A Quick Introduction to Internet Routing

Handouts: Course information, tentative schedule, reading list, potential projects.

Lecture Outline:

- Administrivia.
- Introduction to Internet routing.

1 Internet routing

We begin this course with a quick introduction to Internet routing. The reader is referred to networking textbooks [BG92, LGW00, PD00] for details. The central framework for Internet communication is the TCP/IP protocol stack. Internet protocols can be organized in a layered hierarchy. While a 7-layer hierarchy (the OSI model) is commonly found in most textbooks, the TCP/IP stack can be better captured by a simpler 5-layer hierarchy.

Addressing. Each node in the Internet is assigned a 32-bit long IP address. For example, the IP address of the machine `denali` in our CCS network is 129.10.116.200. The IP address and the hostname `denali.ccs.neu.edu` do not necessarily have a common hierarchical relationship. Thus, one needs to be able to map a hostname (which is what is typically known at the time of initiating a communication) to an IP address. This is done by the *Domain Name Service* (DNS). (We will look at DNS and other related location services in greater detail later on in the course.) You can lookup the IP address of `denali.ccs.neu.edu` by simply doing an `nslookup` on the address on your UNIX account.

Address resolution. Local-area networks use hardware addresses for routing. Thus, we need to translate an IP address to a hardware address before a particular packet, that has reached the appropriate LAN, can be routed to its destination in the LAN. In commonly used address resolution protocols, the node that wants the address translation, sends a broadcast message requesting the translation; the intended destination responds with its hardware address which is then cached at the first node. While hardware addresses are preconfigured, IP addresses are either statically assigned by the network administrator or a particular server, or dynamically assigned by a DHCP server.

Internet Protocol (IP). IP is a protocol that provides a best-effort service between communicating nodes. This service, which is connectionless and unreliable, consists of two key components: *forwarding* and *routing*. Forwarding is the process by which a node takes a packet received along one of its incident (input) links and sends the packet on a particular output link, as determined by a *forwarding table*. Routing is the process of building up the tables that allow the correct output link to be determined for a packet that is being forwarded.

IP is a protocol used for forwarding packets. Every IP datagram, contains the IP address of the destination host. The "network" part of the IP address uniquely determines a single physical network. So IP attempts to deliver the packet to this physical network, after which the LAN can directly send to the packet to its final destination (using an address resolution protocol, if necessary).

The protocols used for constructing the forwarding table (and intermediate routing tables) are the heart of Internet routing. And we will discuss the associated algorithmic problems in greater depth in this course.

Transport protocols. The most commonly used transport protocol for transferring non-real-time data is the Transmission Control Protocol (TCP). It provides a reliable, stream-based protocol. It guarantees in-order delivery through retransmission and sequence number techniques. Another widely-used protocol is the connectionless user datagram protocol (UDP), which is employed for streaming media and in applications where packet losses can be tolerated.

2 Routing protocols

The Internet is organized into domains, referred to as *autonomous systems* (AS). Each AS is assigned a unique ID, which is unrelated to IP addresses. There are currently up to 2^{16} ASs allowed. The ASs connect with one another at Network Access Points and private peering points. Each AS is usually under a single independent administrative domain, and hence can implement routing policies within its network, independent of the other ASs. Consequently, Internet routing can be divided into two parts: *intradomain routing*, which is routing within a domain, and *interdomain routing*, which is routing among domains.

2.1 Intradomain routing

Intradomain routing is implemented using shortest-paths routing protocols. There are two classes of such protocols that are widely deployed. One is the class of distance-vector protocols, based on the Bellman-Ford algorithm, and the other is the class of link-state protocols, based on Dijkstra's algorithm. Both have their advantages and disadvantages. The Bellman-Ford algorithm is inherently distributed and hence scales well; however, it does not respond quickly to changes in the network conditions. A well-known problem for distance-vector protocols is the count-to-infinity problem. Heuristics such as *poison reverse* have been proposed for addressing this problem, but one can show that there are scenarios where these heuristics fail. Link state protocols involve doing an all-to-all broadcast of link state information. While these protocols tend to respond quickly to network dynamics, they are not scalable due to the non-local nature of the protocol.

2.1.1 The count-to-infinity problem

While the distance vector protocols eventually converge to the shortest path values regardless of the initial distance estimates, the time for convergence can be made arbitrarily large and this aspect of the distance vector protocols is a major shortcoming for effective use in the Internet. Consider the following simple network consisting of 4 nodes and 3 links.

