شبكات الحواسيب
Computer Networks

Dr.-Eng. Samer Sulaiman

2021-2022

مفردات المنهاج

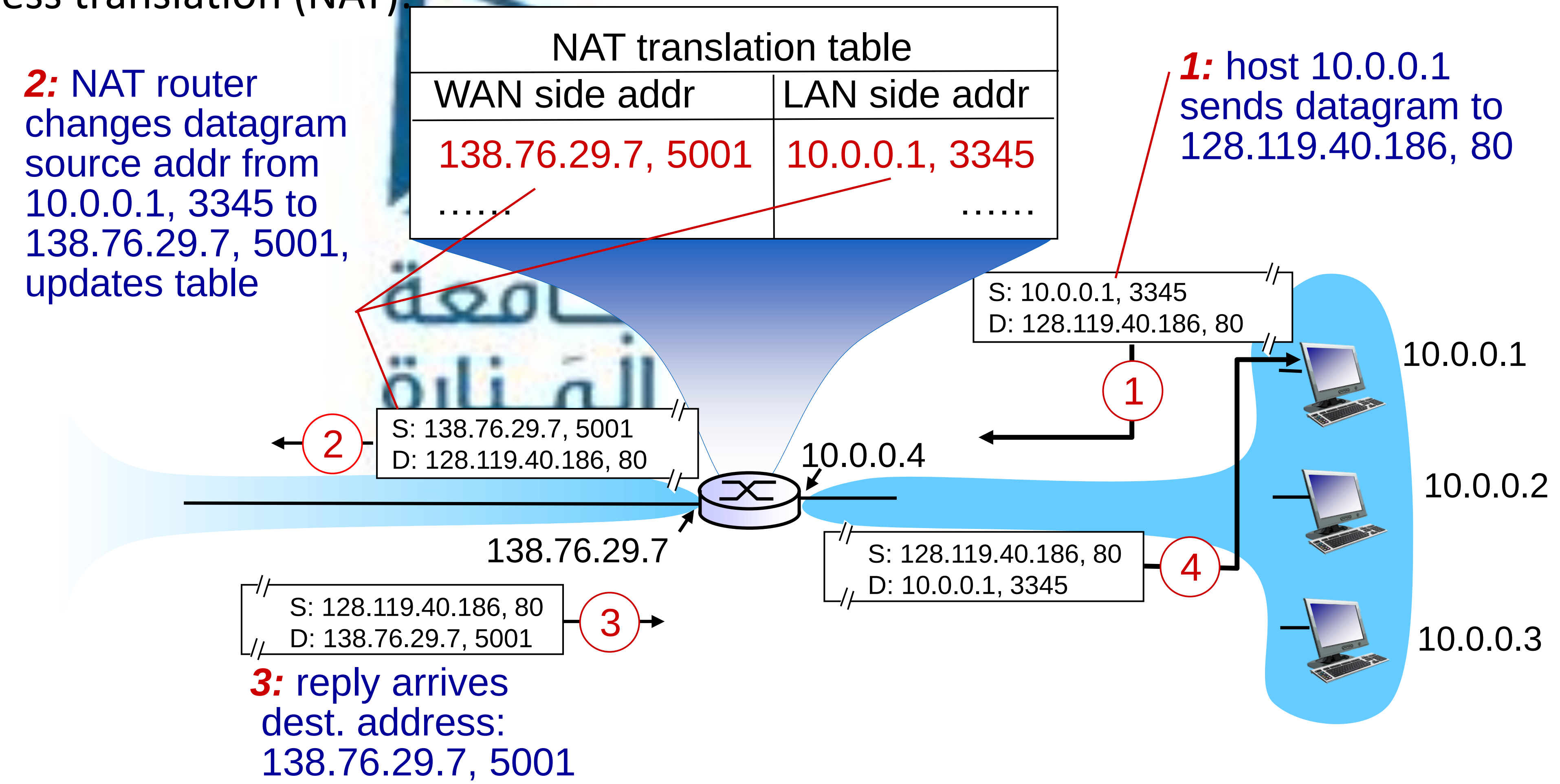
- أساسيات شبكات الحواسيب
  - بنية وهيكلية شبكات الحواسيب
- **طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)**
  - **البروتوكولات والطرق والخوارزميات المستخدمة في كل طبقة**
- تطبيقات شبكات الحواسيب في مجال انترنت الأشياء IOT

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - network address translation (NAT):
    - motivation: local network uses just one IP address as far as outside world is concerned:
      - 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
      - devices inside local net not explicitly addressable, visible by outside world (a security plus)
    - implementation: NAT router must:
      outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
      - remote clients/servers will respond using (NAT IP address, new port #) as destination address
    - **remember** (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
    - incoming datagrams: replace (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - network address translation (NAT):

