

## Chapter 5 Network Layer: Control Plane

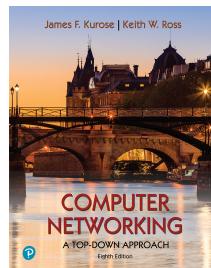
A note on the use of these PowerPoint slides:  
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KVR

All material copyright 1996-2020  
J.F.Kurose and K.W.Ross, All Rights Reserved



*Computer Networking: A  
Top-Down Approach*  
8<sup>th</sup> edition  
J. F. Kurose, K. W. Ross  
Pearson, 2020

1

## Network layer control plane: our goals

- understand principles behind network control plane:
  - traditional routing algorithms
  - SDN controllers
  - network management, configuration
- instantiation, implementation in the Internet:
  - OSPF, BGP
  - OpenFlow, ODL and ONOS controllers
  - Internet Control Message Protocol: ICMP
  - SNMP, YANG/NETCONF

Network Layer: 5-2

## Assignment Project Exam Help

2

<https://powcoder.com>

### Network layer: “control plane” roadmap

- introduction
- routing protocols
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

Network Layer: 5-3

3

## Add WeChat powcoder

### Network-layer functions

- **forwarding:** move packets from router's input to appropriate router output
- **routing:** determine route taken by packets from source to destination

*data plane*

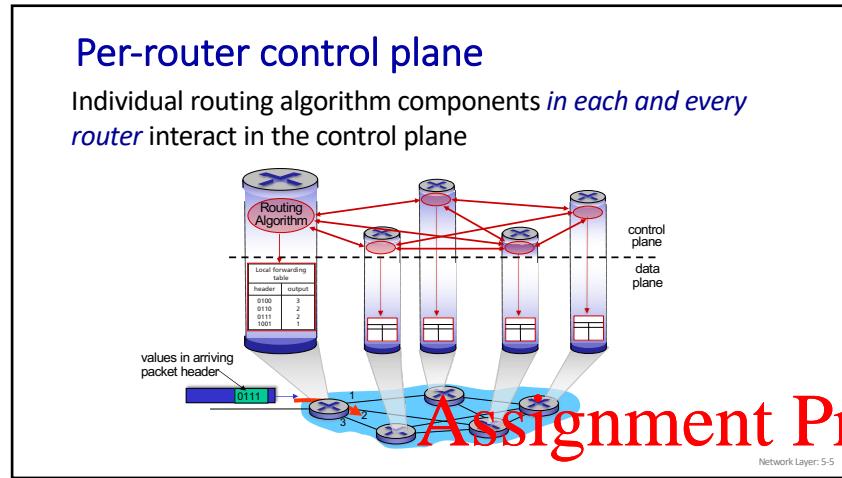
*control plane*

### Two approaches to structuring network control plane:

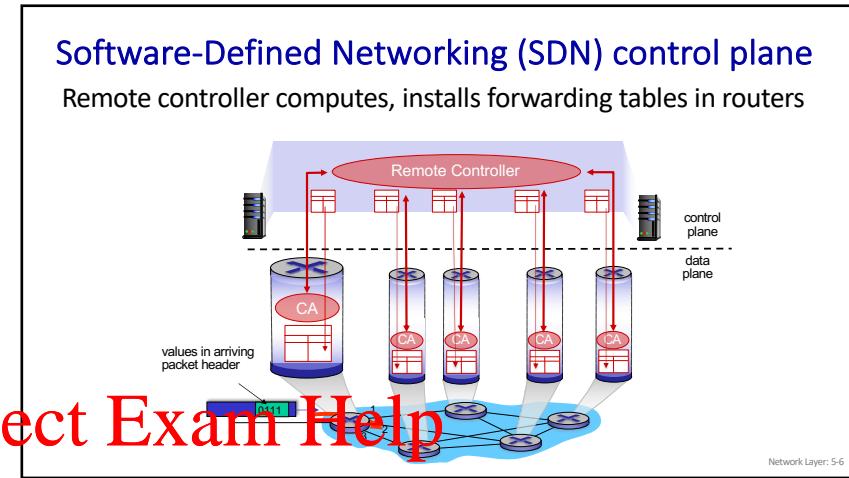
- per-router control (traditional)
- logically centralized control (software defined networking)

Network Layer: 5-4

4

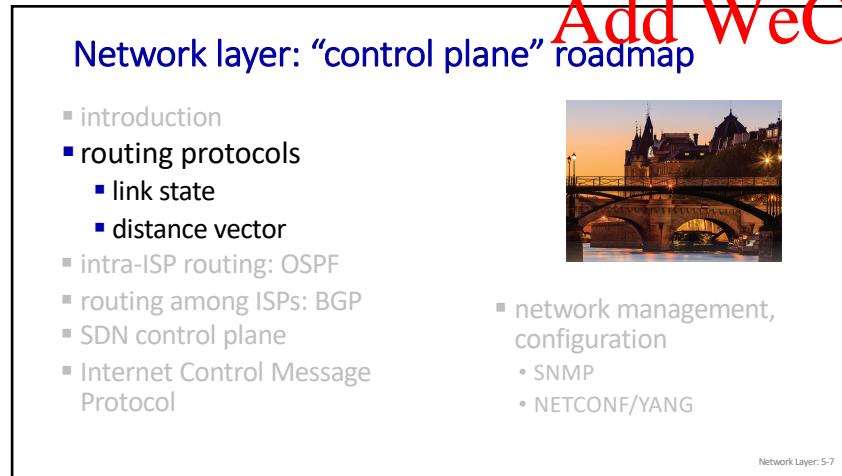


5

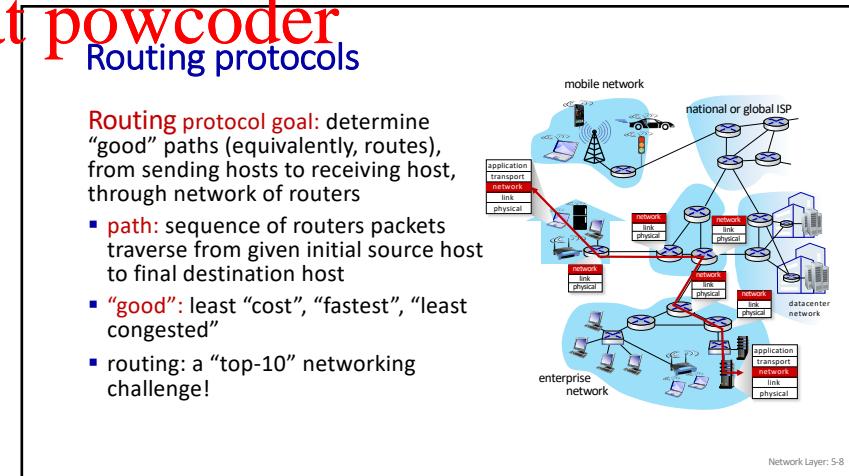


6

<https://powcoder.com>



7



8

### Graph abstraction: link costs

$c_{a,b}$ : cost of *direct* link connecting  $a$  and  $b$   
e.g.,  $c_{w,z} = 5$ ,  $c_{u,z} = \infty$

cost defined by network operator:  
could always be 1, or inversely related  
to bandwidth, or inversely related to  
congestion

graph:  $G = (N, E)$

$N$ : set of routers = { $u, v, w, x, y, z$ }

$E$ : set of links = { $(u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z)$ }

Network Layer: 5-9

9

### Routing algorithm classification

**global:** all routers have *complete* topology, link cost info  
• “link state” algorithms

**static:** routes change slowly over time

**dynamic:** routes change more quickly  
• periodic updates or in response to link cost changes

**decentralized:** iterative process of computation, exchange of info with neighbors  
• routers initially only know link costs to attached neighbors  
• “distance vector” algorithms

global or decentralized information?

Network Layer: 5-10

10

Assignment Project Exam Help

Add WeChat powcoder

### Network layer: “control plane” roadmap

- introduction
- routing protocols
  - **link state**
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol

- network management, configuration
  - SNMP
  - NETCONF/YANG

Network Layer: 5-11

11

### Dijkstra's link-state routing algorithm

- **centralized:** network topology, link costs known to *all* nodes
  - accomplished via “link state broadcast”
  - all nodes have same info
- computes least cost paths from one node (“source”) to all other nodes
  - gives *forwarding table* for that node
- **iterative:** after  $k$  iterations, know least cost path to  $k$  destinations

**notation**

- $c_{x,y}$ : *direct* link cost from node  $x$  to  $y$ ;  $= \infty$  if not direct neighbors
- $D(v)$ : *current* estimate of cost of least-cost-path from source to destination  $v$
- $p(v)$ : predecessor node along path from source to  $v$
- $N'$ : set of nodes whose least-cost-path *definitively* known

Network Layer: 5-12

12

## Dijkstra's link-state routing algorithm

```

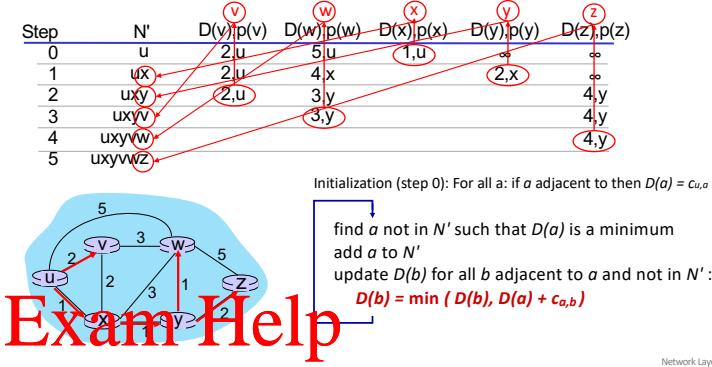
1 Initialization:
2  $N' = \{u\}$  /* compute least cost path from u to all other nodes */
3 for all nodes v
4 if v adjacent to u /* u initially knows direct-path-cost only to direct neighbors */
5 then  $D(v) = c_{u,v}$  /* but may not be minimum cost!
6 else  $D(v) = \infty$ 
7

8 Loop
9 find w not in  $N'$  such that  $D(w)$  is a minimum
10 add w to  $N'$ 
11 update  $D(v)$  for all v adjacent to w and not in  $N'$ :
12  $D(v) = \min(D(v), D(w) + c_{w,v})$ 
13 /* new least-path-cost to v is either old least-cost-path to v or known
least-cost-path to w plus direct cost from w to v */
14
15 until all nodes in  $N'$ 
```