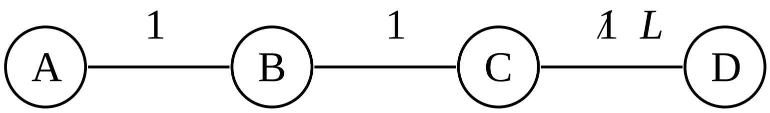

Figure 1: Example network for the count-to-infinity problem.

Suppose initially, each link has weight 1 and each node has calculated the shortest path distance to D. Thus the distances of A, B, and C to D are 1, 2, and 3, respectively. Now suppose that the weight of the link (C,D) changes from 1 to a large positive integer, say L. Assuming a synchronous behavior of the Bellman-Ford algorithm, in the subsequent iteration, C will set its distance estimate to D as 3 since B supposedly has a path to D of length 2. In the following iteration B will change its distance estimate to 4 since the best path it has is through C. In general, after i iterations (i odd), the distance estimates to D of A, B, and C will be $i+2$, $i+1$, and $i+2$, respectively. This will continue until C's estimate reaches L, after which it will not further increase because the link (C, D) has weight L. Thus the number of iterations taken to converge to the correct shortest path values is proportional to L, which could be made arbitrary large. This phenomenon is referred to as the "bad news phenomenon" since the algorithm takes a long time to react to a sudden increase in a link weight. The resultant problem is referred to as the "count-to-infinity" problem.

One apparent fix for the above problem is for a node X to not send its distance estimate to neighbor Y if Y is on the shortest path from X to the destination. Thus, in the above example, if node B does not send its distance estimate to C in the first iteration, then C will calculate L as its shortest distance. Then B will calculate its distance to be $L+1$, and finally, A will calculate its distance as $L+2$, thus converging to a correct solution in 3 iterations. This technique is referred to as the *split horizon*. Unfortunately, it does not solve the count-to-infinity problem for all networks. See the example given in Figure 5.36 of [BG92]. Another heuristic employed to address the above count-to-infinity problem is *split horizon with poison reverse*, in which node Y sends a distance estimate (to destination Z, say) of infinity to neighbor X if X is the next node on the shortest path to Z.

2.2 Interdomain routing

The *de facto* protocol for interdomain routing is the *Border Gateway Protocol* (BGP). The autonomous systems exchange reachability information with one another using BGP. The protocol allows each AS to override distance-based metrics with policy-based metrics when choosing best routes. From an implementation standpoint, BGP can be subdivided into two protocols: iBGP and eBGP. The protocol eBGP is used for communication among BGP routers in different domains, while iBGP is used for communication among routers in the same domain.

Since each AS selects its own routing policy, it is difficult to select a single metric that is acceptable to all ASes. Consequently, a shortest-paths approach is not feasible for interdomain routing. Instead, we have a more modest goal: for every source-destination pair, find any path that is loop-free (assuming one exists).

Using eBGP, routers exchange reachability information by exchanging entire routes (which are specified by a list of ASes). For any destination (in reality, a CIDR block) that an AS can reach, the AS advertises: (i) the destination itself; (ii) the IP address of the next hop on the path; (iii) the list of ASes on the path to destination.

Using iBGP, routers exchange route records in the same way as above; in addition they also exchange policy information represented by local preference rules. An AS may have different preferences for different paths to the same destination. Example reference rules are "Use AS B only to reach this destination", "Use minimum number of ASs to reach these addresses", and "Use AS x in preference to AS y". These preference rules are internal to the AS and are not known to the other ASes. Consequently, they are only exchanged among BGP routers in the same AS.

When a BGP router hears of a route, it applies its local preference rules and recomputes the most preferred route. Then it may advertise this route to other BGP routers both within its AS and outside its AS. BGP is referred to as a *path vector protocol*, since it exchanges paths (to different destinations) with neighboring routers, rather than distances as a distance-vector protocol does.

We refer the reader to networking textbooks for a more detailed description of BGP [PD00, LGW00].

References

[BG92] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[LGW00] A. Leon-Garcia and I. Widjaja. *Communication Networks: Fundamental Concepts and Key Architectures.* McGraw-Hill, 2000.

[PD00] L. Peterson and B. Davie. *Computer Networks: A Systems Approach.* Morgan Kaufmann, 2000.