Tutorial 3
Network Layer

Alan Edwin
Nehir Kırkgöz

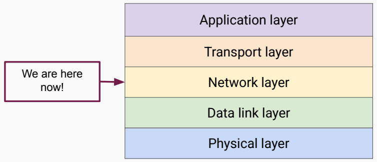
Important Deadlines & Announcements

Self-Study Submission (PART 1) – 600 pts
- Deadline: Friday, 24 April, at 23:59

Self-Study PART 1 Interviews – Pass/Fail
- Date: Friday, 8 May 2026, from 15:30 until 18:30

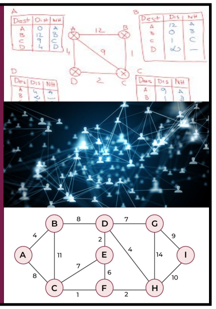
Lab Assignments (Extra) – 3,700 pts
- Deadline: Tuesday, 12 May 2026, 12:45

Where Are We Now?



Menu of the Day

1. **Conceptual Overview**
   a. Distance Vector Routing
   b. Link State Routing
   c. Border Gateway Protocol
2. **IP Protocol**
   a. Hierarchical Routing
   b. Prefix Matching
   c. Traffic Shaping



Conceptual Overview – Recap

What is the main functionality of the Network Layer?

Conceptual Overview – Recap

What is the main functionality of the Network Layer?

Routing packets from source to destination

Conceptual Overview – Recap

What is the main functionality of the Network Layer?

Routing packets from source to destination

A router has two primary processes…

a. **Packet Forwarding** – Handling packets as they arrive
b. **Routing Algorithms** – Filling In/Updating Routing Tables

Important Terminology

Routing Tables – A data table that helps determine the best path for sending data packets across a network

Distance Vector Routing – Introduction

A router C uses distance vector routing and receives the distance vectors on the right from its neighbors.

C starts with an empty routing table and has a distance to A and F of 4 and 1 respectively.

What does the routing table of router C look like after processing these?

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

Distance Vector Routing – Introduction

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

Distance Vector Routing – Introduction

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

From C

To	Distance
A	4
C	0
F	1

Distance Vector Routing – Introduction

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

From C

To	Distance
A	4
C	0
F	1

From C

To	Distance	Link

Distance Vector Routing – Introduction

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

From C

To	Distance
A	4
C	0
F	1

From C

To	Distance	Link
A	4	A

Distance Vector Routing – Introduction

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

From C

To	Distance
A	4
C	0
F	1

From C

To	Distance	Link
A	4	A
B	8 (4 + 4)	A

Distance Vector Routing – Introduction

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

From C

To	Distance
A	4
C	0
F	1

From C

To	Distance	Link
A	4	A
B	8 (4 + 4)	A
C	0	–

Distance Vector Routing – Introduction

From A

To	Distance
A	0
B	4
C	6
D	100
E	34
F	1

From F

To	Distance
A	6
B	85
C	34
D	2
E	1
F	0

From C

To	Distance
A	4
C	0
F	1

From C

To	Distance	Link
A	4	A
B	8 (4 + 4)	A
C	0	–
D	3 (1 + 2)	F
E	2 (1 + 1)	F
F	1	F

Distance Vector Routing – Exercise 1

A router C uses distance vector routing and receives the distance vectors on the right from its neighbors.

C starts with an empty routing table and has a distance to A and F of 5 and 21 respectively.

What does the routing table of router C look like after processing these?

From A

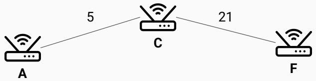
To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

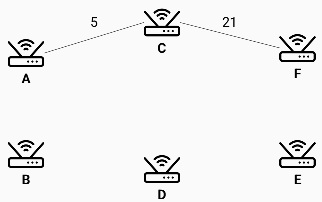
To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

Distance Vector Routing – Exercise 1 Solution

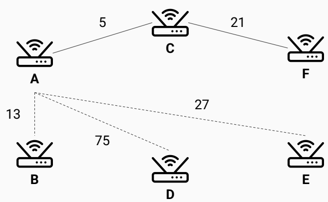
Distance Vector Routing – Exercise 1 Solution



