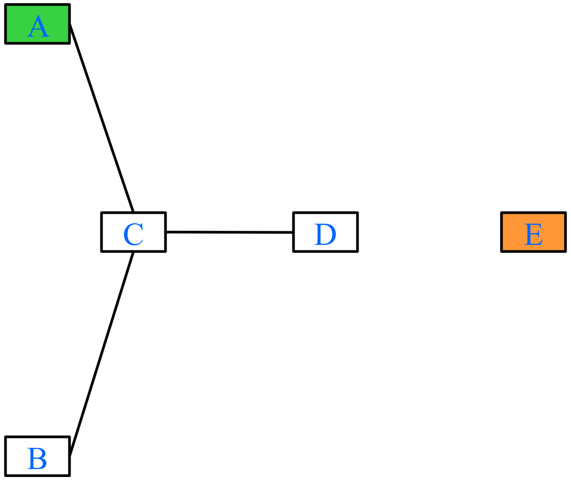
COMP9332 Network Routing and Switching

www.cse.unsw.edu.au/~cs9332

Routing for Delay Tolerant Networks (DTNs)

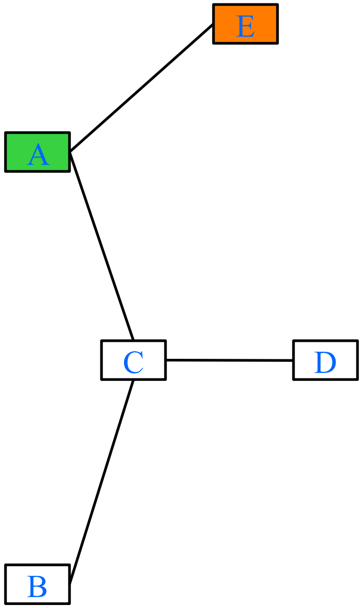
Connectivity at time T0

- Can A send a packet to E using traditional routing protocols?
- No, because no path exists between A and E



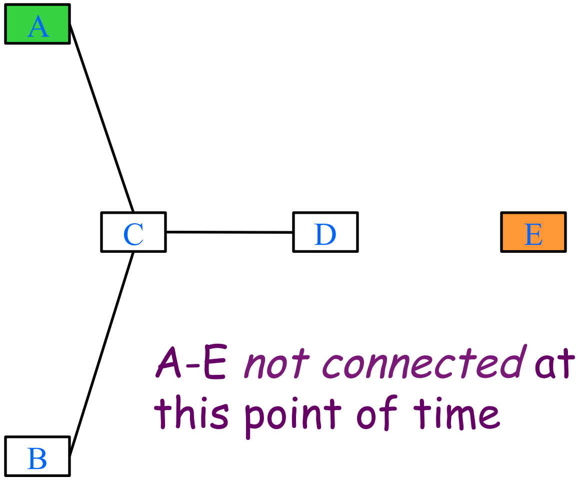
Connectivity at time T1

- Can A send a packet to E?
- Yes, because A is directly connected to E

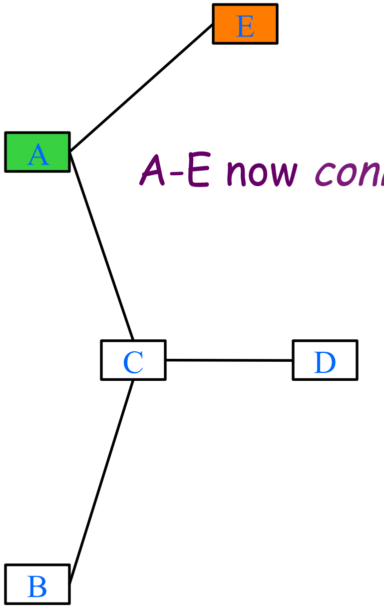


What did we observe?

- Time-varying connectivity
- Just because no path exists NOW, does not mean there will be no path in the near future
- This is the motivation for DTN routing



A-E *not connected* at this point of time



A-E now *connected*

DTN Routing Strategies

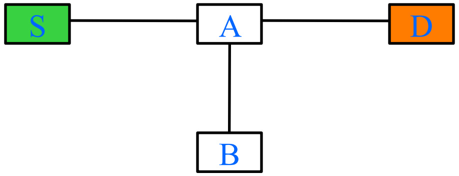
- Direct contact (Single-hop delivery)
- 2-hop relay
- Tree-based flooding
- Epidemic routing

Direct Contact

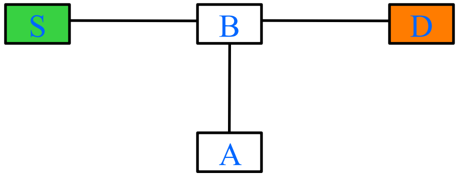
- *Source* must come in contact with *destination* (single-hop delivery)
- Simplest, smallest cost
- Long delay
- Delivery probability = probability of source coming into contact with destination (*p*)

