Delay Tolerant Networks

Network's scalability (Quantitatively)

It has been shown [Gupta, Kumar 2000] that in mobile ad hoc networks $\Theta(N)$ successful transmissions can be scheduled simultaneously. I.E., the *network rate* is $\Theta(N)$ → log R / log N → 1 . Thus, to be regarded as scalable with respect to network size

$$\Psi_N \leq 1$$

For example, for the class of networks under study (assumptions a.1-a.8) (resulting from applying power control techniques)

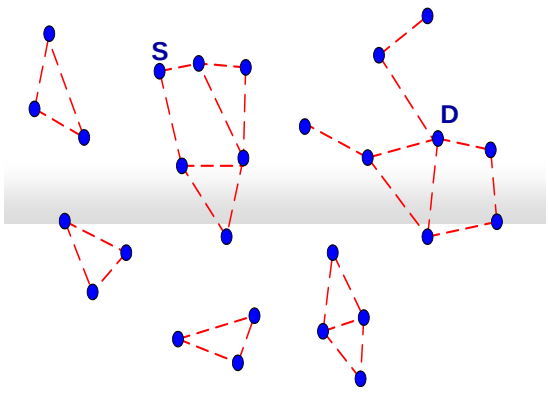
$$\Psi_N = 1.5$$

$\Rightarrow$ **networks under study are not *scalable* w.r.t. to network size**

Network's scalability (Quantitatively)

Note: It has been shown in [Groossglauser&TSE, Infocom 2001] that if the network applications can support infinitely long delays and the mobility pattern is completely random, then the average path length may be reduced to 2 ($\Theta(1)$) regardless of network size and, as a consequence, that network *scalability factor* with respect to network size Ψ_N is equal to 1. Thus, those ad hoc networks (random mobility and capable of accepting infinitely long delays) are the only class of ad hoc networks that are scalable with respect to network size. This work does not consider that class of networks.

What is different?



- A wireless network

that is very sparse and

partitioned

  - disconnected clusters of nodes
- Nodes are (highly) mobile making the clusters change often over time
- No contemporaneous end-to-end path

Reactive/Proactive approaches would not work

- Reactive Protocols
  - route request cannot reach destination
  - path breaks right after or even while being discovered
- Proactive Protocols
  - will fail to converge
  - flood with topology-update packets

A possible approach

- Exploit node mobility to deliver messages

  (Tse et al. exploit mobility to increase capacity)

- A snapshot of connectivity graph is always disconnected.

  Idea: If we overlap many snapshots over time, an end-to-end path will be formed eventually!

- Store-and-forward model of routing:
  1. a node stores a message until an appropriate communication opportunity arises
  2. a series of independent forwarding decisions {time + next hop} that will eventually bring the packet to its destination

Choosing A Next Hop

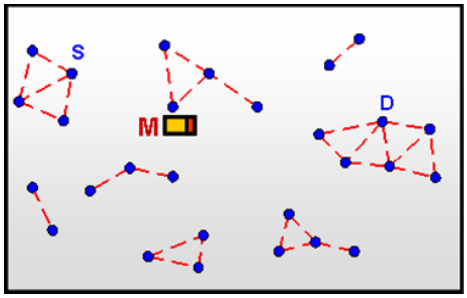
A local and intuitive criterion: A forwarding step is efficient if it reduces the expected distance from destination

usually: reduction of expected distance => reduction of expected hitting time

Efficient Routing : *Ensure that each forwarding step on the average reduces distance to or hitting time with destination*

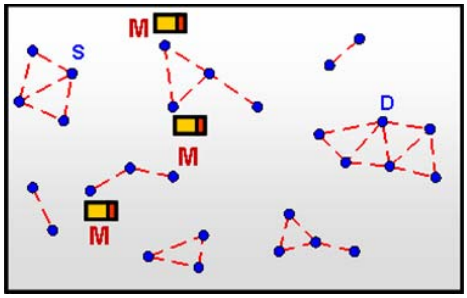
Single/Multiple Copy

- "Single-Copy": only a single copy of each message exists in the network at any time
- "Multiple-Copy": multiple copies of a message may exist concurrently in the network



Single Copy

+ lower number of transmissions
+ lower contention for shared resources



Multiple Copy

+ lower delivery delay
+ higher robustness

Epidemic Routing - The Way

- Each host maintains a buffer of messages unsent (originated or relayed by it)
- Every time it encounters another host, it exchanges info about the pending packets and after the receiver's reply, it forwards to the latter whatever has not been sent yet
- This process is repeated at every meeting

Epidemic Routing II – So?