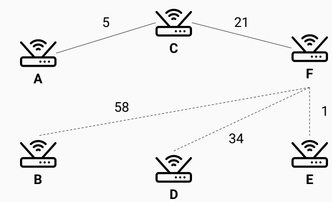
Distance Vector Routing – Exercise 1 Solution



Distance Vector Routing – Exercise 1 Solution



Distance Vector Routing – Exercise 1 Solution



Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

From C

To	Distance
A	5
C	0
F	21

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

From C

To	Distance
A	5
C	0
F	21

From C

To	Distance	Link
A	5	A
B	18 (5 + 13)	A
C	0	–
D	55 (21 + 34)	F
E	22 (21 + 1)	F
F	21	F

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

From C

To	Distance
A	5
C	0
F	21

From C

To	Distance	Link
A	5	A

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

From C

To	Distance
A	5
C	0
F	21

From C

To	Distance	Link
A	5	A
B	18 (5 + 13)	A

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

From C

To	Distance
A	5
C	0
F	21

From C

To	Distance	Link
A	5	A
B	18 (5 + 13)	A
C	0	–

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	
B	
C	
D	
E	1
F	0

From C

What do we set the distance to if a router is unreachable?

From C

To	Distance	Link
A	5	A
B	18 (5 + 13)	A
C	0	–

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

From C

To	Distance
A	5
C	0
F	21

From C

To	Distance	Link
A	5	A
B	18 (5 + 13)	A
C	0	–

Distance Vector Routing – Exercise 1 Solution

From A

To	Distance
A	0
B	13
C	6
D	75
E	27
F	103

From F

To	Distance
A	7
B	58
C	20
D	34
E	1
F	0

From C

To	Distance
A	5
C	0
F	21

From C

To	Distance	Link
A	5	A
B	18 (5 + 13)	A
C	0	–
D	55 (21 + 34)	F
E	22 (21 + 1)	F
F	21	F

Distance Vector Routing – Exercise 1 Solution

From A	
To	**Distance**
A	0
B	13
C	6
D	75
E	27
F	103

From F	
To	**Distance**
A	7
B	58
C	20
D	34
E	1
F	0

From C	
To	**Distance**
A	5
C	0
F	21

From C		
To	**Distance**	**Link**
A	5	A
B	18 (5 + 13)	A
C	0	–
D	55 (21 + 34)	F
E	22 (21 + 1)	F
F	21	F

Distance Vector Routing – Exercise 1 Solution

From A	
To	**Distance**
A	0
B	13
C	6
D	75
E	27
F	103

From F	
To	
A	
B	
C	
D	
E	1
F	0

From C

Why is the distance from A to C <u>different</u> from the distance from C to A?

From C		
To	**Distance**	**Link**
A	5	A
B	18 (5 + 13)	A
C	0	–
D	55 (21 + 34)	F
E	22 (21 + 1)	F
F	21	F

Link State Routing – Introduction

In Link State Routing…

Flooding – Send information about direct neighbours to the whole network

Based on received messages, construct network overview and run shortest path algorithm

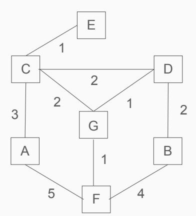
Each router knows the topology of the **entire** network

Link State Routing – Exercise 2

See the system in the Figure. Assume the router has knowledge of the entire network.

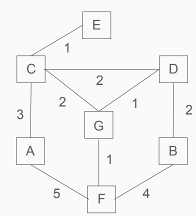
Show what the link state packet transmitted by F is. Who receives this packet?

What is the routing table constructed by F?



Link State Routing – Exercise 2 Solution

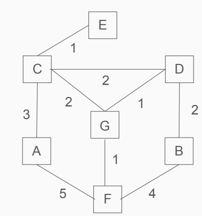
Transmitted by F	
To	**Distance**
A	5
G	1
B	4



Link State Routing – Exercise 2 Solution

Who receives this?