Network Layer: 5-13

13

## Dijkstra's algorithm: an example

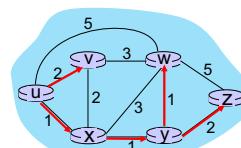


14

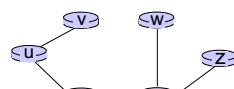
Assignment Project Exam Help

<https://powcoder.com>

## Dijkstra's algorithm: an example



resulting least-cost-path tree from u:



resulting forwarding table in u:

destination	outgoing link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

route from u to v directly

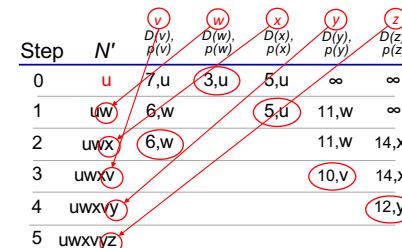
route from u to all other destinations via x

Network Layer: 5-15

15

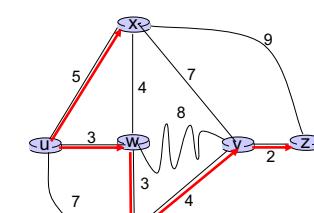
## Add WeChat powcoder

## Dijkstra's algorithm: another example



### notes:

- construct least-cost-path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)



Network Layer: 5-16

16

## Dijkstra's algorithm: discussion

### algorithm complexity: $n$ nodes

- each of  $n$  iteration: need to check all nodes,  $w$ , not in  $N$
- $n(n+1)/2$  comparisons:  $O(n^2)$  complexity
- more efficient implementations possible:  $O(n \log n)$

### message complexity:

- each router must *broadcast* its link state information to other  $n$  routers
- efficient (and interesting!) broadcast algorithms:  $O(n)$  link crossings to disseminate a broadcast message from one source
- each router's message crosses  $O(n)$  links: overall message complexity:  $O(n^2)$

# Assignment Project Exam Help

Network Layer: 5-17

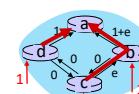
17

## Dijkstra's algorithm: oscillations possible

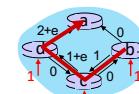
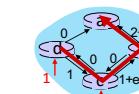
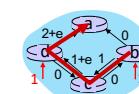
- when link costs depend on traffic volume, **route oscillations** possible

- sample scenario:

- routing to destination  $a$ , traffic entering at  $d, c, e$  with rates  $1, e < 1, 1$
- link costs are directional, and volume-dependent



initially

given these costs,  
find new routing....  
resulting in new costgiven these costs,  
find new routing....  
resulting in new costsgiven these costs,  
find new routing....  
resulting in new costs

Network Layer: 5-18

18

<https://powcoder.com>

## Network layer: “control plane” roadmap

- introduction
- routing protocols**
  - link state
  - distance vector**
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

Network Layer: 5-19

19

## Add WeChat powcoder

## Distance vector algorithm

Based on *Bellman-Ford* (BF) equation (dynamic programming):

Bellman-Ford equation

Let  $D_x(y)$ : cost of least-cost path from  $x$  to  $y$ .  
Then:

$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

$\min$  taken over all neighbors  $v$  of  $x$

$v$ 's estimated least-cost-path cost to  $y$   
direct cost of link from  $x$  to  $v$

Network Layer: 5-20

20

### Bellman-Ford Example

Suppose that  $u$ 's neighboring nodes,  $x, v, w$ , know that for destination  $z$ :

$D_u(z) = \min \{ c_{u,v} + D_v(z), c_{u,x} + D_x(z), c_{u,w} + D_w(z) \}$

$$= \min \{ 2 + 5, 1 + 3, 5 + 3 \} = 4$$

node achieving minimum (x) is next hop on estimated least-cost path to destination (z)

### Distance vector algorithm

**key idea:**

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when  $x$  receives new DV estimate from any neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v \{ c_{x,v} + D_v(y) \} \text{ for each node } y \in N$$

- under minor, natural conditions, the estimate  $D_x(y)$  converge to the actual least cost  $d_x(y)$

Assignment Project Exam Help

Network Layer: 5-21

Network Layer: 5-22

21

22

<https://powcoder.com>

### Distance vector algorithm:

**each node:**

```

    wait for (change in local link
    cost or msg from neighbor)
    recomputes DV estimates using
    DV received from neighbor
    if DV to any destination has
    changed, notify neighbors
  
```

**iterative, asynchronous:** each local iteration caused by:

- local link cost change
- DV update message from neighbor

**distributed, self-stopping:** each node notifies neighbors *only* when its DV changes

- neighbors then notify their neighbors – *only if necessary*
- no notification received, no actions taken!

### Distance vector. example

**t=0**

DV in a:
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

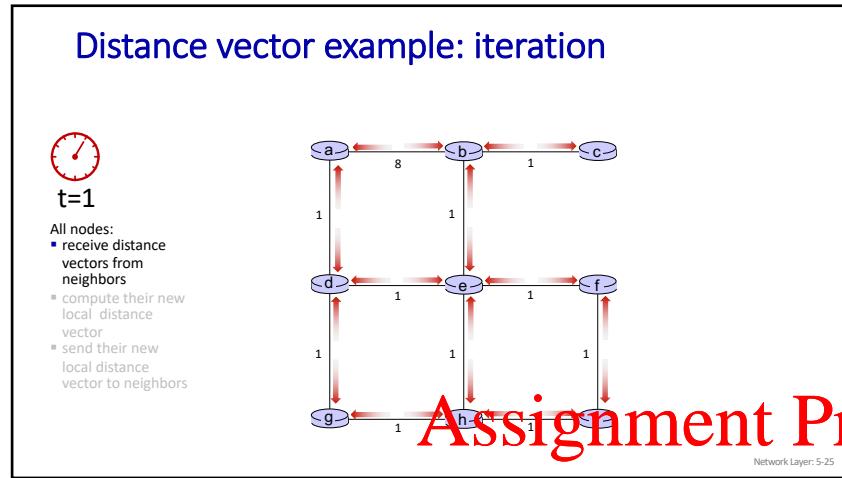
A few asymmetries:  
 - missing link  
 - larger cost

Network Layer: 5-23

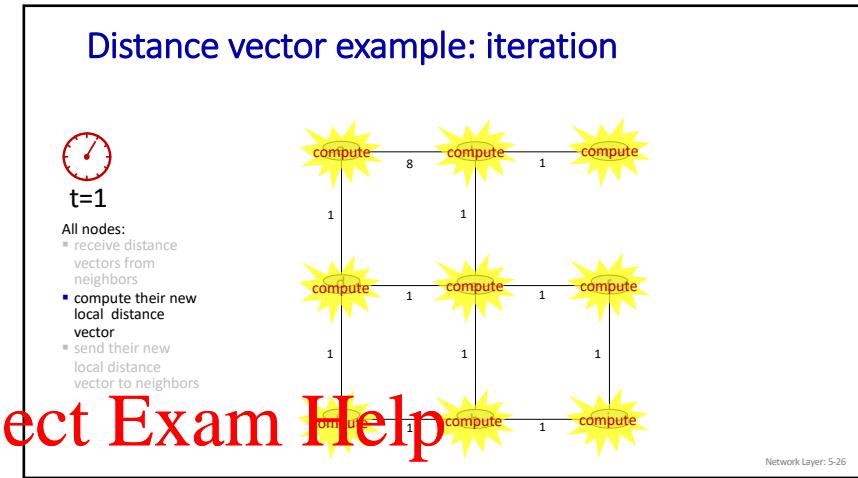
Network Layer: 5-24

23

24

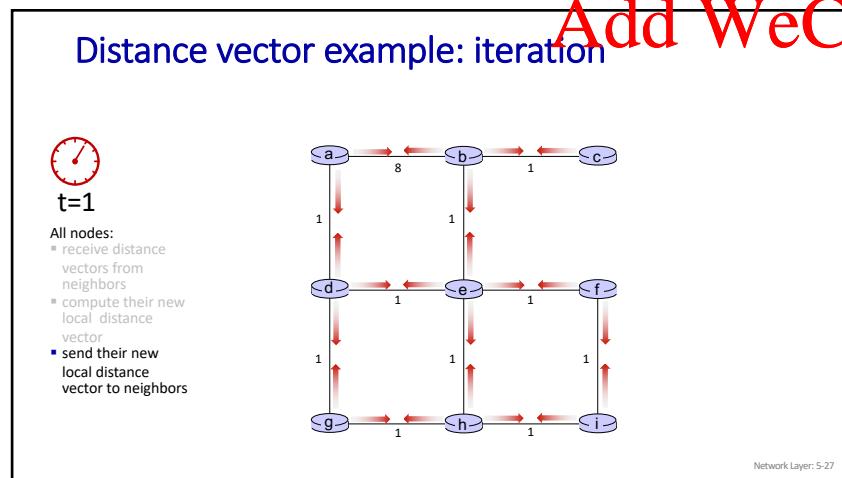


25

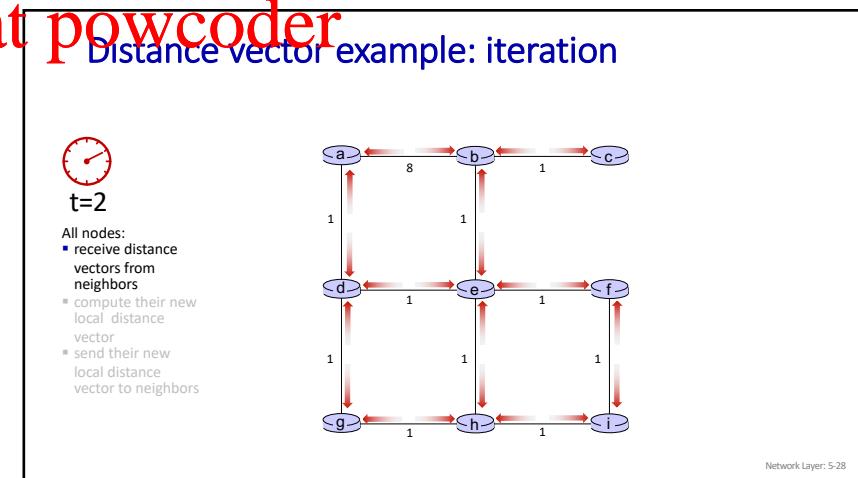


26

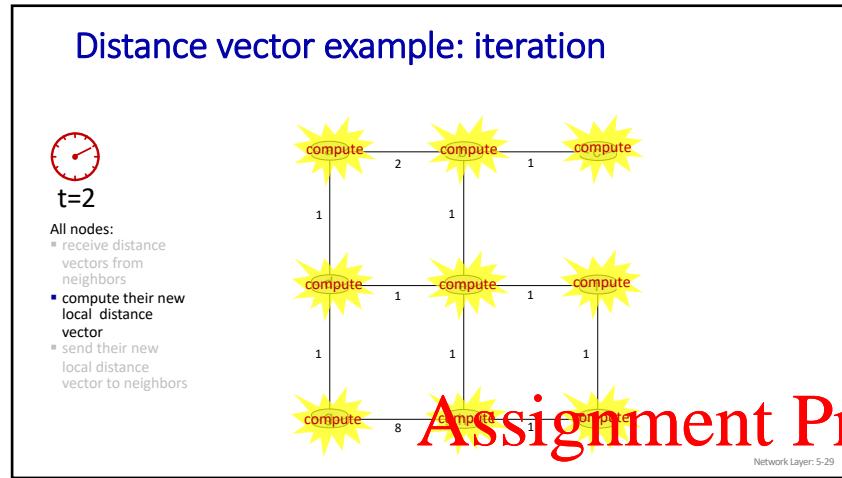
Assignment Project Exam Help  
<https://powcoder.com>



