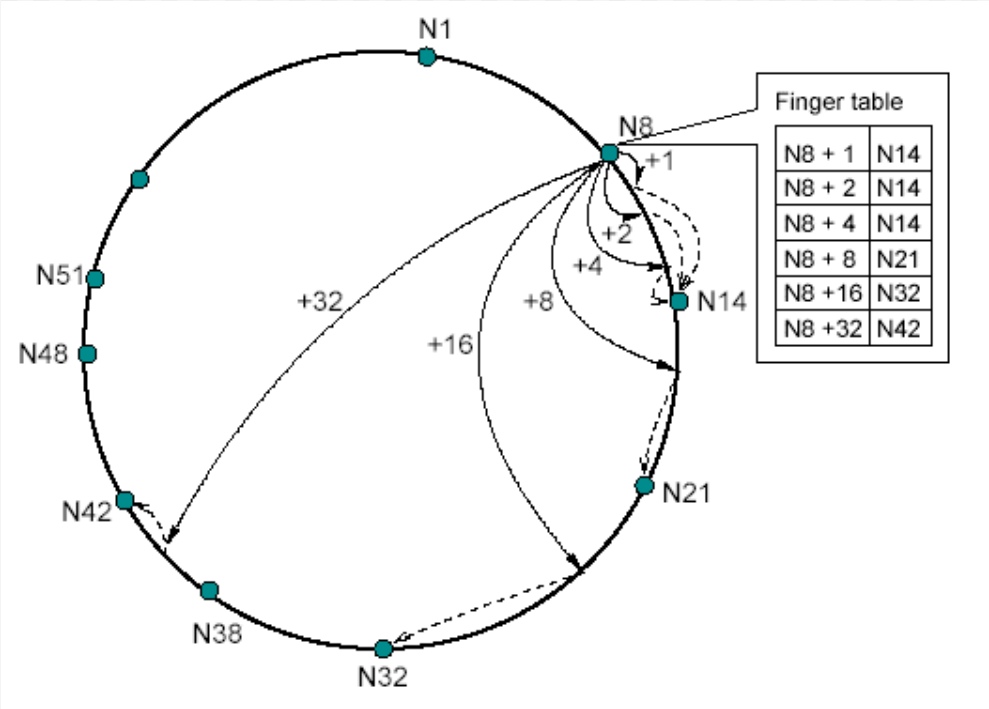
Structured P2P-Chord

- Nodes in a network are organized in a circle
- Each node and each key have assigned identifiers (distributed harsh table: DHT)
  - Node ID: SHA1(IP address)
  - Key ID: SHA1(key itself)
- Each key is assigned to its Successor



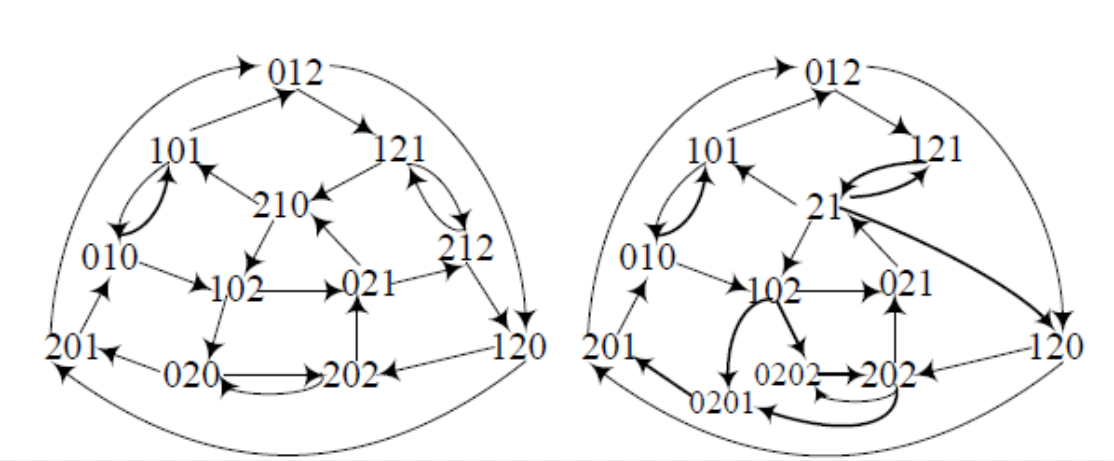
Chord – Finger Table

- The info. Stored in the Finger Table is used for scalable node localization



Other Structured P2P - FISSONE

- D. Li, X. LI, and J. Wu, “Fission E: A Scalable Constant Degree and Low Congestion DHT Scheme”, INFOCOM 2005.