Transmitted by F	
To	**Distance**
A	5
G	1
B	4



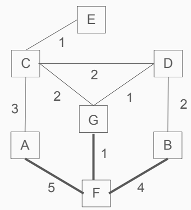
Link State Routing – Exercise 2 Solution

Transmitted by F

To	Distance
A	5
G	1
B	4

Routing Table for F

To	Distance	Link
F	–	–
A	5	A
G	1	G
B	4	B
C		
D		
E		



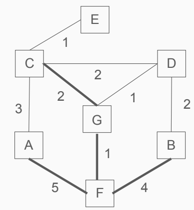
Link State Routing – Exercise 2 Solution

Transmitted by F

To	Distance
A	5
G	1
B	4

Routing Table for F

To	Distance	Link
F	–	–
A	5	A
G	1	G
B	4	B
C	3 (1 + 2)	G
D		
E		



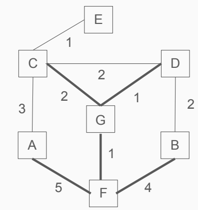
Link State Routing – Exercise 2 Solution

Transmitted by F

To	Distance
A	5
G	1
B	4

Routing Table for F

To	Distance	Link
F	–	–
A	5	A
G	1	G
B	4	B
C	3 (1 + 2)	G
D	2 (1 + 1)	G
E		



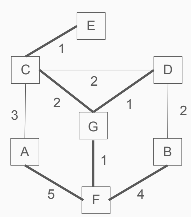
Link State Routing – Exercise 2 Solution

Transmitted by F

To	Distance
A	5
G	1
B	4

Routing Table for F

To	Distance	Link
F	–	–
A	5	A
G	1	G
B	4	B
C	3 (1 + 2)	G
D	2 (1 + 1)	G
E	4 (1 + 2 + 1)	G



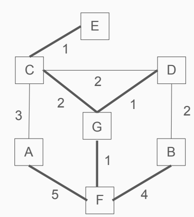
Link State Routing – Exercise 2 Solution

Transmitted by F

To	Distance
A	5
G	1
B	4

Routing Table for F

To	Distance	Link
F	–	–
A	5	A
G	1	G
B	4	B
C	3 (1 + 2)	G
D	2 (1 + 1)	G
E	4 (1 + 2 + 1)	G



Link State Routing – Exercise 2 Solution

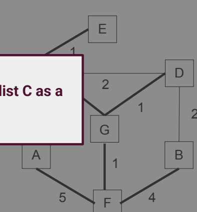
Transmitted by F

To	Distance
A	5
G	1
B	4

Routing Table for F

To	Distance	Link
F		
A		
G		
B		
C	3 (1 + 2)	G
D	2 (1 + 1)	G
E	4 (1 + 2 + 1)	G

Why do we not list C as a link?

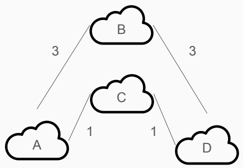


Border Gateway Protocol – Exercise 3

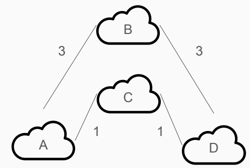
BGP is a policy-based protocol. Whether a provider transmits traffic is based on policy rather than performance.

B provides transit service for the traffic from A and for the traffic to D. A and C have no relation (i.e C will not transmit traffic from A).

What is the best path from A to D and what is the path that will be taken?

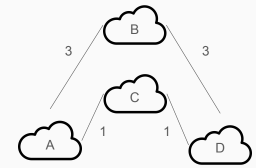


Border Gateway Protocol – Exercise 3



Border Gateway Protocol – Exercise 3

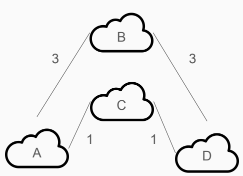
Best Path from A to D: 2 s



Border Gateway Protocol – Exercise 3

Best Path from A to D: 2 s

Route that will be Taken: 6 s

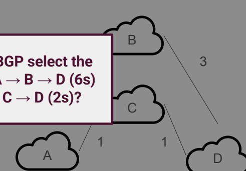


Border Gateway Protocol – Exercise 3

Best Path from A

Route that will be

Why does BGP select the path from A → B → D (6s) <u>over</u> A → C → D (2s)?