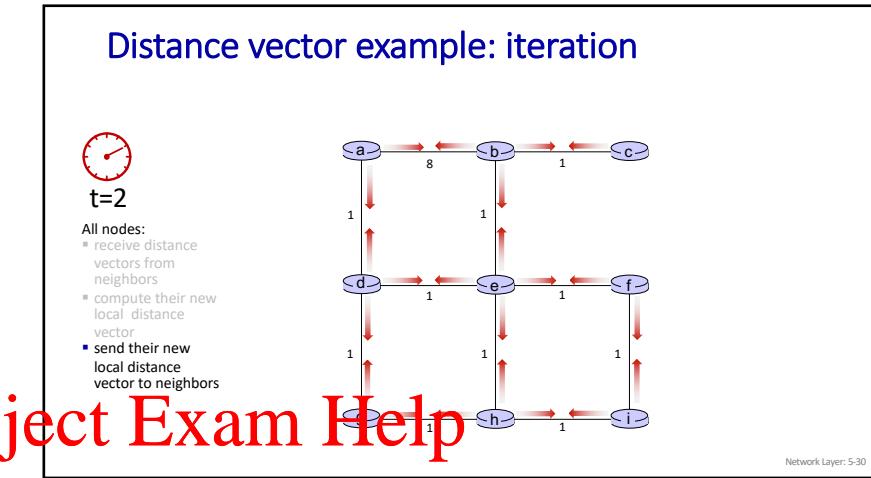
27



28

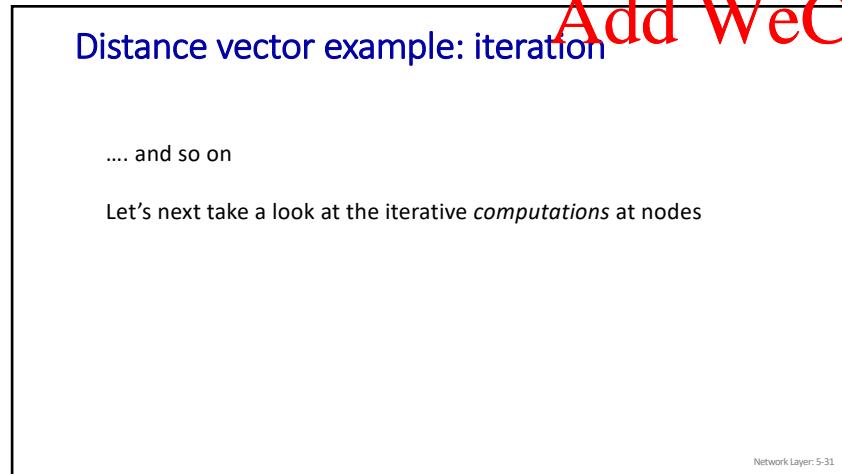


29

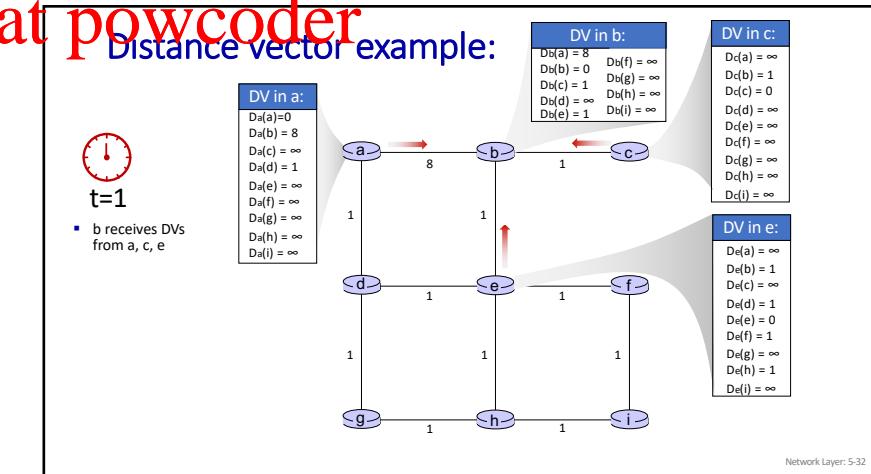


30

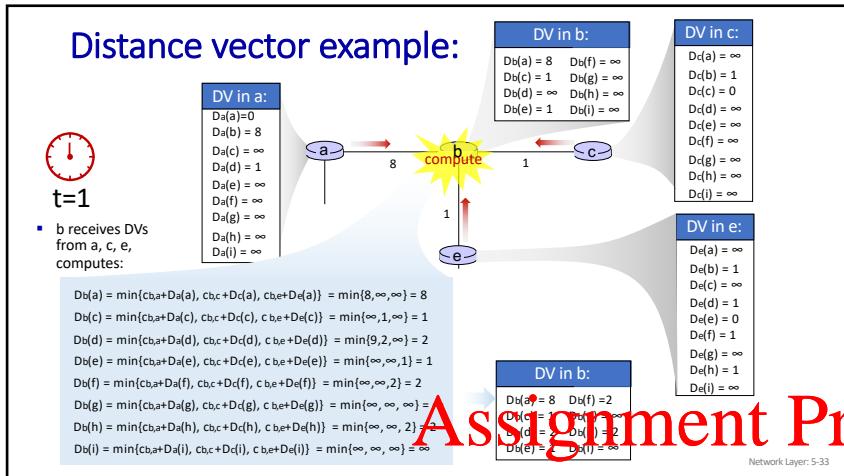
Assignment Project Exam Help  
<https://powcoder.com>