- Expect 100% delivery ratio even under the worst delay case
- In case of no interference (low traffic load, source-destination pairs), it provides the lowest delay
- **But!** Too much overhead, as it floods the network with copies

Other Variations

- Direct Transmission
  - Forward message only to its destination (simplest strategy)
  - Its expected delay is an upper bound for every other protocol
- Two-hop Relay
  - Deliver messages to the first n nodes it encounters
  - The relays will deliver the message only to the destination
- Tree-based Flooding
  - Similar to the above (the task of making copies is distributed to other nodes as well, not only the source node)

Randomized Routing

- Node A forwards message to node B with probability p
  - P(B closer to destination D than A) = P(A closer to D than B)

yet, because transmission speed is faster than the speed of movement it can be shown that

The randomized policy results in a reduction of the expected hitting time to destination at every step

Message Ferrying

- Exploit existing or introduce non-randomness in node mobility, to enhance communication in an environment with intermittent or otherwise no connectivity
- Network devices are classified as regular nodes and message ferries (could be devices that move independently of the need for data delivery – like buses, or specific nodes that move according to the need for connectivity)

Utility-Based Routing

- Destination's location (relative to another node's location) gets indirectly logged in a timer upon encounter
- Location info gets diffused through mobility process
- Define an appropriate utility function $U_X(Y)$ based on timer value $T_X(Y)$
  - e.g. $U_X(Y)$ = - expected hitting time given timer value
- Utility-based routing:

  Node A forwards a message for node D to node B *iff* $U_A(D) < U_B(D)$

Hybrid Methods…

- Seek phase: If utility around node is low, perform randomized forwarding to quickly search nearby nodes
- Focus phase: When a high utility node (i.e. above a threshold) is discovered, switch to utility-based forwarding
  - look for a *good lead* to the destination and follow it

Oracle-Based, Optimal…

- Assume all nodes trajectories (future movements) are known
- Then, *the algorithm picks the sequence of forwarding decisions that minimizes delay*
- Note that flooding (multi-copy strategy) has the same delay as this algorithm when there is no contention

Shortcomings

- Flooding
  - too many transmissions (energy-efficiency concerns)
  - unbounded number of copies per message (scalability issues)
  - under high traffic, high contention for buffer space and bandwidth results in poor performance
- Utility-based
  - high U_{th}: significant delay increase; source takes a very long time until it finds a good next hop (slow start)
  - low U_{th}: degenerates to flooding