Distance Vector Routing vs. Link State Routing – Summary

1) **Distance Vector Routing**
   a) Routers are only aware of their direct neighbours' information
   b) The routing table includes the distance from one router to another

2) **Link State Routing**
   a) Routers are only aware of the entire network
   b) The routing table includes the distance from one router to another as well as the link used to arrive at the destination

Tutorial 3 - Network Layer

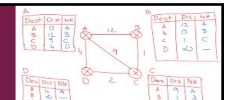
Break

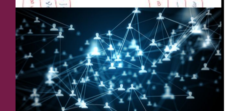

Alan Edwin

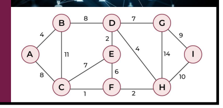

Nehir Kırkgöz

Menu of the Day

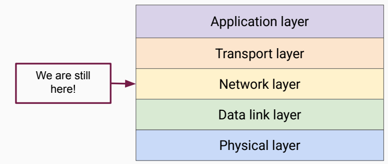
1. ~~Conceptual Overview~~
    a. ~~Distance Vector Routing~~
    b. ~~Link State Routing~~
    c. ~~Border Gateway Protocol~~
2. **IP Protocol**
    a. Hierarchical Routing
    b. Prefix Matching
    c. Traffic Shaping







Where Are We Now?



IP Addresses Recap

IPv4
128.208.2.151

IP Addresses Recap

IPv4
128.208.2.151

Address size:
- 32 bits

IP Addresses Recap

IPv4
128.208.2.151

Address size:
- 32 bits

Number of addresses:

$$2^{32} = 4,294,967,296$$

IP Addresses Recap

IPv4
128.208.2.151

Address size:
- 32 bits

Number of addresses:

$2^{32} = 4,294,967,296$

IPv6
8000::123:4567:89AB:CDEF

IP Addresses Recap

IPv4
128.208.2.151

Address size:
- 32 bits

Number of addresses:

$2^{32} = 4,294,967,296$

IPv6
8000::123:4567:89AB:CDEF

Address size:
- 128 bits

IP Addresses Recap

IPv4
128.208.2.151

Address size:
- 32 bits

Number of addresses:

$2^{32} = 4,294,967,296$

IPv6
8000::123:4567:89AB:CDEF

Address size:
- 128 bits

Number of addresses:

$2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$

IP Addresses Recap

IPv4
128.208.2.151

Address size:
- 32 bits

Number of addresses:

$2^{32} = 4,294,967,296$

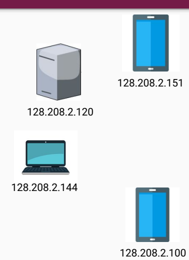
IPv6
8000::123:4567:89AB:CDEF

Number of addresses:

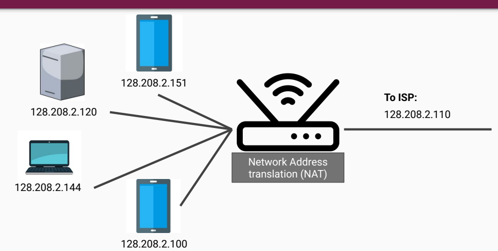
$2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$

If IPv6 is better why do we still have IPv4?

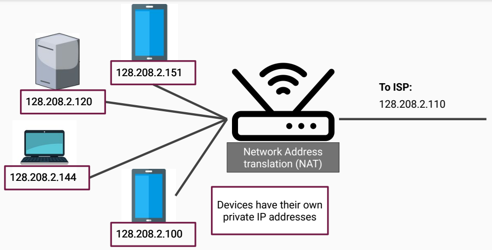
IP Addresses Recap



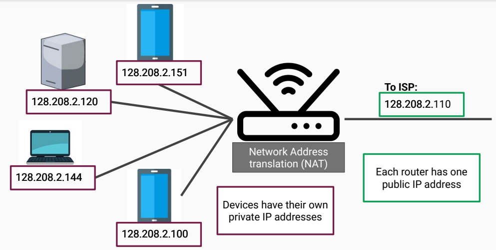
IP Addresses Recap



IP Addresses Recap



IP Addresses Recap



[Recap] Hierarchical Routing

[Recap] Hierarchical Routing

Routers used to keept a map to every device.

[Recap] Hierarchical Routing

Routers used to keept a map to every device.
- Not too many devices

[Recap] Hierarchical Routing

Routers used to keept a map to every device.
- Not too many devices
- Routing tables were small

[Recap] Hierarchical Routing

Routers used to keept a map to every device.
- Not too many devices
- Routing tables were small

Then Internet started booming

4.3 billion IPv4 addresses

Each address has 4 bytes

[Recap] Hierarchical Routing

Routers used to keept a map to every device.
- Not too many devices
- Routing tables were small

Then Internet started booming
- Networks grew
- Routing tables grew large

4.3 billion IPv4 addresses

Each address has 4 bytes

[Recap] Hierarchical Routing

Routers used to keept a map to every device.
- Not too many devices
- Routing tables were small

Then Internet started booming
- Networks grew
- Routing tables grew large

Router memory was depleted.
Increased CPU time to scan tables.
Recomputing Shortest paths was unfeasable.