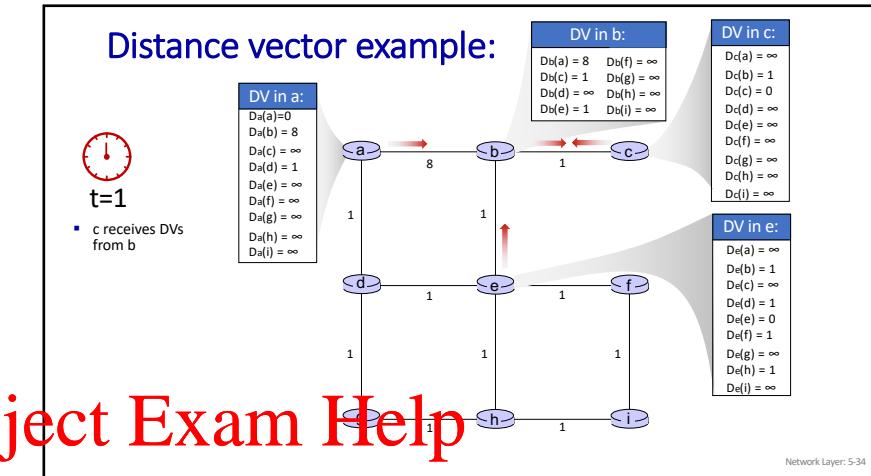
31



32



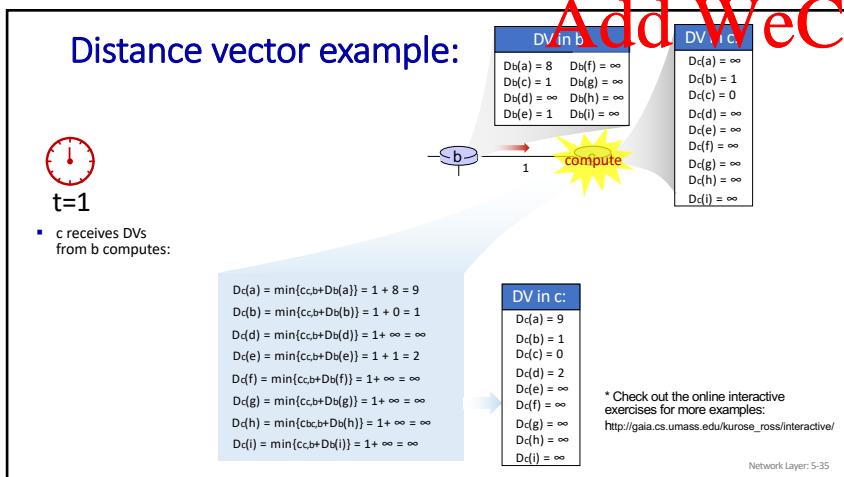
33



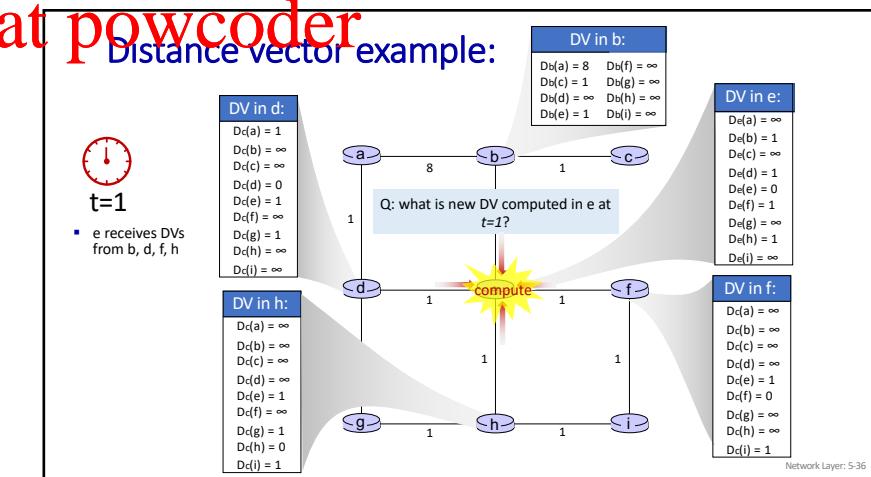
34

# Assignment Project Exam Help

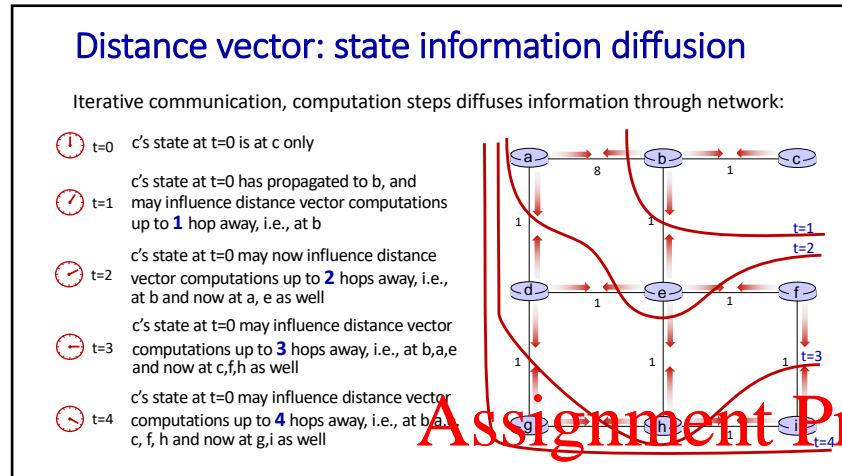
<https://powcoder.com>



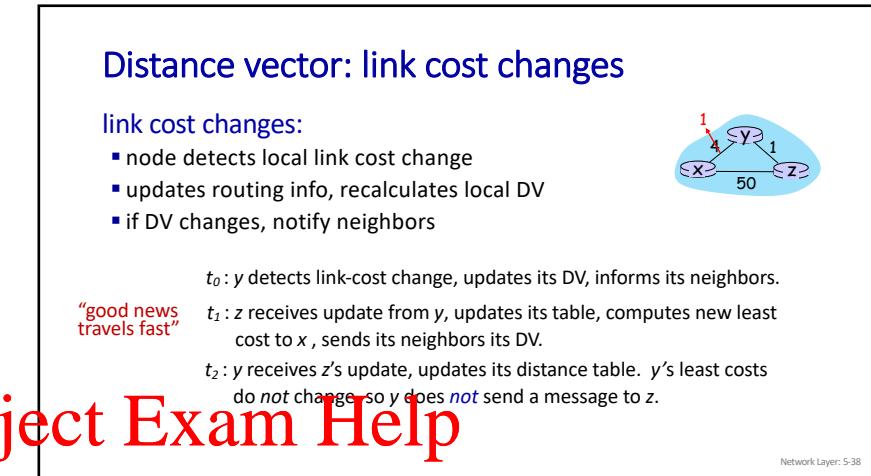
35



36



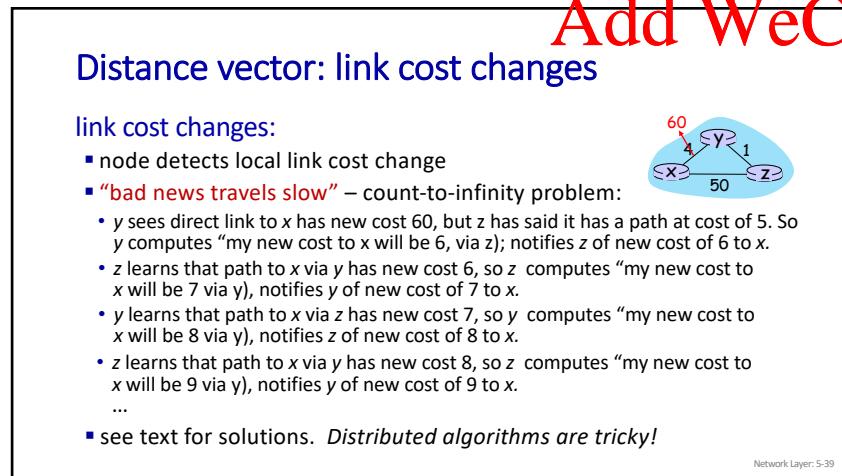
37



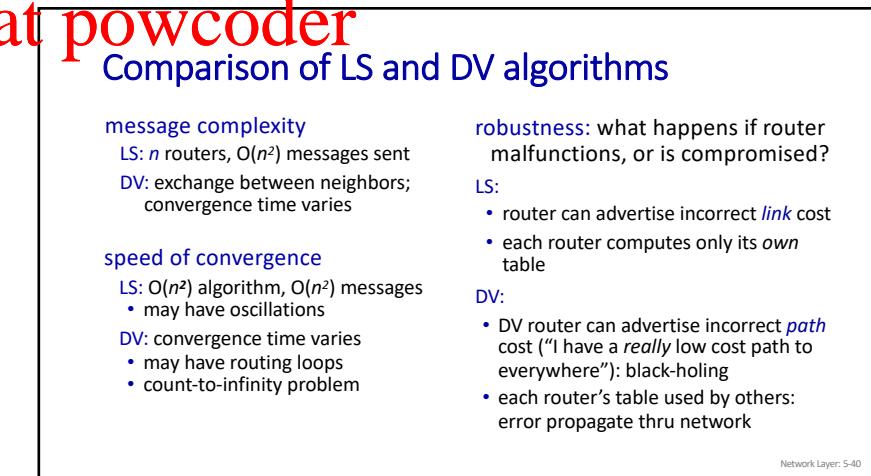
38

Assignment Project Exam Help

<https://powcoder.com>



39



40

## Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

**Assignment Project Exam Help**

Network Layer: 5-41

41

## Making routing scalable

our routing study thus far - idealized

- all routers identical
- network “flat”
- ... not true in practice

**scale:** billions of destinations:

- can't store all destinations in routing tables!
- routing table exchange would swamp links!

**administrative autonomy:**

- Internet: a network of networks
- each network admin may want to control routing in its own network

Network Layer: 5-42

42

<https://powcoder.com>

## Add WeChat powcoder

### Internet approach to scalable routing

aggregate routers into regions known as “autonomous systems” (AS) (a.k.a. “domains”)

**intra-AS (aka “intra-domain”):** routing among *within same AS (“network”)*

- all routers in AS must run same intra-domain protocol
- routers in different AS can run different intra-domain routing protocols
- **gateway router:** at “edge” of its own AS, has link(s) to router(s) in other AS'es

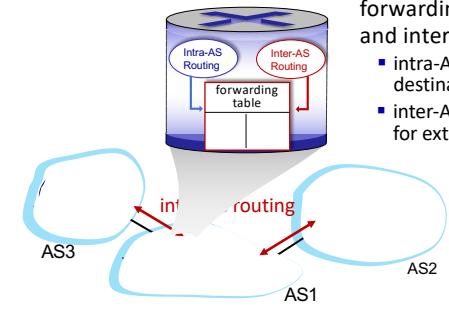
**inter-AS (aka “inter-domain”):** routing *among AS'es*

- gateways perform inter-domain routing (as well as intra-domain routing)

Network Layer: 5-43

43

### Interconnected ASes



forwarding table configured by intra- and inter-AS routing algorithms

- intra-AS routing determine entries for destinations within AS
- inter-AS & intra-AS determine entries for external destinations

Network Layer: 5-44

44

**Inter-AS routing: a role in intradomain forwarding**

- suppose router in AS1 receives datagram destined outside of AS1:
- router should forward packet to gateway router in AS1, but which one?

**AS1 inter-domain routing must:**

- learn which destinations reachable through AS2, which through AS3
- propagate this reachability info to all routers in AS1

Network Layer: 5-45

45

**Inter-AS routing: routing within an AS**

most common intra-AS routing protocols:

- RIP: Routing Information Protocol [RFC 1723]**
  - classic DV: DVs exchanged every 30 secs
  - no longer widely used
- EIGRP: Enhanced Interior Gateway Routing Protocol**
  - DV based
  - formerly Cisco-proprietary for decades (became open in 2013 [RFC 7868])
- OSPF: Open Shortest Path First [RFC 2328]**
  - link-state routing
  - IS-IS protocol (ISO standard, not RFC standard) essentially same as OSPF

Network Layer: 5-46

46

Assignment Project Exam Help  
<https://powcoder.com>

**OSPF (Open Shortest Path First) routing**

- “open”: publicly available
- classic link-state
  - each router floods OSPF link-state advertisements (directly over IP rather than using TCP/UDP) to all other routers in entire AS
  - multiple link costs metrics possible: bandwidth, delay
  - each router has full topology, uses Dijkstra’s algorithm to compute forwarding table
- security:** all OSPF messages authenticated (to prevent malicious intrusion)

Network Layer: 5-47

47

**Hierarchical OSPF**

- two-level hierarchy:** local area, backbone
  - link-state advertisements flooded only in area, or backbone
  - each node has detailed area topology; only knows direction to reach other destinations

Network Layer: 5-48

48

## Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP**
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

**Assignment Project Exam Help**

Network Layer: 5-49

49

## Internet inter-AS routing: BGP

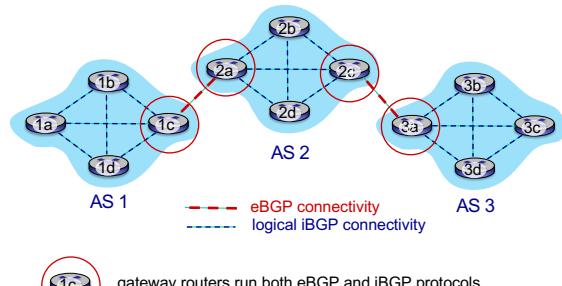
- BGP (Border Gateway Protocol):** *the de facto inter-domain routing protocol*
  - “glue that holds the Internet together”
- allows subnet to advertise its existence, and the destinations it can reach, to rest of Internet: *“I am here, here is who I can reach, and how”*
- BGP provides each AS a means to:
  - **eBGP:** obtain subnet reachability information from neighboring ASes
  - **iBGP:** propagate reachability information to all AS-internal routers.
  - determine “good” routes to other networks based on reachability information and *policy*

Network Layer: 5-50

50

https://powcoder.com

## eBGP, iBGP connections



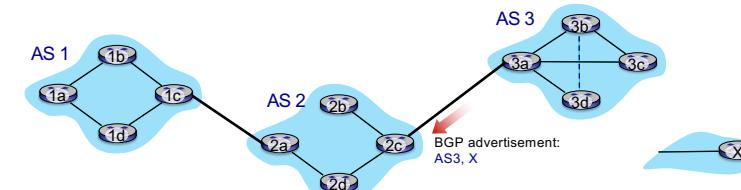
Network Layer: 5-51

51

## Add WeChat powcoder

### BGP basics

- BGP session:** two BGP routers (“peers”) exchange BGP messages over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a “path vector” protocol)
- when AS3 gateway 3a advertises path AS3,X to AS2 gateway 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X