Searching in unstructured P2P networks

- Locating the desired data
- Two types of searching
  - Single file lookup by key or ID
  - Semantic queries
- Searching goal
  - 1 data item
  - As many data items as possible
- Performance metrics
  - Searching quality: query success rate
  - Searching efficiency: # of messages per query

Basic Search Methods

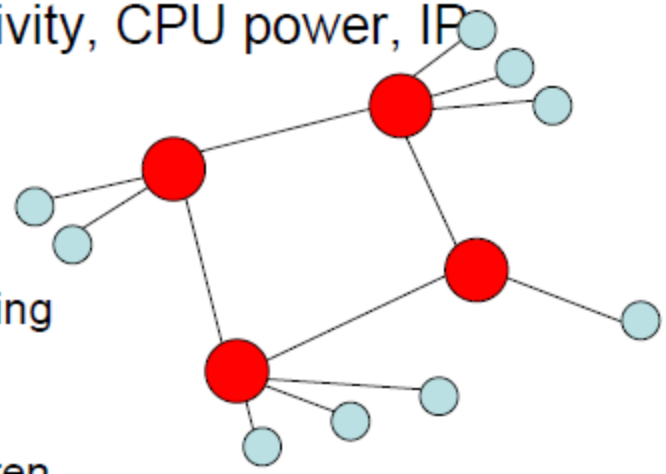
- Basics of P2P protocols (cont'd)
  - **Controlled flooding**: caches (query-id, query-source) to avoid duplicate query processing and uses TTL to prevents a message being forwarded infinitely.
  - **Neighborhood control**: uses the "ping-pong" protocol for maintaining up-to-date neighbors and issues "broadcast-send" to find another neighbor when the current one is lost.

Search in Unstructured P2P

- Sample P2P search protocols (ICDCS 2002)
  - **Iterative deepening**: multiple breadth-first searches with successively large depth limits.
  - **Directed BFS:** sending query messages to just a subset of its neighbors.
  - **Local indices**: each node maintaining an index over the data of all nodes.
  - **Mobile agents**: *swarm intelligence* – the collection of simple ants achieve "intelligent" collective behavior.

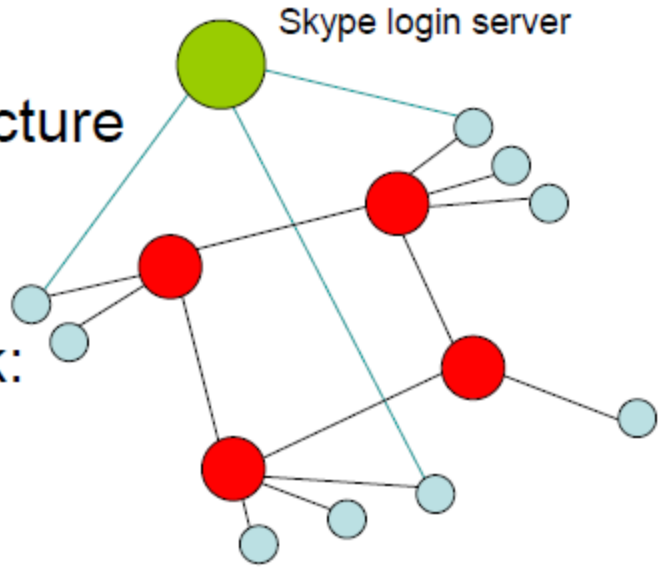
Semi-centralized P2P- KaZaA

- KaZaA is an example of semi-centralized P2P network
- Super-Peers (SPs) are normal peers that have been automatically elected as the super-peers based on their up time, bandwidth, connectivity, CPU power, IP address (public vs private)
- Super-peers maintain a database with:
  - file identifiers, their children are sharing
  - metadata (file name, file size, contentHash, file descriptors)
  - corresponding IP addresses of children
- SP maintain long-lived TCP connections with other SPs



Skype Architecture

- Skype has a similar architecture as its predecessor KaZaA
- There are three types of nodes in the Skype network:
  - Ordinary-peers
  - Super-peers
  - Central login server
- The login server stores all of user names and passwords and ensures that names are unique across the Skype name space



Social Networks

- Social network analysis
  - Views social relationships in terms of network theory consisting of *nodes* and *ties*.
  - Nodes are the individual actors within the networks, and ties are the relationships between the actors.
- Concepts
  - Degree centrality: the number of links.
  - Closeness centrality: mean geodesic distance (shortest path). Between a node and all other nodes.
  - Betweenness centrality: the extend to which a node lie on the paths linking other nodes.
  - 1-hop ego betweenness: how much a node connects nodes that are themselves not directly connected.

