PRoPHET DTN Routing

PRoPHET =>
Probabilistic Routing Protocol using
History of Encounters and Transitivity

Introduction and a Little History!

Concept invented by Avri Doria and Anders Lindgren for the SNC (Sámi Network Connectivity) project in 2002.

Designed for DTNs where there is no fixed topology or schedule.

All data forwarding happens at opportunistic encounters between nodes.

Patterns in the mobility are used to improve the use of resources as compared with Epidemic Routing to which it is related.

Developed and implemented during SNC and SNC+1, assisted by Samo Grašič who built the Prophet DTN infrastructure software used in N4C.

Looking to have the latest version of the specification published as an Experimental RFC under the auspices of the IRTF DTN research group.

Epidemic Routing

Simple but resource expensive scheme for getting DTN bundles to *every* encountered node – and hence necessarily to the intended destination – eventually, provided that bundle lifetime is adequate

Algorithm:

Whenever an opportunistic encounter occurs exchange any bundles with the encountered node that the node doesn't already possess

- Result: If all goes well both nodes have the union of the sets of bundles they both had before the encounter

Advantage:

Guaranteed to find the optimum path…

.. because it tries every path

Disadvantages:

- This is all very well if nodes have infinite storage capacity
- Practical encounters may not last long enough to complete the exchange .. And it would be good to prioritize transfers that expedite delivery

PRoPHET improves Epidemic

In a real network involving human activity, where mobile nodes are controlled or carried by humans:

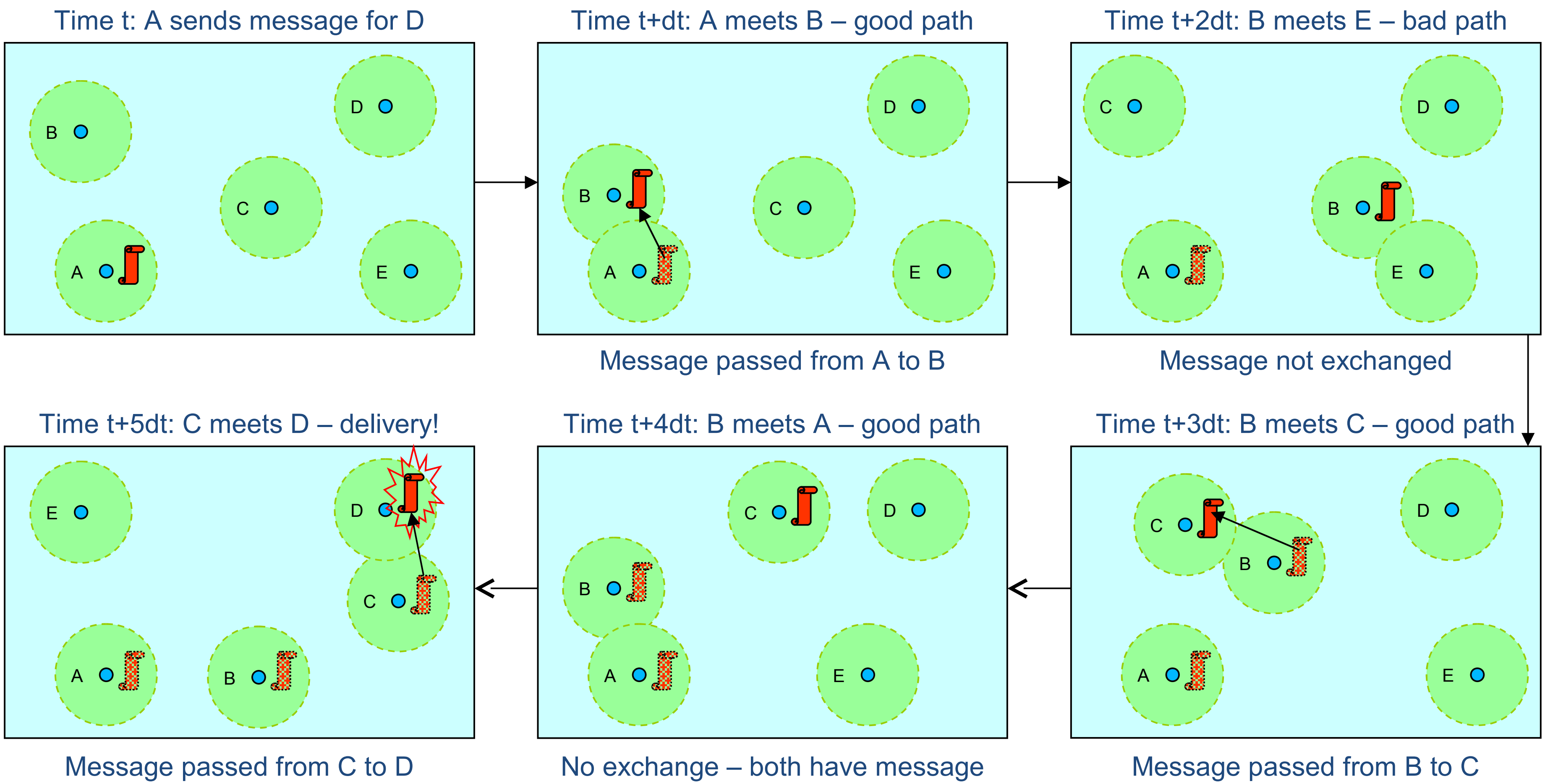
- Mobility is not truly random as in the sense of 'white noise'
- Human activity imposes patterns on the mobility
- The mobility has a characteristic time interval that depends on the sort of activity we are talking about
  - Typically this is much longer than the sort of delays that are normally found in the connected Internet
  - On the order of hours or days in many circumstances such as N4C's test beds