Network Layer: 5-52

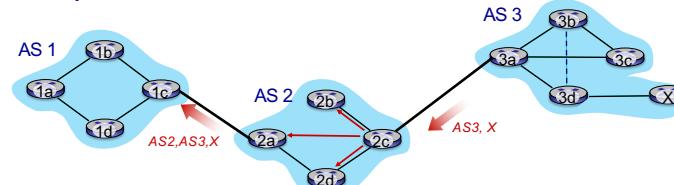
52

## Path attributes and BGP routes

- BGP advertised route: prefix + attributes
  - prefix: destination being advertised
  - two important attributes:
    - **AS-PATH**: list of ASes through which prefix advertisement has passed
    - **NEXT-HOP**: indicates specific internal-AS router to next-hop AS
- **policy-based routing**:
  - gateway receiving route advertisement uses *import policy* to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to *advertise* path to other other neighboring ASes

53

## BGP path advertisement



- AS2 router 2c receives path advertisement **AS3,X** (via eBGP) from AS3 router 3a
- based on AS2 policy, AS2 router 2c accepts path **AS3,X**, propagates (via iBGP) to all AS2 routers
- based on AS2 policy, AS2 router 2a advertises (via eBGP) path **AS2, AS3, X** to AS1 router 1c

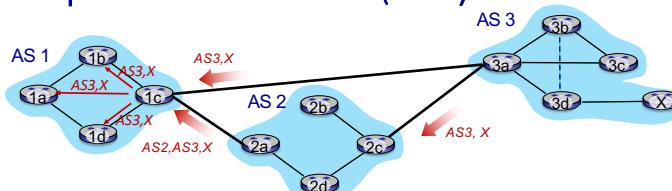
Network Layer: 5-54

54

Assignment Project Exam Help

<https://powcoder.com>

## BGP path advertisement (more)



gateway router may learn about **multiple** paths to destination:

- AS1 gateway router 1c learns path **AS2,AS3,X** from 2a
- AS1 gateway router 1c learns path **AS3,X** from 3a
- based on *policy*, AS1 gateway router 1c chooses path **AS3,X** and advertises path within AS1 via iBGP

55

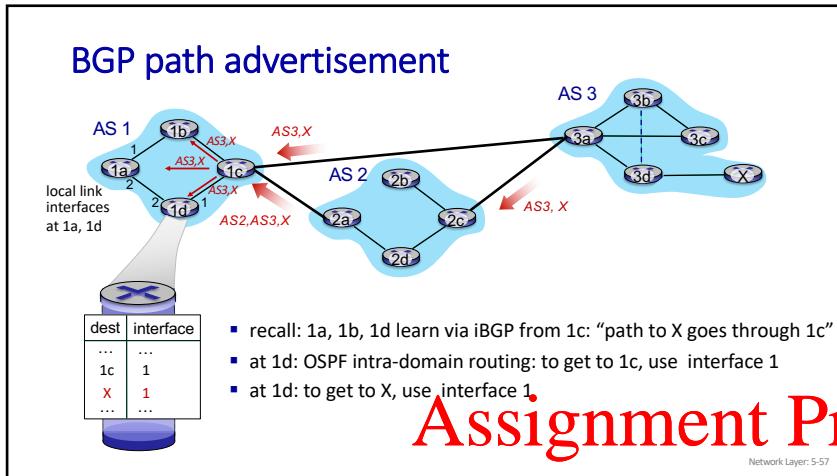
## Add WeChat powcoder

## BGP messages

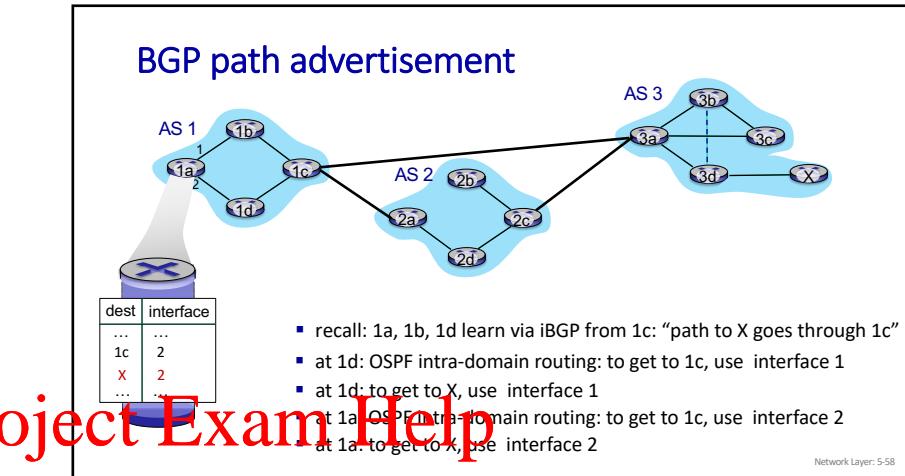
- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - **OPEN**: opens TCP connection to remote BGP peer and authenticates sending BGP peer
  - **UPDATE**: advertises new path (or withdraws old)
  - **KEEPALIVE**: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - **NOTIFICATION**: reports errors in previous msg; also used to close connection

Network Layer: 5-56

56

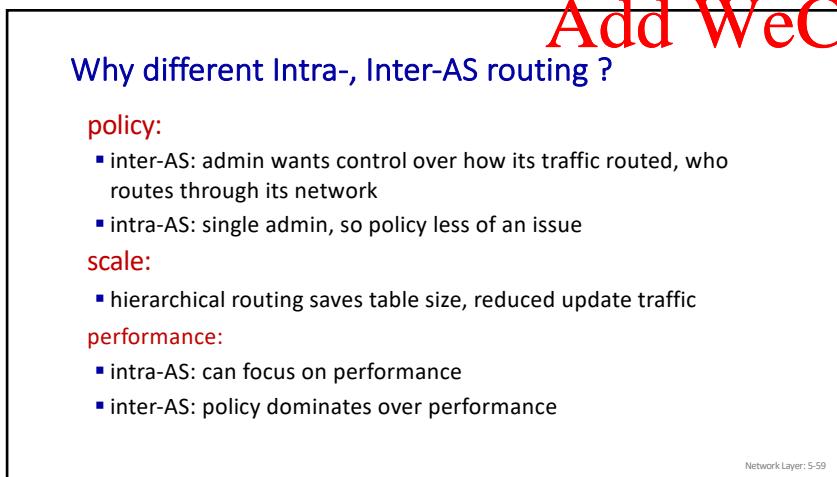


57

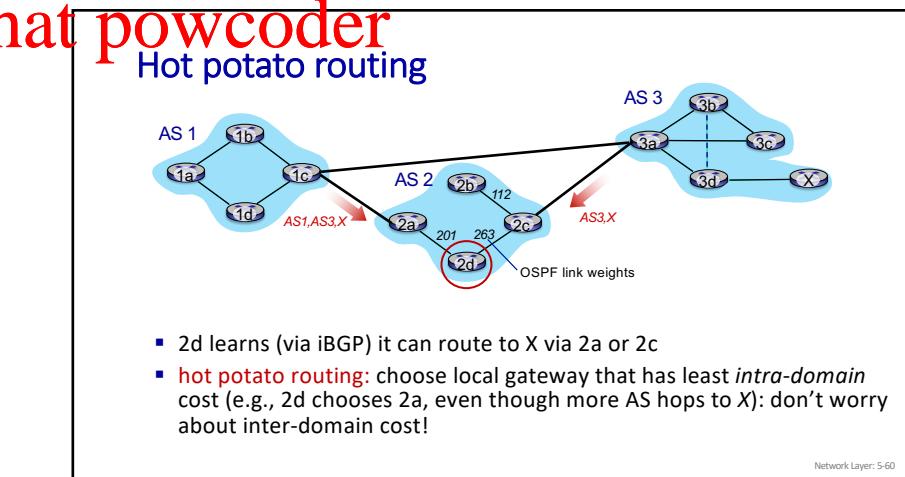


58

# Assignment Project Exam Help



59



### BGP: achieving policy via advertisements

ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs – a typical “real world” policy)

- A advertises path Aw to B and to C
- B *chooses not to advertise* BAw to C!
- B gets no “revenue” for routing CBAw, since none of C, A, w are B’s customers
- C does *not* learn about CBAw path
- C will route CAw (not using B) to get to w

Network Layer: 5-61

61

### BGP: achieving policy via advertisements (more)

ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs – a typical “real world” policy)

- A,B,C are **provider networks**
- x,w,y are **customer** (of provider networks)
- x is **dual-homed**: attached to two networks
- *policy to enforce* x does not want to route from B to C via x
- e.g. x will not advertise to B a route to C

Network Layer: 5-62

62

# Assignment Project Exam Help

### BGP route selection

- router may learn about more than one route to destination AS, selects route based on:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. closest NEXT-HOP router: hot potato routing
  4. additional criteria

Network Layer: 5-63

63

# Add WeChat powcoder

### Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- **SDN control plane**
- Internet Control Message Protocol
- network management, configuration
  - SNMP
  - NETCONF/YANG

Network Layer: 5-64

## Software defined networking (SDN)

- Internet network layer: historically implemented via distributed, per-router control approach:
  - monolithic* router contains switching hardware, runs proprietary implementation of Internet standard protocols (IP, RIP, IS-IS, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)
  - different “middleboxes” for different network layer functions: firewalls, load balancers, NAT boxes, ..
- ~2005: renewed interest in rethinking network control plane

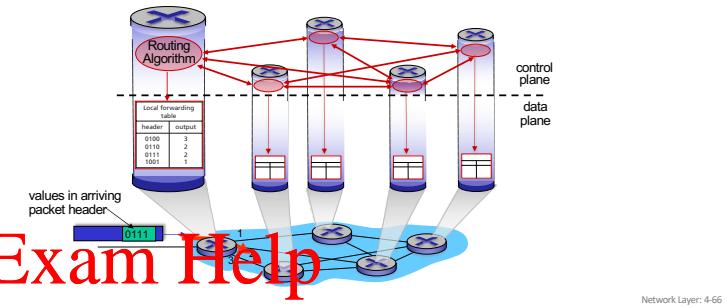
Assignment Project Exam Help

Network Layer: 5-65

65

## Per-router control plane

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



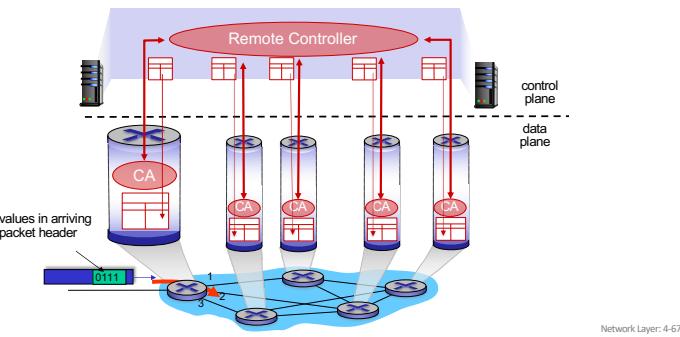
Network Layer: 4-66

66

<https://powcoder.com>

## Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



Network Layer: 4-67

67

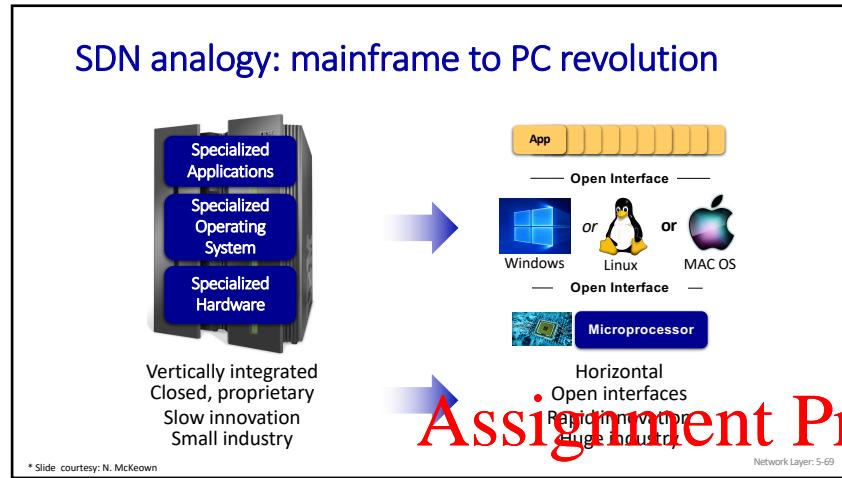
## Software defined networking (SDN)

Why a *logically centralized* control plane?