4.3 billion IPv4 addresses

Each address has 4 bytes

[Recap] Hierarchical Routing

Routers used to keept a map to every device.
- Not too many devices
- Routing tables were small

Then Internet started booming
- Networks grew
- Routing tables grew large

Router memory was depleted.
Increased CPU time to scan tables.
Recomputing Shortest paths was unfeasable.

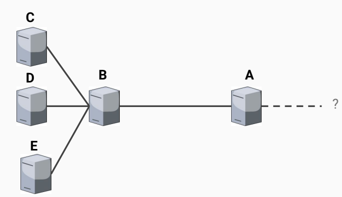
[SOLUTION] Hierarchical routing:
- divide routers into groups called areas/regions
- each router knows the topology of its region and how to reach other regions
- routers don't know the detailed structure of other regions

4.3 billion IPv4 addresses

Each address has 4 bytes

[Recap] Hierarchical Routing

[SOLUTION] Hierarchical routing:
- divide routers into groups called areas/regions
- each router knows the structure of its own region and how to reach other regions
- routers don't know the detailed structure of other regions



[Recap] Prefix Matching

Entry format: [ADDRESS] / [PREFIX_LENGTH]

Example: 127.126.111.29/10

[Recap] Prefix Matching

Entry format: [ADDRESS] / [PREFIX_LENGTH]

Example: 127.126.111.29/10 ← Number of bits representing the subnet

127.126.111.29

[Recap] Prefix Matching

Entry format: [ADDRESS] / [PREFIX_LENGTH]

Example: 127.126.111.29/10 ← Number of bits representing the subnet

01111111.01111110.01101111.00011101

[Recap] Prefix Matching

Entry format: [ADDRESS] / [PREFIX_LENGTH]

Example: 127.126.111.29/10 ← Number of bits representing the subnet

01111111.01111110.01101111.00011101

Subnet: 01111111.01

Host: 111110.01101111.00011101

[Recap] Prefix Matching

Entry format: [ADDRESS] / [PREFIX_LENGTH]

Example: 127.126.111.29/10 ← Number of bits representing the subnet

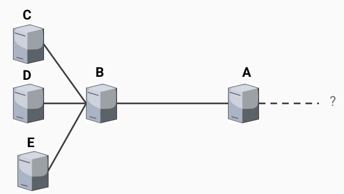
01111111.01111110.01101111.00011101

Subnet: 01111111.01

Host: 111110.01101111.00011101

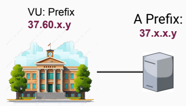
Longest prefix matching sends packets to the output port with the **longest prefix** that matches the **destination address of the packet**.

[Recap] Prefix Matching

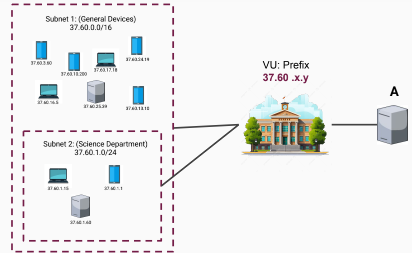


[Recap] Prefix Matching

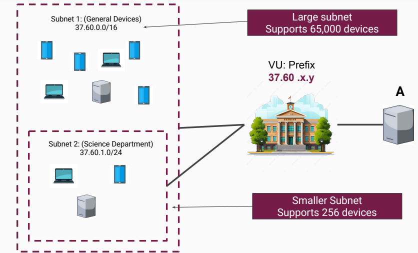
[Recap] Prefix Matching



[Recap] Prefix Matching



[Recap] Prefix Matching



Exercise

Consider the following routing table:

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Exercise

Consider the following routing table:

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

An incoming IP packet carries the destination address **137.70.32.128**. On which port is this packet forwarded? Assume that the router uses the **longest matching prefix**.