Example of Direct Contact Delivery

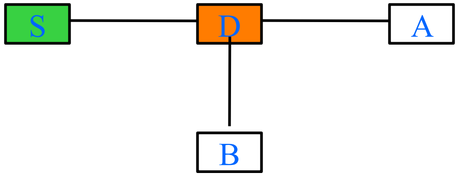
T0: Cannot deliver



T1: Cannot deliver



T2: Can deliver

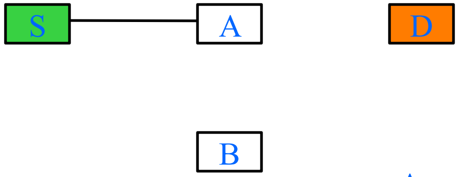


2-hop Relay

- Source copies message to *n* nodes it comes to contact with (none of these are destinations)
- These *n* nodes are called relays (relays are not allowed to copy the message to any other nodes except the destination)
- Source and *n* relays deliver the message to destination if they come in contact with the destination
- Delivery probability = $1 - (1-p)^{n+1}$ (assuming each node contacts destination with an independent probability p)
  - Approx $(n+1)p$ for small p *(delivery prob. increases with n)*

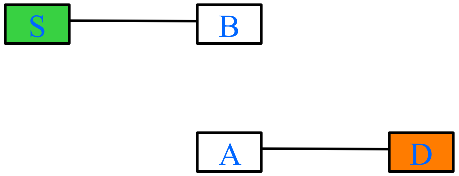
Example of 2-hop Relay (n=1)

T0: S copies to A. Cannot yet deliver because A is not connected to D.



A works as a relay

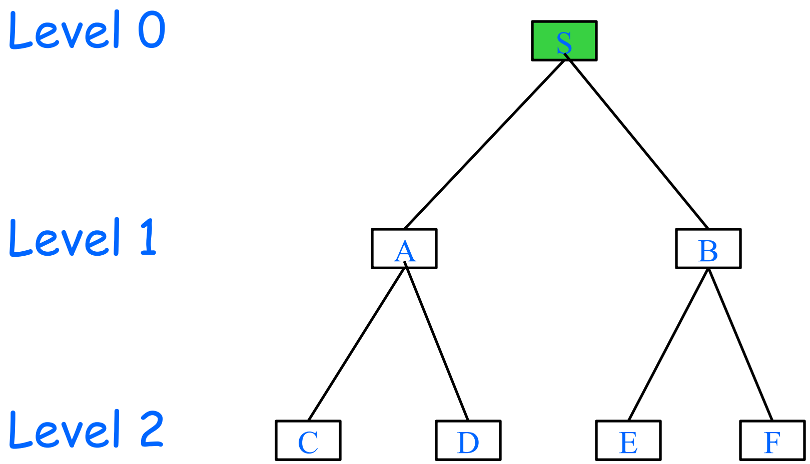
T1: Can deliver now.



Tree-based Flooding

- Extends 2-hop relay to *n*-hop ($n > 2$)
- Relays can copy to other relays (tree depth = n -1)
- Both *breadth* and *depth* can be controlled
- Breadth=m if each node is allowed to make a maximum of m copies
- Max copies = $\sum_{i=0}^{n-1} m^i$

Example of Tree-based Flooding n=3 (depth = 2),m=2



A to F are relays

Epidemic Routing

- Replicate and synchronize when two nodes come in contact
  - Step 1: exchange only the summary vectors
  - Step 2: exchange only the messages that they do not have
  - Outcome: both nodes have identical database (synchronized)
- Eventually all nodes will receive every message!
- High message overhead, but high delivery probability, and low delay

RFC 6693, August 2012

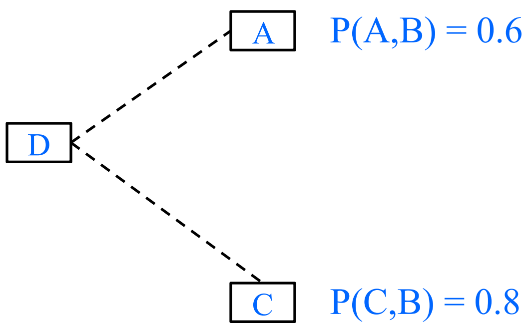
Probabilistic Routing

- Motivation
  - Epidemic routing has too much redundancy
  - Contact probabilities could be learned and used to reduce redundant copies
- RFC 6693 - Probabilistic Routing Protocol for Intermittently Connected Networks
  - released in August 2012
  - Relays are selected based on their probabilities to deliver the message to the destination