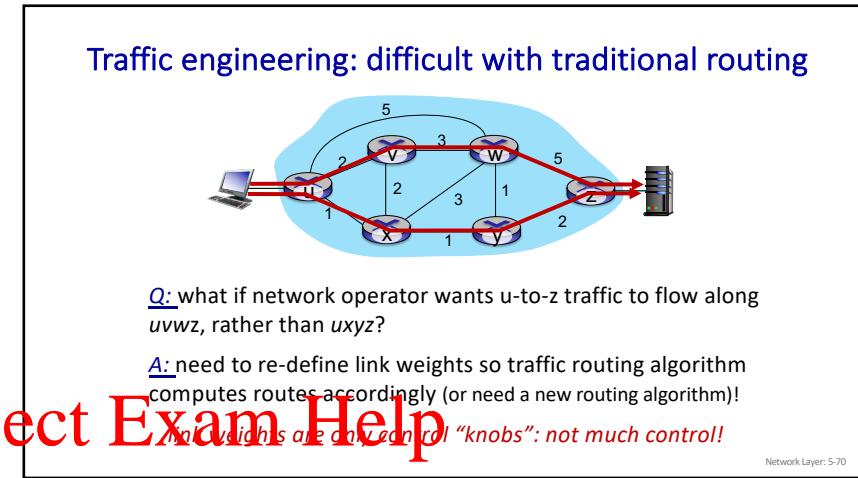
- easier network management: avoid router misconfigurations, greater flexibility of traffic flows
- table-based forwarding (recall OpenFlow API) allows “programming” routers
  - centralized “programming” easier: compute tables centrally and distribute
  - distributed “programming” more difficult: compute tables as result of distributed algorithm (protocol) implemented in each-and-every router
- open (non-proprietary) implementation of control plane
  - foster innovation: let 1000 flowers bloom

Network Layer: 5-68

68

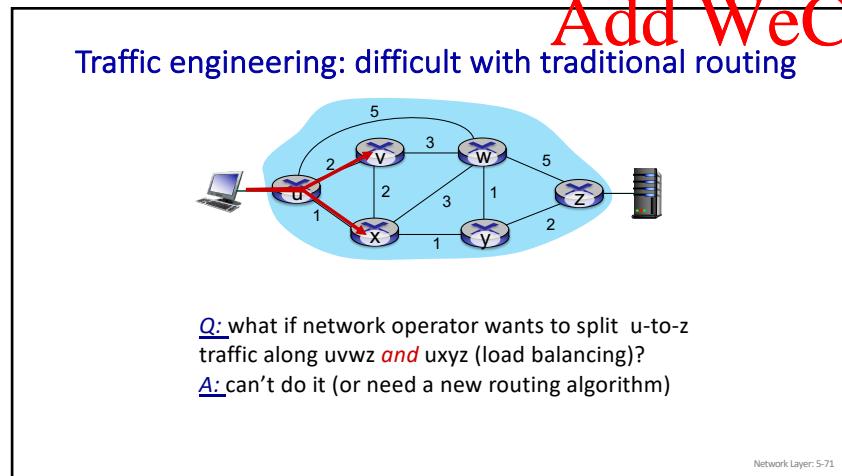


69

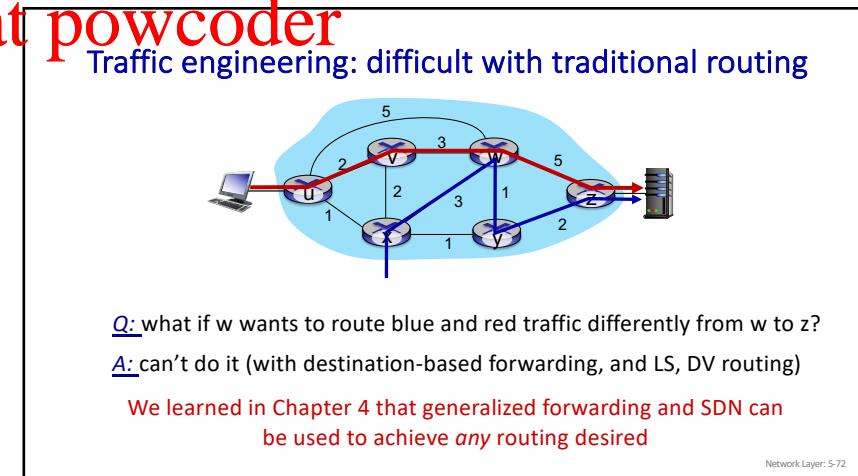


70

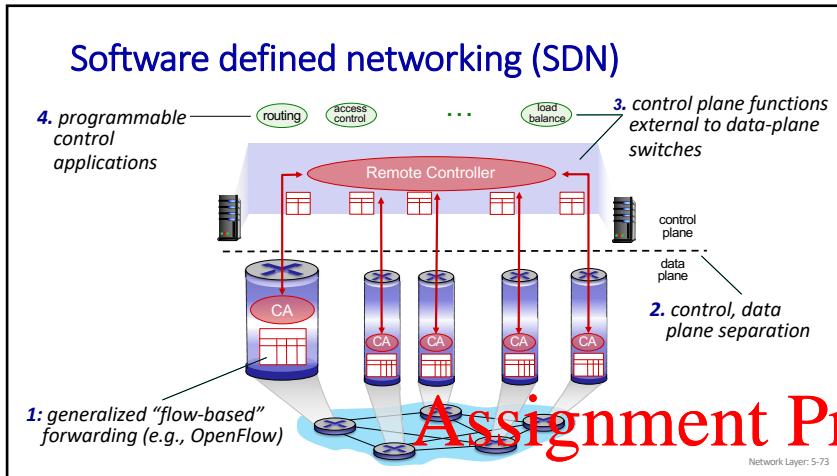
<https://powcoder.com>



71



72

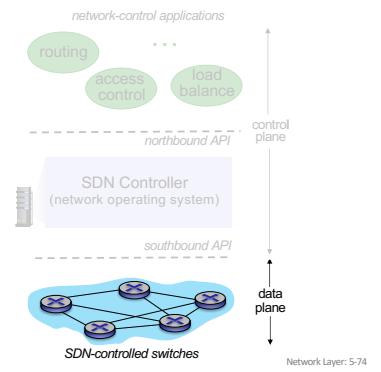


73

## Software defined networking (SDN)

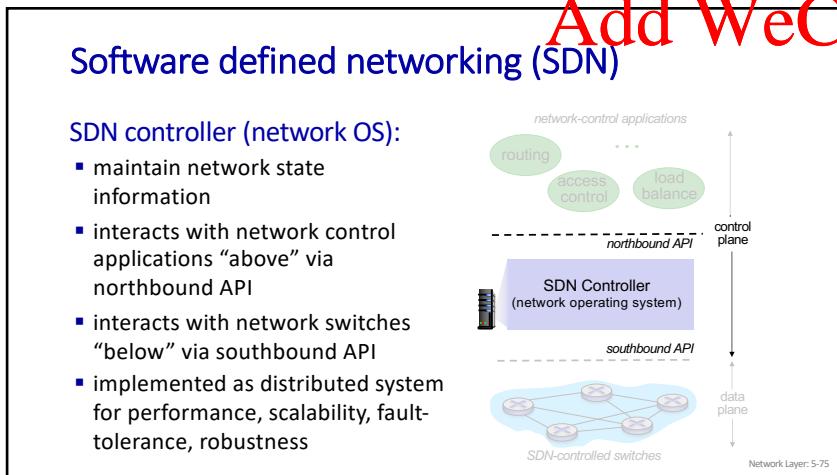
### Data-plane switches:

- fast, simple, commodity switches implementing generalized data-plane forwarding (Section 4.4) in hardware
- flow (forwarding) table computed, installed under controller supervision
- API for table-based switch control (e.g., OpenFlow)
  - defines what is controllable, what is not
- protocol for communicating with controller (e.g., OpenFlow)



74

<https://powcoder.com>



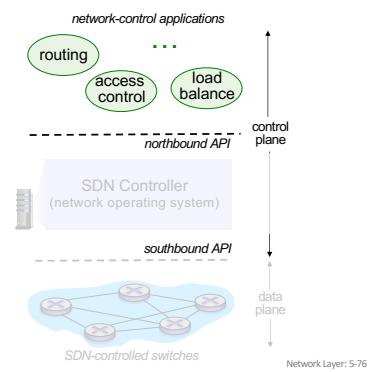
75

Add WeChat powcoder

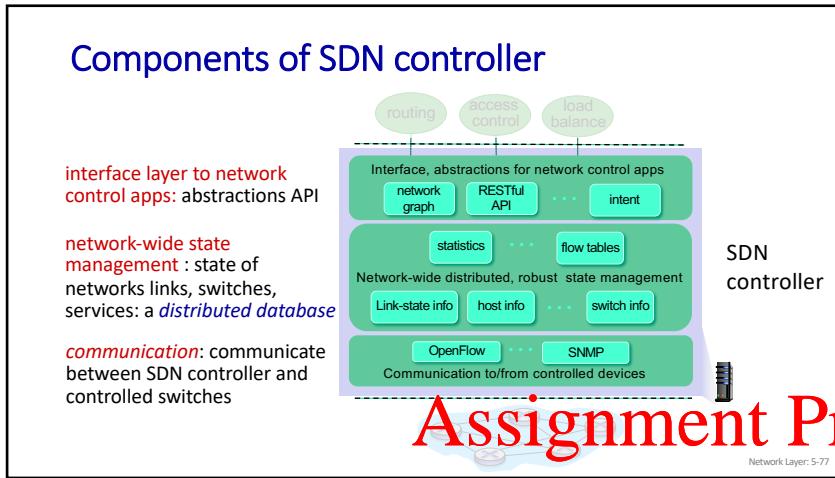
## Software defined networking (SDN)

