6.852, Tina Nolte

1

Motivation points and introduction:

Mobile ad hoc networks are formed by a collection of, potentially mobile, wireless nodes. Communication links form and disappear as nodes come into (link formation) and go out of (link failure) each other's communication range. Highly variable message delays. These are very similar to asynchronous networks with link failures except that links may crop up again in other places.

2

Can be envisioned as a collection of routers which are free to move arbitrarily... come equipped with transmitters. Comm link status could be affected by anything from position, transmission power levels, antenna patterns, and interference. Potentially rapidly changing and unpredictable topology.

3

Applications include anything from home networking, search and rescue, to military applications.

4

To formally model these systems we could envision something similar to our IOA model with the ability of automata to change their signatures and interfaces corresponding to changing network conditions. Modelling for this area is a new subject and there are a great many approaches, including ideas from hybrid systems at very low levels, taking into account node speeds, distance, and a variety of other factors all the way to assuming strong network power where a network could report back those nodes that are currently neighbours.

5

In particular, consider the model where processes query the network layer and receive back a list of neighbours.

6

Research in mobile ad hoc has centered on routing and medium access control protocols. Little work has been done in group services.

7X

Existing distributed algorithms will work on top of the ad hoc routing protocols since they are designed to hide the topology. However, having a core set of algorithms can lead to more efficient and cleaner implementations of solutions. There are many examples where this principle occurs: perhaps a process wants to perform some local area computation... it would be much better off if it evaluated directly who those other processes are that should participate rather than using a routing protocol that may do more work than is needed for the problem.

8

Group communication is an important building block. It can be used as a high level service and allows applications to be oblivious to the details of the dynamic network environment. This leads to the development of layering approaches to accomplishing tasks in distributed networks, where application services are running on top of system services that are running on increasingly primitive system services. It is difficult in this setting since links can repeatedly fail and recover.

9

What are some features of group communication? Maintaining information regarding group membership, letting nodes within a group communicate in an ordered manner.

10

Previous group comm services have yielded several approaches that use a token to implement total order. One is to circulate token in virtual ring and  when a proc receives a token it assigns labels to its messages from the sequences of numbers in the token and multicasts. The sequence number carried is incremented with each message and we can establish total order.

But these need the existence of a virtual ring to operate on. The next couple of minutes concentrate on token circulation on networks with arbitrary topology that change dynamically. These algorithms are short and uncomplicated but serve to illustrate just how new this field is and what kinds of algorithms people are considering. I hope we can approach this as a mock brainstorming session, since "try and see" is the theme of the paper... this is a new area in a very exciting time where we can try very simple solutions to get a feel for what might be the right way to approach a problem.

11

Now for some definitions:

A token visits a node if it was received by the group communication service running on that node.

A token is routed by a node if the node simply forwarded the token to another node. A token may be routed by a node without visiting it.

A round is a minimal length execution sequence in which each node is visited at least once.

Round length is the number of node visits made by the token in one round.

12

Motivation for mobility aware algorithms:

A mobility aware algorithm is as it sounds... it's an algorithm that does not rely on a routing protocol and instead monitors properties of the current network status and makes decisions based on that information. What's a good example of a situation where we'd want a mobility aware algorithm?

13

Take this example... procs are 1 through 6 but are arranged as 1,3,5,2,6,4,1.

14

Now say that we decided to visit the nodes in order numerically. What is the length of a round (6... not more since we are only talking about where the node actually visits)? What's the advantage to visiting the nodes in the order they are arranged (lower message overhead since the hops between nodes visited consecutively shrinks)? While this is a simple example it should serve to illustrate that a mobility aware algorithm can result in significant time and message complexity reduction.

15

In the interest of providing solutions that are not so expensive, we're here going to go over some very simple algorithms that rely on local neighborhood information at most to make its decisions.

16

In order to do this we need to actually know who our neighbours are. Here it is accomplished using "hello" messages that are broadcast at hello intervals. Each node thinks it has a node j as a neighbour if it recently received a hello message from j. On the other hand, if a node does not receive a message for some threshold number of hello intervals then it decides that the node it did not hear from is no longer a neighbour. This kind of service can be implemented as part of a group communication service or it could be something done by the network layer and made available to the group communication protocol.

17

There are now 6 algorithms that we are going to present having some of the following sorts of properties:

18

Local versus global: A local algorithm sends tokens to nodes chosen from the list of the current suspected neighbours while a global algorithm tries to direct the token to any node that is in the network. Particular global algorithms might send tokens directly to their intended destinations and intermediate nodes just relay the token without being visited or the token will be passed along and visit all other nodes on the way to the intended destination.

19

Recency versus frequency: Recency algorithms make decisions on who the next recipient of the token is based on how recently nodes have had the token while frequency based algorithms make the decision based on how frequently the nodes have had the token.