Solution

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Solution

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Solution

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Solution

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Solution

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Solution

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

B: (137.70.16.0/20)
10001001.01000110.00010000.00000000/20

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Solution

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

B: (137.70.16.0/20)
10001001.01000110.00010000.00000000/20

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Solution

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

B: (137.70.16.0/20)
10001001.01000110.00010000.00000000/20 ✗

Solution

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

B: (137.70.16.0/20)
10001001.01000110.00010000.00000000/20 ✗

C: (137.64.0.0/10)
10001001.01000000.00000000.00000000/10

Solution

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

B: (137.70.16.0/20)
10001001.01000110.00010000.00000000/20 ✗

C: (137.64.0.0/10)
10001001.01000000.00000000.00000000/10 ✓

Solution

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

B: (137.70.16.0/20)
10001001.01000110.00010000.00000000/20 ✗

C: (137.64.0.0/10)
10001001.01000000.00000000.00000000/10 ✓

D: (0.0.0.0/0)
00000000.00000000.00000000.00000000/0 ✓

Solution

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000.11000000/26 ✗

B: (137.70.16.0/20)
10001001.01000110.00010000.00000000/20 ✗

C: (137.64.0.0/10)
10001001.01000000.00000000.00000000/10 ✓ ← Answer: C (Longest Prefix)

D: (0.0.0.0/0)
00000000.00000000.00000000.00000000/0 ✓

Solution

Prefix	Port
137.70.32.192/26	A
137.70.16.0/20	B
137.64.0.0/10	C
0.0.0.0/0	D

Destination IP: (137.70.32.128)
10001001.01000110.00100000.10000000

A: (137.70.32.192/26)
10001001.01000110.00100000

B: (137.70.16.0/20)
10001001.01000110.00010000

Why should we match with longest prefix?

C: (137.64.0.0/10)
10001001.01000000.00000000.00000000/10 ✓ ← Answer: C (Longest Prefix)

D: (0.0.0.0/0)
00000000.00000000.00000000.00000000/0 ✓

Recap - CIDR (Classless Inter-Domain Routing)

Idea: use prefix matching to shorten the routing table

Recap - CIDR (Classless Inter-Domain Routing)

Idea: use prefix matching to shorten the routing table

Example routing Table

Address	Hop
128.18.10.18	A
128.221.207.217	A

Recap - CIDR (Classless Inter-Domain Routing)

Idea: use prefix matching to shorten the routing table

Example routing Table

Address	Hop
128.18.10.18	A
128.221.207.217	A

(128.18.10.18)
10000000.00010010.00001010.00010010

(128.221.207.217)
10000000.11011101.11001111.11011001

Recap - CIDR (Classless Inter-Domain Routing)

Idea: use prefix matching to shorten the routing table

Example routing Table

Address	Hop
128.18.10.18	A
128.221.207.217	A

Both has same destination

(128.18.10.18)
10000000.00010010.00001010.00010010

(128.221.207.217)
10000000.11011101.11001111.11011001

Recap - CIDR (Classless Inter-Domain Routing)

Idea: use prefix matching to shorten the routing table

Example routing Table

Address	Hop
128.18.10.18	A
128.221.207.217	A

(128.18.10.18)
10000000.00010010.00001010.00010010

(128.221.207.217)
10000000.11011101.11001111.11011001

First 8 bits are the same

Recap - CIDR (Classless Inter-Domain Routing)

Idea: use prefix matching to shorten the routing table

Example routing Table

Address	Hop
128.18.10.18	A
128.221.207.217	A

(128.18.10.18)
10000000.00010010.00001010.00010010

(128.221.207.217)
10000000.11011101.11001111.11011001

Aggregate

Address	Hop
128.0.0.0/8	A

Recap - CIDR (Classless Inter-Domain Routing)

Consider the following routing table:

Prefix	Port
137.70.160.0/22	A
137.70.164.0/22	A
137.70.168.0/22	B
137.70.172.0/22	A

Exercise - CIDR (Classless Inter-Domain Routing)

Consider the following routing table:

Prefix	Port
137.70.160.0/22	A
137.70.164.0/22	A
137.70.168.0/22	B
137.70.172.0/22	A

Can you use CIDR to shorten this routing table?

If so, what does the resulting table look like?