### network-control apps:

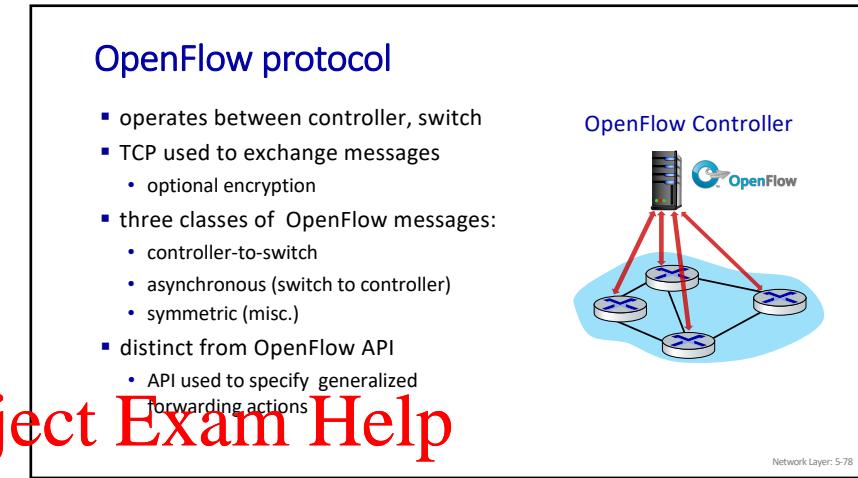
- "brains" of control: implement control functions using lower-level services, API provided by SDN controller
- *unbundled*: can be provided by 3<sup>rd</sup> party: distinct from routing vendor, or SDN controller



76

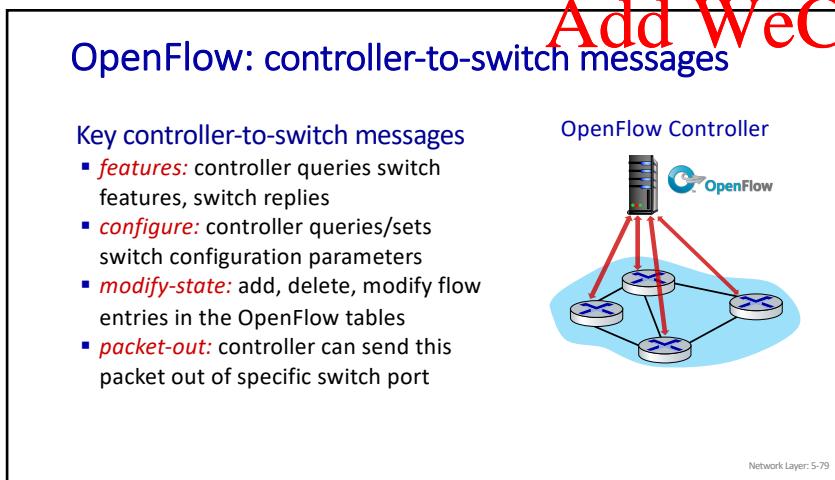


77

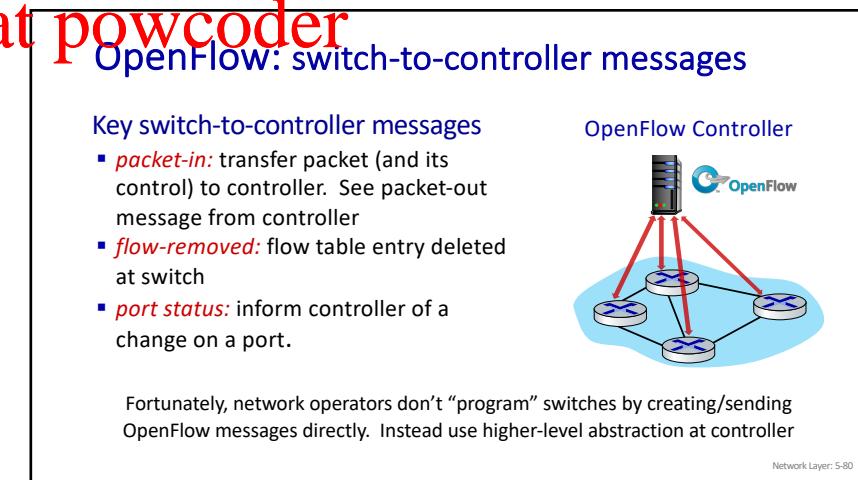


78

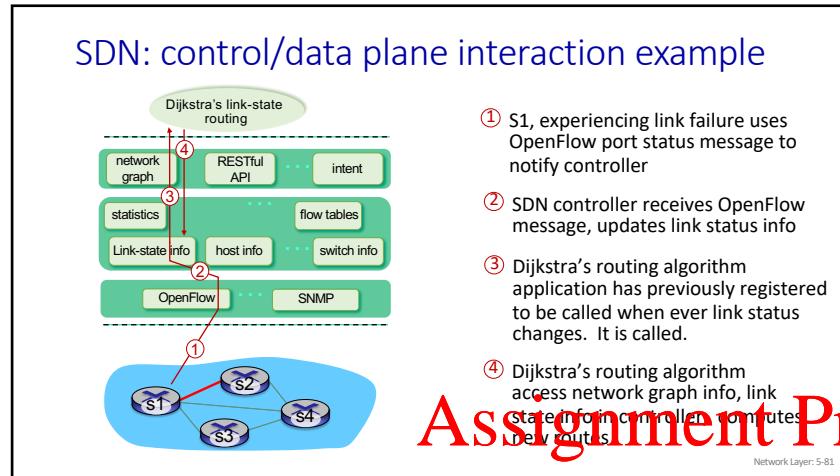
<https://powcoder.com>



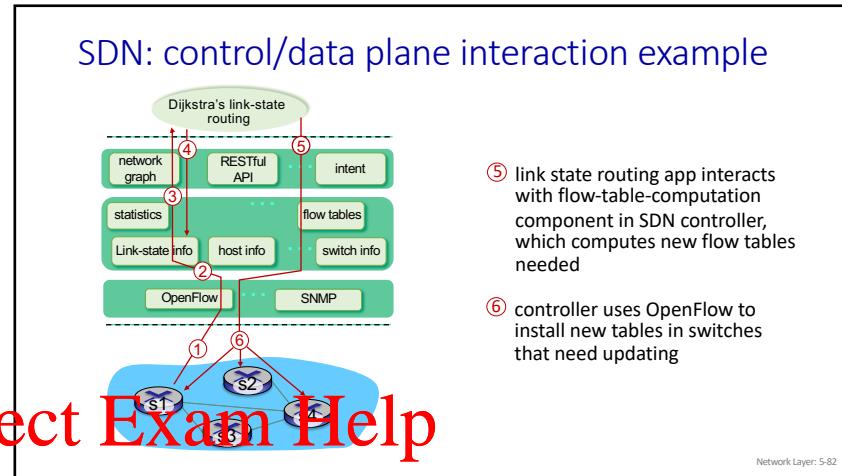
79



80



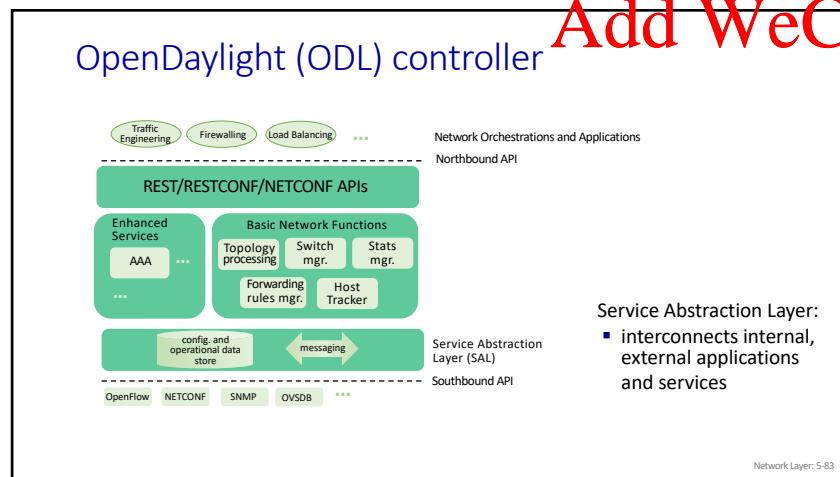
81



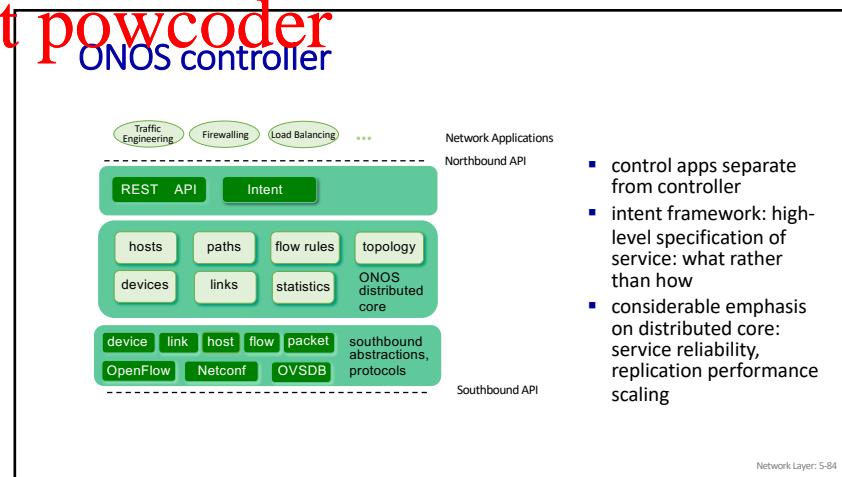
82

# Assignment Project Exam Help

<https://powcoder.com>



83



84

## SDN: selected challenges

- hardening the control plane: dependable, reliable, performance-scalable, secure distributed system
  - robustness to failures: leverage strong theory of reliable distributed system for control plane
  - dependability, security: “baked in” from day one?
- networks, protocols meeting mission-specific requirements
  - e.g., real-time, ultra-reliable, ultra-secure
- Internet-scaling: beyond a single AS
- SDN critical in 5G cellular networks

Network Layer: 5-85

85

# Assignment Project Exam Help

How will implementation of network functionality (SDN vs. protocols) evolve?



