UNPLUGGED

Network Routing Simulation Activity

Background

Computer Networks

Computer networks allow computers, located in different physical locations, to communicate with one another. This is similar to how telephone networks allow communication between telephones (and their users) that are in different locations. The *Internet* is a special type of network that allows computers all over the world to communicate. The Internet is a network of networks.

Networks consist of *hosts* and *links*. Hosts are computers that are part of a network. Links connect hosts together, allowing them to communicate with one another. Some links use a physical medium, like fiber optic, ethernet, telephone, and coaxial cable. Other link types send information through the air using radio waves or satellite signals.

Routing Messages in Networks

A *router* is a special type of host that is responsible for making sure that data sent on the network makes it to its destination. A router keeps a table called a *Routing Table* that tells it where to send packets in order to get them to their destination. Messages sent on bigger networks, especially on the Internet, often go through a number of routers on their way to their destination. Routers forward packets using message destinations to determine where to send the messages.

Gateway routers are routers that sit on the border of two or more networks and facilitate the passing of messages between the networks.

Message Standards and the Internet Protocol

Message addresses follow standard formats to allow computers on the network to communicate and work together to route messages. The Internet uses the *Internet Protocol (I.P.)* for addressing messages. I.P. addresses consist of 4 numbers in the range 0 –255 (i.e. one byte) that are written separated by decimals.

i.e. *192.168.1.1*

I.P. addresses allow the routers in the Internet to communicate with one another and figure out where to send packets.

Messages are routed based on each number in the I.P. address, from left to right. As the numbers are processed, each gives more and more specific information about where the computer with that I.P. address is located. This is similar to how postal addresses consist of a country, province (or region), city, street and street number.

e.g. *123 Apple Lane, Appleton, HS, 1H2 S3A, A Country*

A postal address gives more specific location information as one reads it backwards in chunks. That is, from country to postal code to city to street to number.

Unlike postal addresses, I.P. addresses are not necessarily based on geographic locations. Partially matching addresses, i.e. those with the same prefix, might be located close together geographically but this is not necessarily the case. The Internet is constantly changing and getting bigger, so it is not possible for it to have a geographically based addressing scheme.

ROUTING SIMULATION ACTIVITY

Students, working in groups to simulate the routers and build up routing tables, can act out this algorithm. Acting out the algorithm gives students an understanding of how routing tables are generated, and gives them an appreciation of how computers can work together to execute a distributed algorithm.

The Algorithm

```
# Note: this algorithm is performed simultaneously by
#       all routers
#
# initialize routing table
enter 0 as the distance to your router (& "-" as gateway)

while your routing table changes:
  exchange routing tables with each of your neighbours
  for each neighbour's routing table:
    for each destination that is new or has a shorter
distance than yours:
      enter neighbour as the gateway to the destination
      enter the neighbour's distance + 1 in the distance
column
      # +1 is for the link between you and your neighbour
    end #foreach
  end #foreach
end #while
```

The algorithm ends when a routers table stops changing. The final table contains information about all routers in the network. For each destination, the table contains the neighbour to send messages through in order for the message to get them, as well as the distance to the destination. This algorithm generates routing tables that route messages by shortest possible path through the network.

This algorithm is slightly different than the one presented in the session. The difference is that it allows the routing table to be updated based on each neighbour's table individually. This has the same result, and many students will figure this out on their own. The problem with this version is that students may have to change their tables, which makes them messier (or requires erasing). On the other hand, the version done in the session requires synchronization to be able to examine all the neighbours' tables at once, and it can be difficult (and boring) to keep students synchronized. The choice is up to you. [If you have ideas about how to refine the algorithm please let me know!]

Did you know? This algorithm is called the *Floyd Warshall* or *all-pairs shortest path* algorithm. It is an algorithm that is usually covered in upper-year university math and computer science courses! It is a *distributed algorithm*, as multiple routers work together to determine their routing tables.

Props

Routing tables are written on pre-made blank table grids with one row per router and columns for *destination*, *gateway router*, and *distance*. To keep students from knowing the exact number of routers, additional routers can be added to the table (i.e. have A-Z in the table but only 18 of these routers actually exist). This allows students to better understand that they do not need to know the topology, or number of routers, for this algorithm to work. Each router needs 1 table to update + 1 copy per link to exchange.