Solution - CIDR (Classless Inter-Domain Routing)

A: 137.70.160.0/22
10001001.01000110.10100000.00000000

A: 137.70.164.0/22
10001001.01000110.10100100.00000000

B: 137.70.168.0/22
10001001.01000110.10101000.00000000

A: 137.70.172.0/22
10001001.01000110.10101100.00000000

Solution - CIDR (Classless Inter-Domain Routing)

A: 137.70.160.0/22
10001001.01000110.10100000.00000000

A: 137.70.164.0/22
10001001.01000110.10100100.00000000

B: 137.70.168.0/22
10001001.01000110.10101000.00000000

A: 137.70.172.0/22
10001001.01000110.10101100.00000000

Solution - CIDR (Classless Inter-Domain Routing)

A: 137.70.160.0/22
10001001.01000110.10100000.00000000

A: 137.70.164.0/22
10001001.01000110.10100100.00000000

B: 137.70.168.0/22
10001001.01000110.10101000.00000000

A: 137.70.172.0/22
10001001.01000110.10101100.00000000

Solution - CIDR (Classless Inter-Domain Routing)

A: 137.70.160.0/22
10001001.01000110.10100000.00000000

A: 137.70.164.0/22
10001001.01000110.10100100.00000000

B: 137.70.168.0/22
10001001.01000110.10101000.00000000

A: 137.70.172.0/22
10001001.01000110.10101100.00000000

Prefix	Port
137.70.160.0/20	A

Solution - CIDR (Classless Inter-Domain Routing)

A: 137.70.160.0/22
10001001.01000110.10100000.00000000

A: 137.70.164.0/22
10001001.01000110.10100100.00000000

B: 137.70.168.0/22
10001001.01000110.10101000.00000000

A: 137.70.172.0/22
10001001.01000110.10101100.00000000

Prefix	Port
137.70.160.0/20	A

Solution - CIDR (Classless Inter-Domain Routing)

A: 137.70.160.0/22
10001001.01000110.10100000.00000000

A: 137.70.164.0/22
10001001.01000110.10100100.00000000

B: 137.70.168.0/22
10001001.01000110.10101000.00000000

A: 137.70.172.0/22
10001001.01000110.10101100.00000000

Prefix	Port
137.70.160.0/20	A

Solution - CIDR (Classless Inter-Domain Routing)

A: 137.70.160.0/22
10001001.01000110.10100000.00000000

A: 137.70.164.0/22
10001001.01000110.10100100.00000000

B: 137.70.168.0/22
10001001.01000110.10101000.00000000

A: 137.70.172.0/22
10001001.01000110.10101100.00000000

Prefix	Port
137.70.160.0/20	A
137.70.168.0/22	B

Recap - Token Bucket

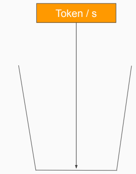
Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams

Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique

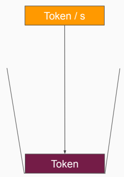
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



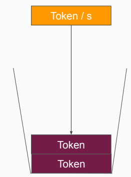
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



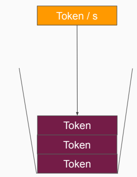
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



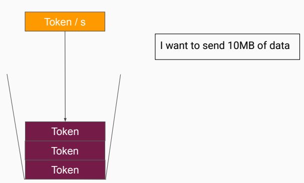
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



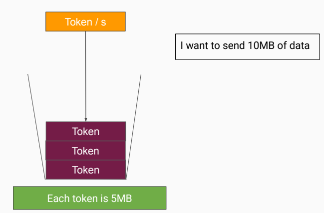
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



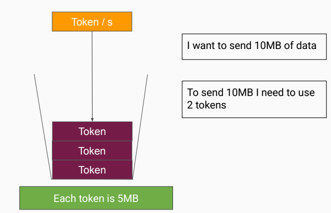
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



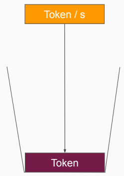
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



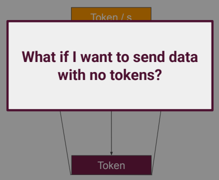
Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



Recap - Token Bucket

Traffic Shaping Idea: Turning unpridicatble bursts of data, into steady predictable streams
Token Bucket: Traffic Shaping Technique



Exercise - Token Bucket

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

Exercise - Token Bucket

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

1. For how long is the machine able to send data at a rate of **80MB/s**?
2. If the machine generates **4000 MB** in total, how long will it take to send all this data?