Network Science (NS)

A brief history

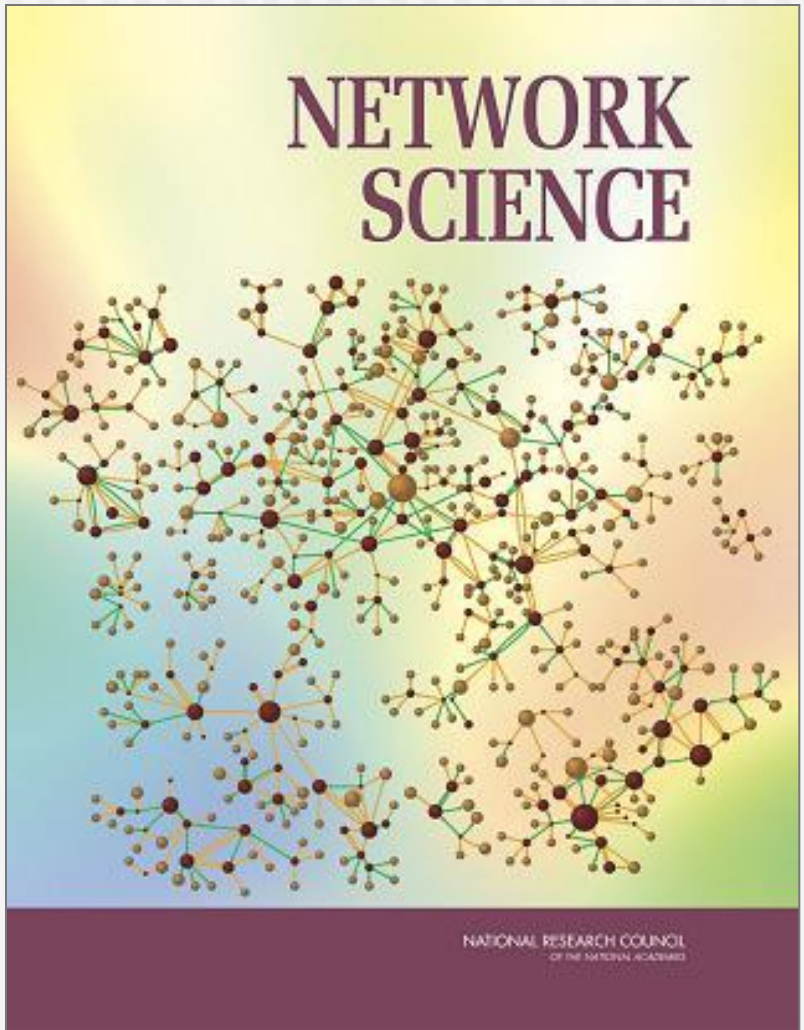
- Graph theory (Euler) and prob. theory (Erdos): random graph
- Social networks: exponential random graph, small-world
- DOD initiative: Network Science (2005)
- NSF NetSE program (2008)

NS: the study of network representations of physical, biological, and social phenomena leading to predictive models

Scope: technological (electronic data), natural (biological, cognitive), and social (social networks)

DoD Network Science Report

- Society depends on a diversity of complex networks
- Global communication and transportation networks
  - provide advanced technological implementations, however
  - behavior under stress still cannot be predicted reliably
- Biological and social networks
  - We do not fully understand these networks, nor the manner with which they operate



On-going projects

- Sensor nodes
  - Smart dust (http://robotics.eecs.berkeley.edu/~pister/SmartDust)
    - Autonomous sensing and communication in a cubic millimeter
    - Macro motes: 20 meter comm. range, one week lifetime in continuous op. and 2 years with 1% duty cycling.

On-going projects

- Sensor nodes
  - Smart dust (http://robotics.eecs.berkeley.edu/~pister/SmartDust)
    - Autonomous sensing and communication in a cubic millimeter
    - Macro motes: 20 meter comm. range, one week lifetime in continuous op. and 2 years with 1% duty cycling.
  - PicoRadio (http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/PN3/)

On-going projects

- Power-Aware Ad Hoc and Sensor Networks
  - μAMPS (μ-Adaptive Multi-domain Power aware Sensors) (http://www-mtl.mitedu/research/icsystems/uamps)
    - Innovative energy-optimized solution at all levels of the system hiearchy
  - PACMAN (http://pacman.usc.edu)

On-going projects

- Sensor Networks
  - WINS (Wireless Integrated Network Sensors) (http://www.janet.ucla/WINS)
    - Distributed network and internet access to sensors, controls, and processors that are deeply embedded in equipment.
  - SensoNet (http://www.ece.gatech.edu/research/labs/bwn)

On-going projects

- Distributed Algorithms
  - SCADDS (Scalable Coordination Architectures for Deeply Distributed Systems) (http://www.isi.edu/scadds)
    - Directed diffusion, adaptive fidelity, localization, time synchronization, self-configuration, and sensor-MAC

On-going projects

- Power conservation algorithms
  - Span (Chen et al, MIT).
  - PAMAS (Power Aware Multi Access protocol with Signaling for Ad Hoc Net works) (Singh, SIGCOMM, 1999).

On-going projects

- Distributed query processing
  - COUGAR device database project (http://www.cs.cornell.edu/database/cougar/index.htm)
  - Database (http://cs.rutgers.edu/dataman/)

On-going projects

- Security for Sensor Networks
  - SPINS (Security Protocols for Sensor Networks) (http://www.ece.cmu.edu/~adrian/project.html)