Week 8 Lecture 1

Network Layer: Data Plane + Control Plane

# NAT: network address translation (very important)

**Private addresses**

10.0.0.0/8 (16,777,216 hosts)

172.16.0.0/12 (1,048,576 hosts)

192.168.0.0/16 (65536 hosts)

The above IP addresses are not routable. A network IP address that is not private is public

Diagram

Description automatically generated

The NAT-enabled router does not look like a router to the outside world. Instead, the NAT router behaves to the outside world as a single device with a single IP address. In Figure 4.25, all traffic leaving the home router for the larger Internet has a source IP address of 138.76.29.7, and all traffic entering the home router must have a destination address of 138.76.29.7.

Suppose a user sitting in a home network behind host 10.0.0.1requests a Web page on some Web server (port 80) with IP address 128.119.40.186. The host 10.0.0.1 assigns the (arbitrary) source port number 3345 and sends the datagram into the LAN. The NAT router receives the datagram, generates a new source port number 5001 for the datagram, replaces the source IP address with its WAN-side IP address 138.76.29.7, and replaces the original source port number 3345 with the new source port number 5001.

When generating a new source port number, the NAT router can select any source port number that is not currently in the NAT translation table. NAT in the router also adds an entry to its NAT translation table. The Web server, blissfully unaware that the arriving datagram containing the HTTP request has been manipulated by the NAT router, responds with a datagram whose destination address is the IP address of the NAT router, and whose destination port number is 5001. When this datagram arrives at the NAT router, the router indexes the NAT translation table using the destination IP address and destination port number to obtain the appropriate IP address (10.0.0.1) and destination port number (3345) for the browser in the home network. The router then rewrites the datagram’s destination address and destination port number and forwards the datagram into the home network.

**Disadvantages**

* NAT violates the architectural model of IP:
  + Every IP address uniquely identifies a single node on Internet
  + routers should only process up to layer 3
* NAT changes the Internet from connection less to a kind of connection-oriented network
* NAT possibility must be considered by app designers, e.g., P2P applications

**Advantages**

* range of addresses not needed from ISP: just one IP address for all devices
* can change addresses of devices in local network without notifying outside world
* can change ISP without changing addresses of devices in local network

Graphical user interface, application

Description automatically generated

10.248.15.210 means it is a private IP address, the WAN side IP address is 129.94.8.210.

# Table Description automatically generatedIPV6 (not included in exam)

* initial motivation: 32-bit address space soon to be completely allocated.
* IPv6 datagram format:
  + fixed-length 40-byte header
  + no fragmentation allowed

Network layer, control plane

# 5.1 Introduction

Two approaches to structuring network control plane:

* per-router control (traditional)

Individual routing algorithm components in each and every router interact with each other in control plane to compute forwarding tables

* logically centralized control (software defined networking)

# Routing protocols

Abbreviations:

* Autonomous System (AS)
* Open Shortest Path First (OSPF)
* Routing Information Protocol (RIP)
* Border Gateway Protocol (BGP)

# Intra-domain routing protocol

Each AS runs an intra-domain routing protocol that establishes routers within its domain

**link state (lecture 1)**

Least-cost paths => shortest paths (hop count)

|  |  |
| --- | --- |
| Link State (Global) | Distance Vector (Decentralised) |
| Routers maintain cost of each link in the network | Routers maintain next hop & cost of each destination. |
| Connectivity/cost changes flooded to all routers (each router knows exactly how the network looks like) | Connectivity/cost changes iteratively propagate form neighbour to neighbour (Only knows its neighbours) |
| Converges quickly (less inconsistency, looping, etc.) | Requires multiple rounds to converge |
| Limited network sizes, used in small networks | Scales to large networks |

* Each node maintains its local “link state” (LS): i.e., a list of its directly attached links and their costs
* Each node floods its local link state: on receiving a new LS message, a router forwards the message to all its neighbours other than the one it received the message from (Flooding LSAs)

## Dijkstra’s algorithm

Text

Description automatically generated

A picture containing diagram

Description automatically generated

This is going to be in the exam!

* In the initialization step, the currently known least-cost paths from A to its directly attached neighbours, particular that the cost to C is set to 5 since this is the cost of the direct (one hop) link from A to C. The costs to E and F are set to infinity because they are not directly connected to u.
* In the first iteration, we look among those nodes not yet added to the set N′ and find that node with the least cost as of the end of the previous iteration. That node is D, with a cost of 1, and thus D is added to the set N′. Line 12 of the LS algorithm is then performed to update D(v) for all nodes v, yielding the results shown in the second line in the table. The cost of the path to B and C is unchanged. The cost of the path to C (which was 5 at the end of the initialization) through node x is found to have a cost of 4. Hence this lower-cost path is selected and C’s predecessor along the shortest path from A is set to D. Similarly, the cost to E (through D) is computed to be 2, and the table is updated accordingly.
* In the second iteration, nodes B and E are found to have the least-cost paths (2), and we break the tie arbitrarily and add E to the set N′ so that N′ now contains A, D, and E. The cost to the remaining nodes not yet in N′, that is, nodes B, C, and F, are updated via line 12 of the LS algorithm, yielding the results shown in the third row in the table
* And so on . . .

Link state issues:

* Issue #1: Scalability
* Issue #2: Transient Disruptions (loop, if one of the routers goes offline, some routers know about failure before others, then the shortest paths are no longer consistent, which causes transient forwarding loops, it takes time for all routers to update the tables)

Link cost is dynamic, which causes oscillations