3/7/2016

Zebranet -- Project Proposals Due Weds

Store & forward / data muling ("zebraing") / delay tolerant networks

Simple scenario -- route is known / unchanging, but frequency of connectivity may be low / unpredictable.

Conventional store and forward design, i.e., mail routing, early email / usenet

How does mail routing work?

(FedEx -- all airmail Memphis, then on to region airports, then by truck...)

How about email?

(DNS to look up mail server for a given domain)

Used to be "bang paths"

MIT!ATT!Berkeley!BerkeleyEECS!joe

MIT!!{ATT,ComCast}!Berkeley...

Even in a network with fixed nodes, there are some quesitons:

Who is responsible for delivering message?
- origin
- "custody transfer"

When to re-transmit/retry? This boils down to how to estimate end-to-end (E2E) latency. (What's the problem with retrying too frequently? We will end up resending packets we've already sent!)

Hop-by-hop acks vs e2e acks?

In any case, don't delete until received

How long do I need to keep around the fact that i've already ack'd a packet?

Now suppose route isn't known -- have to discover. How do I do that? If I want to reach a particular destination, and I know nodes won't move?

"AODV" -- ad-hoc, on-demand, distance vector

See: http://www.antd.nist.gov/wctg/aodv_kernel/aodv_guide.pdf

Basically flood a request asking for path to a given destination

<u>Now suppose nodes can move?</u> Hard because distance vector returned route may be totally invalid by the time we discover it!

Some solutions:

1. Flooding

    Problems: flood storms

    state on nodes (have to keep track of a lot of messages)

    When to retry/retransmit, and whom?

    When can you forget about a message?

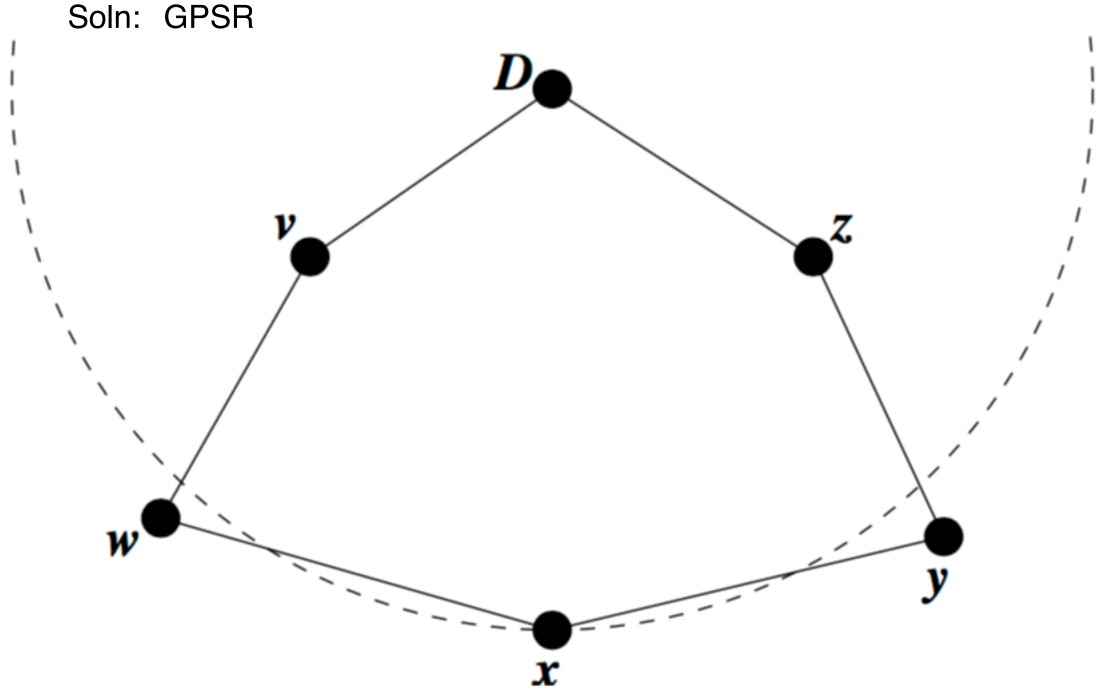
2. Geo-routing

If nodes know their location, and the approximate location of their neighbors, maybe we can use geographic "routes'

Basic idea: look at all your neighbors, send message to one that is closer to the destination than you are

    Problem: "holes"

    Soln: GPSR



(source: Karp & Kung: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, Mobicom 2000)

Basic solution: "right hand rule" -- send to first neighbor counter-clockwise to yourself, relative to the path from yourself to the destination, along with information about your distance to destination.