Delivery Predictability Metric

- It is a probability (a fraction between 0 and 1) assigned to each (relay) node
- Same node has different probabilities for different destinations
  - P(A,B) is the delivery predictability for A delivering a message to B
  - P(A,C) is the delivery predictability for A delivering a message to C
- It could be asymmetric
  - P(A,B) could be different than P(B,A)

Reducing Relaying Overhead with Delivery Predictability



- If D has a message to send to B, it would select C as the relay node (message will NOT be copied to A saving overhead)

Delivery Predictability Vector

- Each node maintains a delivery predictability for each destination
- If it does not have any current estimate for a given destination, it is assumed zero
- Delivery predictability to itself is 1
- Example of delivery predictability vector for node A
  - B=0.1, C=0.6, D=0.7 (currently it has estimates for three destinations, B, C, and D)

Delivery Predictability Exchange and Update

- When two nodes come in contact, they first exchange their delivery predictability vectors and then update them for each other
- Nodes that often encounter each other, should have high delivery predictability for each other, and vice versa

Delivery Predictability Update for First Time Encounters

- When meeting for the first time, or after a long time, the delivery predictability is set to 0.5
  - If $P(A,B) < \textit{threshold}$, $P(A,B) = 0.5$
- There is an aging process, which would reduce this value (0.5) to a small value (smaller than the threshold) if not encountered for a long time.

Aging Delivery Predictability

- Last encounter time is saved
- When encountering the same node next time, the time difference is used to age the delivery predictability first before exchanging it with the peer
- Aging formula (P'(A,B) is the old value of P(A,B))
  - $P(A,B) = P'(A,B) \times \gamma^k$
  - Where k is time units passed since last encounter and $0<\gamma<1$

Increasing Delivery Predictability with New Encounters

$$P(A,B) = P'(A,B) + r[1 - \delta - P'(A,B)]$$

- $0<r<1$ is a scaling rate for increase
- Delta provides an upper bound for P(A,B) and is recommended a small value δ=0.01

- Example (assume r=0.3 and (1-δ)=0.99)
- Encounter 1: P(A,B)=0.5
- Encounter 2: P(A,B) = 0.5 + 0.3(0.99-0.5) = 0.5+0.147 = 0.647
- Encounter 3: P(A,B) = 0.647 + 0.3(0.99-0.647) = 0.647+0.1029 = 0.7499

Use of Transitivity for Updating Delivery Predictability

- If A frequently encounters B, and B frequently encounters C, then B is a good choice for A to forward a message destined for C
- Therefore, when A meets B, A should also update P(A,C) based on what B says about P(B,C)

$$P(A,C) = \max[P'(A,C), P(A,B)P(B,C)\beta]$$

0<=β<=1 is a scaling factor to control the impact of transitivity on delivery probability

Forwarding Strategy

- Each nodes carries a number of messages for different destinations
- When A meets B, which messages it should forward to B
- A forwarding strategy is needed to make the forwarding decisions
- We shall examine three different forwarding strategies (taken from RFC 6693)
  - Assume that A meets B and A is making the forwarding decision

Forwarding Strategy 1

- Forward the message only if P(B,D) > P(A,D)
  - D is the destination of the message
- Note that this decision will be made for any node A encounters with
- Example: A encounters B, C, and F, where P(A,D)=0.5, P(B,D)=0.6, P(C,D)=0.4, and P(F,D)=0.45
- Question: A will forward the message to which node(s)?

Forwarding Strategy 2

- Forward the message only if $P(B,D) > P(A,D)$ AND $R<R_{max}$
  - R is the number of relays A has copied the message to so far
  - R_{max} is the maximum relays allowed
- Example: A encounters B, C, F, and G in order, where $P(A,D)=0.5$, $P(B,D)=0.6$, $P(C,D)=0.65$, $P(F,D)=0.52$, and $P(G,D)=0.45$ [$R_{max}=2$]
- Question: A will forward the message to which node(s)?

Forwarding Strategy 3

- Forward the message only if P(B,D) > P(A,D) OR P(B,D) > threshold
- This is similar to strategy 1 except the message is forwarded epidemically among nodes with very high delivery predictability (higher than a threshold)
- Example: A encounters B, C, and F, where P(A,D)=0.5, P(B,D)=0.6, P(C,D)=0.4, and P(F,D)=0.45 [threshold=0.43]
- Question: A will forward the message to which node(s)?

Queuing Policy

- A node may have to drop a message due to storage limitation
- Which message to drop?
- We need a queuing policy
- We shall examine two different queuing policies

First in First out (FIFO)

- The message that was queued first will be dropped
- Simple to implement
- May not be the best strategy in terms of optimising the delivery rates
- Let us look at a slightly more complicated strategy that has a better prospect in improving delivery rate

Evict Most Forwarded First (MOFO)

- The message that has been forwarded (copied to another relay node) the most number of times should be dropped first
- It would improve delivery rate, but it requires the node keep track of the number of times each message has been forwarded (one counter for each message)