The routers are connected with physical links that allow routing tables and messages to be exchanged between neighbours. We have experimented with different mediums for this and have found that most cloth string is too flimsy, especially for long links. For this reason we decided to use telephone cable. Plastic string could also be used. The medium must also be smooth to prevent the messages from getting stuck on the line. Paint stir sticks were attaches to the ends of the lines to allow them to be held easily and labeled.

The lines also need a way for messages to be attached and passed down them. We chose to use large black paper clips to do this as they allowed the papers to be easily clipped on and removed, and also slid down the lines quite well.

Network Design and Setup

The activity can be done with 1+ students per router, but 2+ is better as it is difficult for students to update all the copies of the routing table and send/receive messages at the same time.

Having 3 or more links per router makes it more exciting for the students. Only 1 link for a router means that all routing will use that link (boring!). Only 2 links means that every packet will go through one or the other and the topology is still limited. But, with 3+ students will have to think more when updating the tables and sending out packets.

When planning and laying out the network, try to minimize the number of overlapping links. This will make it easier to pass messages without getting in the way of other links (i.e. link interference).

Colour-coding the routers makes it easier to describe and assign router groups, as well as easier for students to figure out which routing tables are theirs and which are neighbours. The routers can be coloured in most topologies such that a router and all its neighbours are different colours. For example, a network with 3 links per router (i.e. a 3-connected graph) only needs 4 colours to do this.

Comparison to other routing table generation algorithms

Some networks are static – i.e. the hosts and connections stay the same. In such networks the routers can be given a fixed routing table that is generated based on knowledge of the network's fixed topology.

In dynamic networks, where links and hosts can be added and removed, the routers need to update their routing tables to take changes into account. Routers in dynamic networks can use algorithms like the Floyd-Warshall algorithm to re-generate their routing tables from time to time, and take into account changes to the network topology.

Some applications are sensitive to data loss and require a minimum amount of bandwidth in order to function properly. Examples are telephones and streaming movie and video. Such applications require the ability to *reserve* bandwidth on the network that only they can use. This is how telephone networks and some computer networks work. In the future, the Internet may have this capability too. Other types of applications, such as web browsers and chat programs, can handle data loss better. They can re-send or re-request information when it is lost on the network or takes too long to arrive. Supporting reserved bandwidth means that the network cannot use shortest path routing as transfers without reservations may need to route around reserved portions of the network, taking non-shortest path routes.

Routing in the Internet uses what is called *path-based* routing. Routers in the Internet store paths to destinations instead of just the gateway to pass packets on to (like the algorithm looked at here). The Internet does not use shortest-path routing.

Using this Activity to Teach the Curriculum

Related Curriculum Expectations

Hardware, Interfaces, and Networking Systems

- explain the role of a network in accessing computer software resources
- describe the issues involved in maintaining a software library (e.g., access, backup, version control)
- relate hardware requirements to user software demands

Hardware, Interfaces, and Networking Systems

- explain the function and interaction of the basic components of network configurations
- identify similarities and differences among several network topologies and protocols

Tying it in

This activity can be discussed in the context of:

- *network topology:* discussing the components and structure of a network and how it facilitates communication
- *types of networks:* LANs vs. WANs, vs. the Internet static vs. dynamic, …
- *routers and routing algorithms:* discussing the function of routers, routing tables, and routing table generation
- *the Internet:* how it works, I.P. addressing, and message routing
- distributed applications: how they – e.g. chat programs, e-mail, distributed file systems, networked printers, etc. – communicate

References

- *How the Internet Infrastructure Works:* http://computer.howstuffworks.com/internet-infrastructure.htm
- *How Routing Algorithms Work:* http://computer.howstuffworks.com/routing-algorithm.htm
- *How Instant Messaging Works:* http://computer.howstuffworks.com/instant-messaging.htm
- *How Interplanetary Internet Will Work:* http://computer.howstuffworks.com/interplanetary-internet.htm