20

The algorithms here also make use of an assumption to protect against potential loss of a token resulting in an inactive system. Since the current token holder might not know exactly who its neighbours are it may choose a node to forward the token to that is no longer a neighbour. In this case, a TCP protocol is assumed, running on top of the routing protocol, which will eventially deliver the token to the recipient unless it has been partitioned away.

21

ALGO Local Frequency

This keeps track of how often a node has been visited and sends the token to the least-frequently visited neighbour of the token holder. The token keeps track of a count variable that indicates the number of token visits to each node.

If there is no mobility and the topology is connected (which we'll be assuming thorughout these analyses) then each node is visited infinitely often. Why? (There must be some non-starved node since somebody has the token... since the graph is connected, there has to be a node that has at least one starved node as a neighbour, meaning that when x gets the token he will pass it to one of his starved neighbours.)

22

There is a rather unfortunate turn of events that the algorithm can take... it can, on certain graphs, start to develop unbounded round lengths. Take for example the following graph where ties are broken according to the smaller identifier.

Somebody trace through the execution on the graph:

1 to 2, 3 to 2, 2 to 1,3,4,5, 4 to 2, 5, 5 to 2,4

The sequence winds up being 1,2,3,2,4,5,4,5,2,1,2,3,2,1,2,3... You have to repeat nodes that are isolated, resulting in 2 having a large count and so procs on the other side will then pass back and forth, etc.

23

However, the algorithm also experiences dramatic changes to round length depending on small changes in topology. Consider adding an edge between 1 and 3 and the round length stabilizes to 5. Then the execution is 1,2,3,1,2,4,5 and stabilizes to 4,5,2,3,1 to 3,1,2,4,5 to 4,5,2,1,3.

Consider what this means for mobile networks... links may be disappearing and being added very frequently, meaning this algorithm will suffer from a great deal of instability.

24

ALGO Local Recency

This keeps track of how recently a node has been visited and sends the token to the least recently visited neighbour of the token-holder. The token keeps track for each node the last time it was visited. Each visit the time value that the token has is updated by 1.

25

Does this algo have the same problem as LF does on the previous graph? (No... it ensures round length no more than 7 where we have 1,2,3,2,4,5,2,repeat). In any static connected graph, a round length of 2n can be achieved since nodes can be visited according to the spanning tree of the graph. This algorithm tries to do better by taking advantage of cycles to cut back on backtracking. However, there are recursively constructed graphs that result in round lengths exponential in n.

26

ALGO Global Frequency

The token is sent to the node that has been visited least frequently among all nodes in the system. Ties may occur often in this algorithm so a tiebreaker procedure needs to be used. What is the round length in this algorithm (number of nodes in the network)? Do nodes have to be visited in the same order each round (No... say A, B, C fully connected... may visit A B C B A C A B C)?

27

ALGO Global Recency

The token is sent to the node that has been visited least recently among all nodes in the system. When can ties occur (First round only since it is the only time that there is no process that is most recently visited)? What is the round length(number of nodes)? Do nodes have to be visited in the same order each round (Yes... once they have been visited they all have distinct last visited timestamps which will determine exactly the order they are visited in the future)?

28

ALGO Global with next

The algorithm first determines the node with the smallest number of times visited or first last visited time. Then the token is sent to the neighbour of the token holder on the route to the node with whichever of the two properties above was being tested for. These obviously require that some outside information be available to determine who the neighbour is on the path to the final destination.

29

Using the smallest number of times visited metric turns out to not be such a great idea. Why is that(since intermediate nodes are visited then their frequency becomes much higher than the rest and the other nodes are visited many times before returning to x again, giving a very large round length)?

30

Simulation results

In static topologies, the global algorhtms had the best round length by definition though LR was a good performer... the global algorithms had to suffer from potentially larger algorithmic overhead.

Least Recently was the best for dynamic networks, which is what we were interested in. There are algorithms that do better than Least Recently in the static case that are covered in the paper, but that is not surprising... it is intuitive that algorithms designed to work well in static cases might ignore issues that commonly arise in dynamic ones.

LF continued to demonstrate the stability problems that we examined earlier.

For simulations in dynamic network settings, the hello intervals and the speeds of the agents were varied. As might be expected a larger hello interval saved on hello messages but introduced larger algorithm costs.

31

Questions

The claim in the paper is that LR seems to be the best algorithm but we know that there are cases where it seem to break down... how do we characterize those graphs that result in nice executions?

32

The paper only analyzes these algorithms in the static connected case. What would an analysis look like in a more interesting case? What sorts of things would one want to discuss in those more dynamic cases? For example, one might want to be able to make statements about things like activity within partitions of a network or perhaps we could attempt to make a claim about the behaviour of an algorithm based on some "maximum network change" metric.

33

This is obviously a somewhat new area as these algorithms are all simplistic in the extreme. They don't tolerate token loss or long term partitions of a network... two very realistic situations that should be handled.