If someone on path is closer to you, they can exit this "perimeter mode". Ohterwise they forward according to the same rule.

If packet comes back to a node, there is no path.

(This is provably true in "planar" graphs -- what's that?) A graph with no edges that intersect each other -- e.g., a graph embedded in a euclidian coordinate system has this property.

3. Some heuristic, e.g., estimate Pr(connectvity) from every node to basestation as in ZebraNet

<u>Let's discuss ZebraNet:</u>

- Radio architecture: two radios, one 8km, on very short range

- Long range radio implements its own mac

- Slotted TDMA approach for short range radio (describe)

- Long range radio is to deliver data back to basestation

- <u>Do they really need the short range radios?</u> Can't tell -- seems like maybe not. They say the are the zebras are in is 400 km^2. How big is a circular region of this area?

a = pi r ^ 2
r= sqrt(400/3.14) = 11.2 km!

So it seems like we are nearly in a situation of all notes being fully connected to base!

- Short range protocols:

- basic flooding, with no "memory" (though they must have actually not stored duplicate packets)

- history based -- keep a "hierarchy level" that estimates how frequently a node connects to a basestation -- what is the protocol, exactly?

- Level starts at X
- Every time successfully deliver a packet to basestation, level=max(level-1,1)
- Every time we fail to deliver, level = min(level+1,X)

In experiments, they tell us they keep a "delete list" of messages that have been successfully delivered (terrible way to write a paper!)

<u>How did they evaluate this?</u>
(Hint: They didn't put it on real zebras! Bait and switch! Why show us those hardware pictures!)

Simulation -- zebras on a grid, moving according to mobility models in paper

Simulate radios that can transmit a fixed range

Show success graphs :

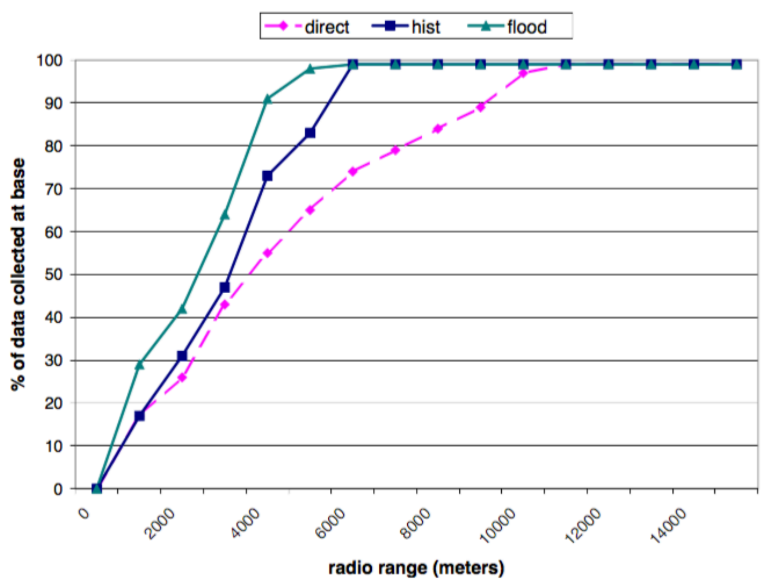


Figure 10: Success rate with infinite storage and bandwidth.

A little hard to take seriously given the simplicity of the simulation.

Ex: How would a different base station movement model have affected things? It traverses a rectangular pattern at a fixed rate. What if it just stayed put, or went to a few locations and waited for longer? Why is the basestation only on for the times it is on? Is there any attempt to synchronize when it is on w/ times zebras are transmitting (since the only transmit 30 mins of every 2 hours, this means if base is only active 3 hours a day, only one window will successfully connect w/ base). Paper makes contradictory statements, i.e.:

> range of the base station. Since the nodes do not know when the base station will be available, base discovery is done from noon till midnight every day. This is typically overlapped with Peer Dis-

and then

> **Simulation Methodology.** Our simulations consist of four communication phases that occur within 30 minutes every two hours, *i.e.* from 12:00-12:30, 2:00-2:30, 4:00-4:30, etc, over an entire month. This timeline

Ex: Would it have been better to use a higher power radio? A bigger antenna on the basestation?

That's all: How are projects? Project partners