Module 8
Routing

Lesson

28

Routing III

8.3.1 BORDER GATEWAY PROTOCOL

Border Gateway protocol (BGP) is an inter-autonomous system protocol that first appeared in 1989. BGP is based on path vector routing. A need was felt to introduce a new protocol for routing between ASes because the previous two protocols were found lacking in certain aspects needed for routing in the internet.

Distance vector routing is not a good candidate because there are occasions when the route with the smallest hop count is not necessarily the preferred route. For example we may not want a packet to pass through a particular AS that is not secure even though it is the shortest route. Also distance vector routing does not specify the actual path used. A router that received the distance vector may be fooled if the shortest path is actually calculated through the receiving router itself (explained with an example later).

Link state routing is also not good because an internet is usually too big for this routing method. To use link state routing for the whole internet would require each router to have a huge link state database. It would also take a long time to calculate its routing table using the Dijkstra's algorithm.

PATH VECTOR ROUTING AND PATH VECTOR MESSAGES

Each entry in the routing table contains the destination network, the next router, and the path to reach the destination which is usually defined as an ordered list of autonomous systems that a packet should travel to reach its destination. The autonomous boundary routers advertise the reachability of the routers in their own AS. The AS boundary router receives this information via the interior routing algorithm such as RIP or OSPF.

Each router verifies that this information received is consistent with its routing policy, updates its routing table if satisfied and modifies the message before sending the message to the next neighbor. After all the paths come in from the neighbors, the router that has to select a path to a destination router in another AS, examines them to see which is the best. It checks the remaining routes and selects the best possible taking into consideration the various routing policies that it has to follow.

TYPES OF MESSAGES

BGP uses four different types of messages: open, update, keep-alive, and notification

To create a neighborhood relationship, a router running BGP opens a connection with a neighbor and sends an **open message**. If the neighbor accepts the neighborhood relationship, it responds with a **keep-alive message.** The **update message** is used by the router to remove previously advertised routers, announce a new route to a destination or both. A **notification message** is sent by a router whenever an error condition is detected or a router wants to close the connection.

8.3.2 FLOW-BASED ROUTING

The algorithms studied so far take only the topology into account. They do not consider the load. In this section we will study a static algorithm that uses both topology and load for routing. It is called flow-based routing.

In some networks, the mean data flow between each pair of nodes is relatively stable and predictable. For example, in a corporate network for a retail store chain, each store might send orders, sales reports, inventory updates, and other well-defined types of messages to known sites in a predefined pattern, so that the total volume of traffic varies little from day to day. Under conditions in which the average traffic from i to j is known in advance and, to a reasonable approximation, constant in time, it is possible to analyze the flows mathematically to optimize the routing.

The basic idea behind the analysis is tttat for a given line, if the capacity and average flow are known, it is possible to compute the mean packet delay on that line from queueing theory. From the mean delays on all the lines, it is straightforward to calculate a flow-weighted average to get the mean packet delay for the whole subnet. The routing problem then reduces to finding the routing algorithm that produces the minimum average delay for the subnet.

To use this technique, certain information must be known in advance. First the subnet topology must be known. Second, the traffic matrix, *Fij'* must be given. Third, the line capacity matrix, *C;j'* specifying the capacity of each line in bps must be available. Finally, a (possibly tentative) routing algorithm must be chosen.

8.2.3 LINK STATE ROUTING

Distance vector routing was used in the ARPANET until 1979, when it was replaced by link state routing. Two primary problems caused its demise. First, since the delay metric was queue length, it did not take line bandwidth into account when choosing routes. Initially, all the lines were 56 kbps, so line bandwidth was not an issue, but after some lines had been upgraded to 230 kbps and others to 1.544 Mbps, not taking bandwidth into account was a major problem. Of course, it would have been possible to change the delay metric to factor in line bandwidth, but a second problem also existed, namely, the algorithm often took too long to converge, even with tricks like split horizon. For these reasons, it was replaced by an entirely new algorithm now called link state routing. Variants of link state routing are now widely used. The idea behind link state routing is simple and can be stated as five parts.

Each router must

1. Discover its neighbors and learn their network addresses.
2. Measure the delay or cost to each of its neighbors
3. Construct a packet telling all it has just learned
4. Send this packet to all other routers.
5. Compute the shortest path to every other router.

8.2.4 HIERARCHICAL ROUTING

As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them. At a certain point the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network. When hierarchical routing is used, the routers are divided into what we will call regions, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions. When different networks are connected together, it is natural to regard each one as a separate region in order to free the routers in one network from having to know the topological structure of the other ones. For huge networks a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups and so on, until we run out of names for aggregations. As an example of a Multi-level hierarchy, consider how a packet might be routed

from Kharagpur to Los Angeles, USA. The Kharagpur router would know the detailed topology within West Bengal but would send all out-of-state traffic to the Kolkata router. The Kolkata router would be able to route traffic to other domestic routers, but would send foreign traffic to Mumbai. The Mumbai router would be programmed to direct all traffic to the router in the destination country responsible for handling foreign traffic, say in New York. Finally, the packet would work its way down the tree in USA until it got to Los Angeles.

Unfortunately there is a penalty involved, because in hierarchical routing, the best route may not be followed.

Objective Questions

28.01

Subjection Questions

28.11

Level 2 Questions

28.21