Network Layer: 5-86

86

<https://powcoder.com>

## Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- **Internet Control Message Protocol**



- network management, configuration
  - SNMP
  - NETCONF/YANG

Network Layer: 5-87

87

# Add WeChat powcoder

## ICMP: internet control message protocol

- used by hosts and routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer “above” IP:
  - ICMP messages carried in IP datagrams
- **ICMP message:** type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Network Layer: 4-88

88

## Traceroute and ICMP

**source sends sets of UDP segments to destination**

- 1<sup>st</sup> set has TTL =1, 2<sup>nd</sup> set has TTL=2, etc.

**datagram in *n*th set arrives to *n*th router:**

- router discards datagram and sends source ICMP message (type 11, code 0)
- ICMP message possibly includes name of router & IP address

**when ICMP message arrives at source, record RTT**

**stopping criteria:**

- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops

Network Layer: 4-89

89

## Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol
- network management, configuration
  - SNMP
  - NETCONF/YANG

Network Layer: 5-90

90

# Assignment Project Exam Help

## What is network management?

- autonomous systems (aka “network”): 1000s of interacting hardware/software components
- other complex systems requiring monitoring, configuration, control:
  - jet airplane, nuclear power plant, others?

**“Network management includes the deployment, integration and coordination of the hardware, software, and human elements to monitor, test, poll, configure, analyze, evaluate, and control the network and element resources to meet the real-time, operational performance, and Quality of Service requirements at a reasonable cost.”**

Network Layer: 5-91

91

## Components of network management

**Managing server:** application, typically with network managers (humans) in the loop

**Network management protocol:** used by managing server to query, configure, manage device; used by devices to inform managing server of data, events.

**Managed device:** equipment with manageable, configurable hardware, software components

**Data:** device “state” configuration data, operational data, device statistics

Network Layer: 5-92

92

### Network operator approaches to management

**CLI (Command Line Interface)**

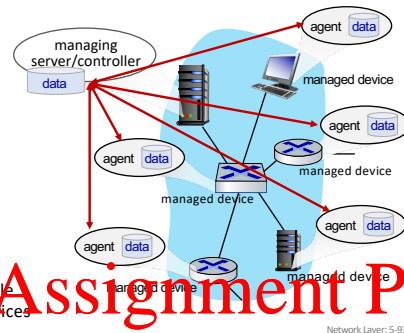
- operator issues (types, scripts) direct to individual devices (e.g., via ssh)

**SNMP/MIB**

- operator queries/sets devices data (MIB) using Simple Network Management Protocol (SNMP)

**NETCONF/YANG**

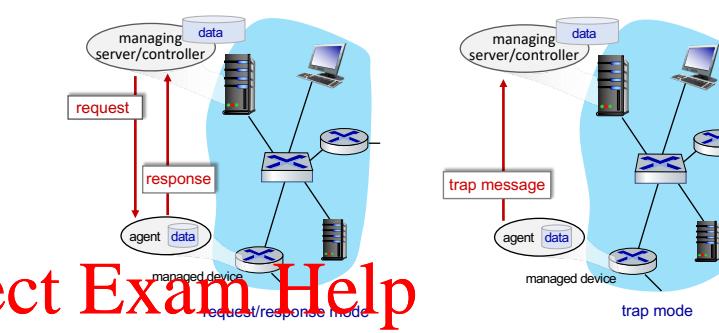
- more abstract, network-wide, holistic
- emphasis on multi-device configuration management.
- YANG: data modeling language
- NETCONF: communicate YANG-compatible actions/data to/from/among remote devices



Network Layer: 5-93

### SNMP protocol

Two ways to convey MIB info, commands:



request/response mode

trap mode

Network Layer: 5-94

93

94

# Assignment Project Exam Help

<https://powcoder.com>

### SNMP protocol: message types

Message type	Function
GetRequest GetNextRequest GetBulkRequest	manager-to-agent: "get me data" (data instance, next data in list, block of data).
SetRequest	manager-to-agent: set MIB value
Response	Agent-to-manager: value, response to Request
Trap	Agent-to-manager: inform manager of exceptional event

Network Layer: 5-95

### SNMP protocol: message formats

message types 0-3



PDU type (0-3) Request ID Error Status (0-5) Error Index Name Value Name Value ....

message type 4



PDU type 4 Enterprise Agent Addr Trap Type (0-7) Specific code Time stamp Name Value ....

SNMP PDU

Network Layer: 5-96

95

96

## SNMP: Management Information Base (MIB)

- managed device's operational (and some configuration) data
- gathered into device **MIB module**
  - 400 MIB modules defined in RFC's; many more vendor-specific MIBs
- Structure of Management Information (SMI):** data definition language
- example MIB variables for UDP protocol:

Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPIPDatagrams	32-bit counter	total # datagrams delivered
1.3.6.1.2.1.7.2	UDPNoPorts	32-bit counter	# undeliverable datagrams (no application at port)
1.3.6.1.2.1.7.3	UDInErrors	32-bit counter	# undeliverable datagrams (all other reasons)
1.3.6.1.2.1.7.4	UDPOutDatagrams	32-bit counter	total # datagrams sent
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	One entry for each port currently in use

Network Layer: 5-97

97

## NETCONF overview

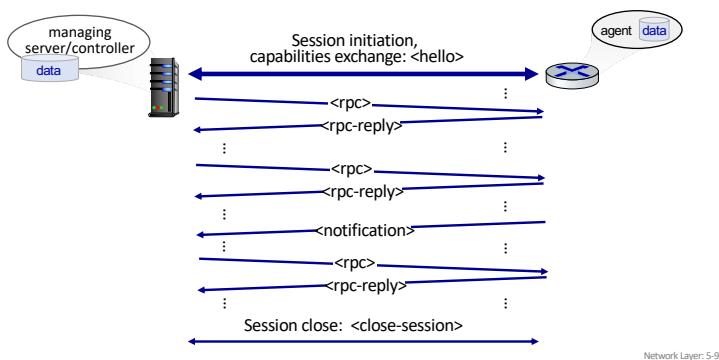
- goal:** actively manage/configure devices network-wide
- operates between managing server and managed network devices
  - actions: retrieve, set, modify, activate configurations
  - atomic-commit** actions over multiple devices
  - query operational data and statistics
  - subscribe to notifications from devices
- remote procedure call (RPC) paradigm
  - NETCONF protocol messages encoded in XML
  - exchanged over secure, reliable transport (e.g., TLS) protocol

Network Layer: 5-98

98

Assignment Project Exam Help  
<https://powcoder.com>

## NETCONF initialization, exchange, close



99

## Selected NETCONF Operations

NETCONF	Operation Description
<get-config>	Retrieve all or part of a given configuration. A device may have multiple configurations.
<get>	Retrieve all or part of both configuration state and operational state data.
<edit-config>	Change specified (possibly running) configuration at managed device. Managed device <rpc-reply> contains <ok> or <rpcerror> with rollback.
<lock>, <unlock>	Lock (unlock) configuration datastore at managed device (to lock out NETCONF, SNMP, or CLIs commands from other sources).
<create-subscription>, <notification>	Enable event notification subscription from managed device

Network Layer: 5-100

100

## Sample NETCONF RPC message

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <rpc message-id="101" note message id
03   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
04     <edit-config> change a configuration
05       <target>
06         <running/> change the running configuration
07       </target>
08       <config>
09         <top xmlns="http://example.com/schema/
10           1.2/config">
11             <interface>
12               <name>Ethernet0/0</name> change MTU of Ethernet 0/0 interface to 1500
13               <mtu>1500</mtu>
14             </interface>
15           </top>
16         </config>
17     </edit-config>
18 </rpc>

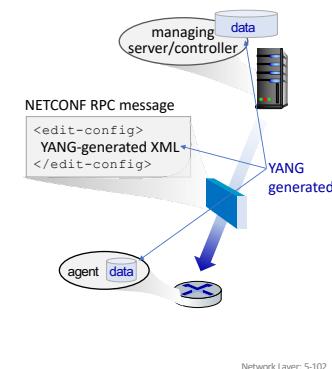
```

Network Layer: 5-101

101

## YANG

- data modeling language used to specify structure, syntax, semantics of NETCONF network management data
  - built-in data types, like SMI
- XML document describing device, capabilities can be generated from YANG description
- can express constraints among data that must be satisfied by a valid NETCONF configuration
  - ensure NETCONF configurations satisfy correctness, consistency constraints



102

# Assignment Project Exam Help

## Network layer: Summary

**we've learned a lot!**

- approaches to network control plane
  - per-router control (traditional)
  - logically centralized control (software defined networking)
- traditional routing algorithms
  - implementation in Internet: OSPF , BGP
- SDN controllers
  - implementation in practice: ODL, ONOS
- Internet Control Message Protocol
- network management

*next stop: link layer!*

Network Layer: 5-103

103

# Add WeChat powcoder

## Network layer, control plane: Done!

- introduction
- routing protocols
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol
- network management, configuration
  - SNMP
  - NETCONF/YANG



Network Layer: 5-104

104

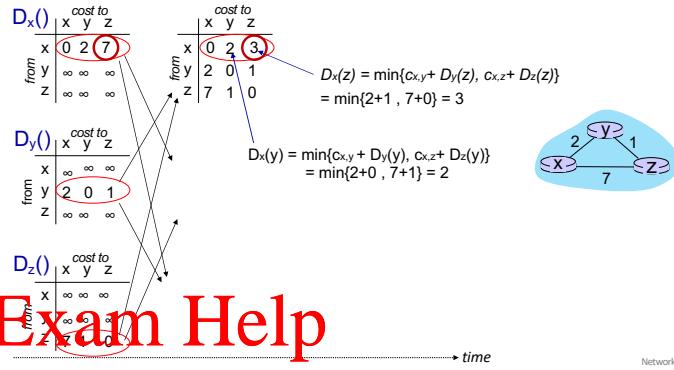
Additional Chapter 5 slides

## Assignment Project Exam Help

Network Layer: 5-105

105

### Distance vector: another example

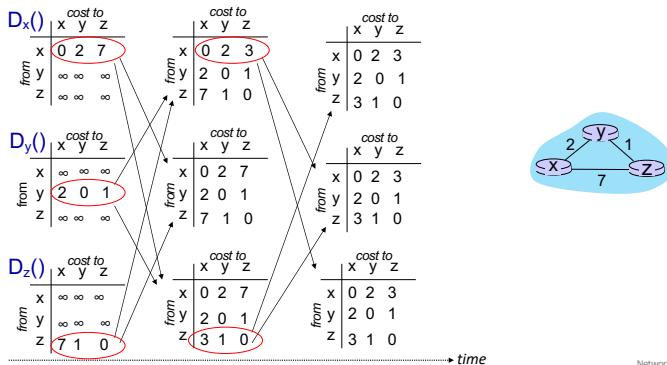


106

<https://powcoder.com>

### Distance vector: another example

Add WeChat powcoder



107