Solution - Token Bucket

Part 1:
Data Rate = **80MB/s**

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

1. For how long is the machine able to send data at a rate of **80MB/s**?

Solution - Token Bucket

Part 1:
Data Rate = **80MB/s**
Refil Rate = **10 Tokens/s (10MB/s)**

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

1. For how long is the machine able to send data at a rate of **80MB/s**?

Solution - Token Bucket

Part 1:
Data Rate = **80MB/s**
Refil Rate = **10 Tokens/s (10MB/s)**
Bucket = **560 Tokens**

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

1. For how long is the machine able to send data at a rate of **80MB/s**?

Solution - Token Bucket

Part 1:
Data Rate = **80MB/s**
Refil Rate = **10 Tokens/s (10MB/s)**
Bucket = **560 Tokens**

Every Second:
Bucket Looses **80MB/s**
Bucket Gains **10MB/s (10 tokens)**
Net Loss = **70MB/s**

$$\frac{560}{70} = 8s$$

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

1. For how long is the machine able to send data at a rate of **80MB/s**?

Solution - Token Bucket

Part 2:
It takes 8s to deplete the bucket

$$80_{MB/s} \times 8_s = 640_{MB}$$

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

2. If the machine generates **4000 MB** in total, how long will it take to send all this data?

Solution - Token Bucket

Part 2:
It takes 8s to deplete the bucket

$$80_{MB/s} \times 8_s = 640_{MB}$$

We still need to send

$$4000_{MB} - 640_{MB} = 3360_{MB}$$

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

2. If the machine generates **4000 MB** in total, how long will it take to send all this data?

Solution - Token Bucket

Part 2:
It takes 8s to deplete the bucket

$$80_{MB/s} \times 8_s = 640_{MB}$$

We still need to send

$$4000_{MB} - 640_{MB} = 3360_{MB}$$

We ran out of tokens. Limited to 10MB/s

$$\frac{3360}{10} = 336_s$$

A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

2. If the machine generates **4000 MB** in total, how long will it take to send all this data?

Solution - Token Bucket

Part 2:
It takes 8s to deplete the bucket

$$80_{MB/s} \times 8_s = 640_{MB}$$

We still need to send

$$4000_{MB} - 640_{MB} = 3360_{MB}$$

We ran out of tokens. Limited to 10MB/s

$$\frac{3360}{10} = 336_s \qquad \boxed{336 + 8 = 344_s}$$

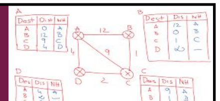
A user machine generates data at a rate of **80MB/s**. It uses a token bucket that currently contains **560 tokens** and has a capacity of **750 tokens**. The bucket refills at a rate of **10 tokens per second**. Assume that **1 token = 1 MB**.

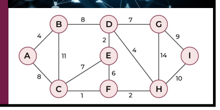
2. If the machine generates **4000 MB** in total, how long will it take to send all this data?

Summary

1. **Conceptual Overview**
   a. Distance Vector Routing
   b. Link State Routing
   c. Border Gateway Protocol

2. **IP Protocol**
   a. Hierarchical Routing
   b. Prefix Matching
   c. Traffic Shaping





Important Deadlines & Announcements

Self-Study Submission (PART 1) – 600 pts
- Deadline: Friday, 24 April, at 23:59

Self-Study PART 1 Interviews – Pass/Fail
- Date: Friday, 8 May 2026, from 15:30 until 18:30

Lab Assignments (Extra) – 3,700 pts
- Deadline: Tuesday, 12 May 2026, 12:45

Quiz Time

House Rules & Expectations

- Keep the Golden Exercise quiz facing down until told otherwise.
- The duration of the Golden Exercise quiz is 15 minutes.
- You are forbidden from using any devices during the entirety of the Golden Exercise quiz. Devices include but are not limited to mobile phones, personal computers, and calculators.
- You are forbidden from collaborating with your colleagues to solve the Golden Exercise quiz.
- Fill in your selected choice by filling in the boxes provided:
  - ■ **Good**
  - ☒ **Bad**
- The Golden Exercise quiz needs to be turned in to a Teaching Assistant at the end of the Tutorial.

If any of these expectations are not met, your submission will be discarded and you will not be awarded any points for this Golden Exercise quiz.