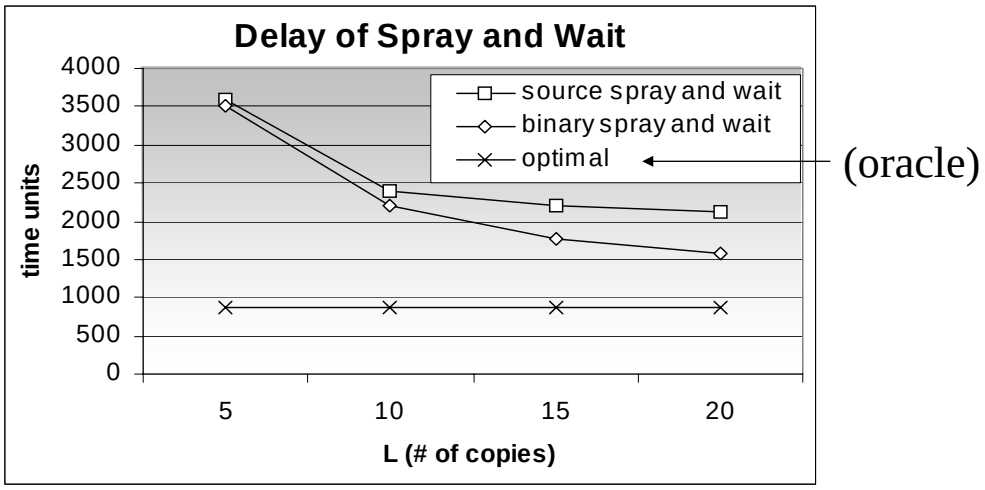
Efficient Routing: Design Goals

- Performance goals:
  - ✓ perform significantly fewer transmissions than flooding-based schemes under all conditions
  - ✓ better delay than existing single and multi-copy schemes; close to optimal
- Additional goals:
  - ✓ scalability: good performance under a wide range of values for various parameters (e.g. number of nodes)
  - ✓ simplicity: require little knowledge about the network

Spray and Wait

- Source Spray and Wait
  - Source starts with L copies
  - whenever it encounters a new node, it hands one of the L copies
  - this is the slowest among all (opportunistic) spraying schemes
- Optimal Spray and Wait (Binary)
  - source starts with L copies
  - whenever a node with n > 1 copies finds a new node, it hands half of the copies that it carries
  - optimal spreads the L copies faster than any other spraying schemes

Spraying Matters!

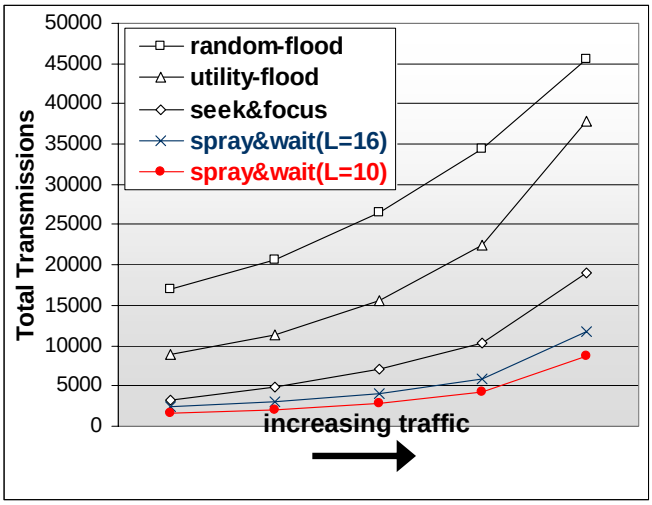


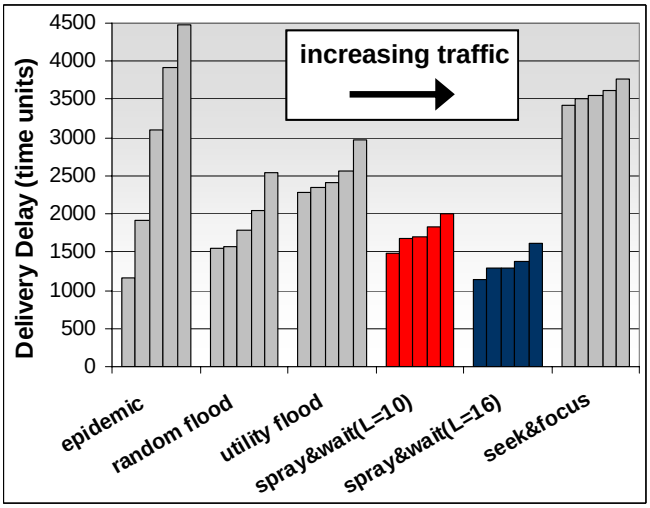
100x100 network with 100 nodes

1. Efficient spraying becomes more important for large L
2. Few copies suffice to achieve a delay only 2x the optimal!

Scenario A: Effect of Traffic Load

(500x500 grid, M = 100 nodes, Tx Range = 10)





	Transmissions	Delay
Low traffic	>10x epidemic 3-4x other multi-copy	same as epidemic 1.4-2.2x other schemes
High traffic	1.8-3.3x	same as above