PRoPHET extracts the essence of the mobility pattern of nodes

- abstracts 'History of Encounters' into *delivery predictability* parameter
- parameter is used to prune the Epidemic routing paths
- aim to avoid sending bundles only on paths that have a low probability of reaching the bundle's destination

Delivering a Bundle

This sequence of diagrams shows how a set of Wi-Fi equipped nodes moving in an area that is much bigger than the range of Wi-Fi might deliver a bundle



Node with Wi-Fi range

Message from A to D

Copy of message

Folly Consulting

Modelling the Mobility Pattern

Every node maintains a set of *Delivery Predictabilities (DPs)* for nodes it has encountered (reasonably) recently

- DP for node B stored in node A: $P_A(B)$
- DP before/after encounter: $P_A(B)_{old}$ / $P_A(B)_{new}$

The DPs evolve over time as nodes encounter each other, thus…

Basic DP Evolution Equations in node A when it meets node B

1. Direct encounter: DP for encountered node increases at each encounter
   - $P_A(B)_{new} = 0.5$ [on first encounter – when $P_A(B)_{old}$ is 0]
     - don't know whether there will be more meetings so 'hedge our bets'..
   - $P_A(B)_{new} = P_A(B)_{old} + (1 - P_A(B)_{old}) * P_{encounter}$ [subsequent encounters]
1. Decay over time: All DPs are decreased if the node hasn't been encountered
   - $P_A(B)_{new} = P_A(B)_{old} * \gamma^K$ [K is number of time units since last decay]
     - If $P_A(B)$ gets very small, set it to 0 and treat next encounter (if any) as first
1. Transitive rule: If B is a good path to C and A meets B frequently then nodes that meet A might want to give messages for C to node A
   - $P_A(C)_{new} = \max(P_A(C)_{old}, P_B(C), * P_A(B)_{new} * \beta)$ [β is a constant]
     - Encountered node (B) sends its set of DPs to A for use in the Transitive Rule

The real world is a bit more complicated ($P_{encounter}$ is not a constant)

N4C

PRoPHET Protocol

Manages encounters between pairs of nodes.
  - Effectively a point-to-point protocol

Stage 1: detect a new neighbor
  - Mechanism depends on link layer and separate discovery protocol
  - For example, DTN2 has IP and Bluetooth based mechanisms

Stage 2: execute PRoPHET *Hello protocol* to confirm PRoPHET support
  - Agree first/second roles between connected pair; exchange identities

Stage 3: information exchange – details on next slide
  - On completion both nodes have sent and received some bundles according to relative values of updated DPs in the two nodes

Stage 4: extra bundle exchange
  - If new bundles arrive due to local applications or new encounters determine if they should be sent to connected node(s)

Stage 5: periodically repeat complete information exchange

Neighbor disappears: break off connection – may happen at any stage

PRoPHET Information Exchange

PRoPHET Information exchange stage:

- Both nodes apply decay equation (2) to own DP sets
- Nodes send these sets of DPs to the encountered neighbor
- Nodes apply direct encounter and transitive equations combining its own DPs and the set of DPs received from the encountered neighbor
- In each node, compare updated local DPs with DPs received from encountered neighbor:
  - According to policy and current DPs determine if any bundles held by this node should be offered to the encountered node
- Each node sends offers to other node
  - Node receiving offers decides which bundles to accept according to local policy e.g., will the bundle fit into available storage? is it too old?
  - Having decided which bundles to accept it sends acceptance responses to offering node posibly modifying the order to suit its own policy
- Offering node then sends accepted bundles to the other node in the order requested by the accepting node

PRoPHET in N4C

- The Prophet DTN infrastructure code as developed by Samo Grašič has been extensively used for
  - The LTU web/email/nsim/podcast experiments in Swedish Lapland.
  - The MEIS meterological and radiological monitoring experiments in Kočevje, Slovenia.
- The PRoPHET routing protocol code in the DTN2 reference implementation has been tested and had limited experimentation in conjunction with the PyMail DTN nomadic email system developed by Folly Consulting and deployed to a limited extent in Swedish Lapland.
- A new version of PRoPHET has been researched, designed and specified by Elwyn Davies of Folly Consulting.
- Extensive simulation of the new and older versions of PRoPHET have been carried out by Samo Grašič at LTU to demonstrate some issues with the older version and compare it with the new version.
  - An academic paper is in preparation

Folly Consulting

PRoPHET Resources

Current Internet Draft specifying Version 2 of the PRoPHET protocol:

http://tools.ietf.org/html/draft-irtf-dtnrg-prophet-09

Source code of Prophet DTN Infrastructure software:

http://code.n4c.eu/code/PRoPHET/

Source code and documentation of the PyMail nomadic email system

http://code.n4c.eu/code/PyMail/

Folly Consulting