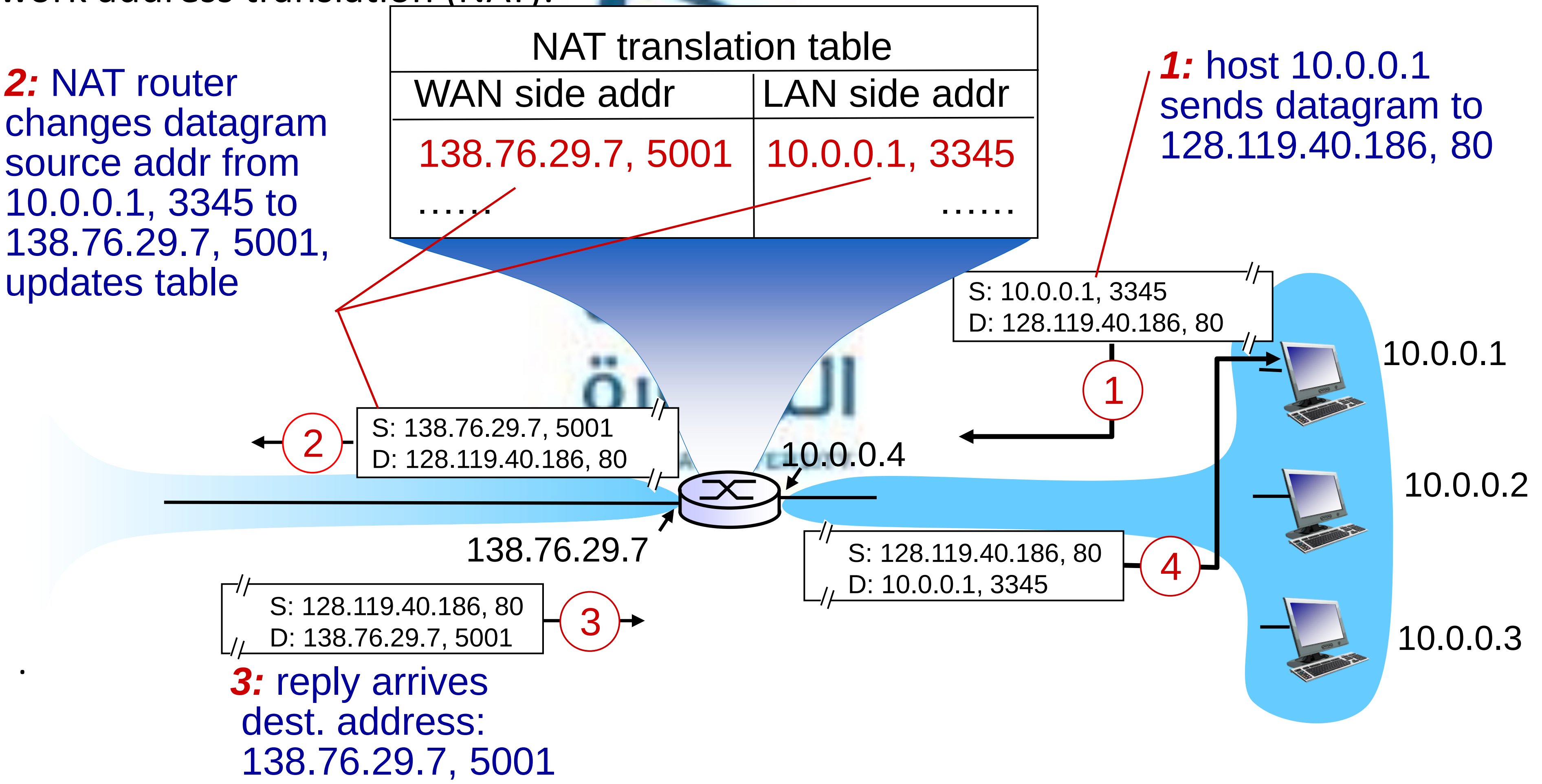


طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - network address translation (NAT):
    - motivation: local network uses just one IP address as far as outside world is concerned:
      - 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
      - devices inside local net not explicitly addressable, visible by outside world (a security plus)
    - implementation: NAT router must:
      - outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #). . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr
      - remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
      - incoming datagrams: replace (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - network address translation (NAT):



- .

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Routing protocols:
    - Routing protocol goal: determine "good" paths (equivalently, routes), from sending hosts to receiving host, through network of routers
      - path: sequence of routers packets will traverse in going from given initial source host to given final destination host
      - "good": least "cost", "fastest", "least congested", the least monetary cost,
      - routing: an important topic in networking
  - Routing algorithm
    - classification:
      - Q: global or decentralized information?
        - global:
          - all routers have complete topology, link cost info
          - "link state" algorithms
        - decentralized:
          - router knows physically-connected neighbors, link costs to neighbors
          - iterative process of computation, exchange of info with neighbors
          - "distance vector" algorithms

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Routing algorithm
    - classification:
      - Q: static or dynamic?
        - static:
          - routes change slowly over time
        - dynamic:
          - routes change more quickly
          - periodic update
          - in response to link cost changes
    - A link-state routing algorithm
      - Dijkstra’s algorithm
        - net 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 dest.’s

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

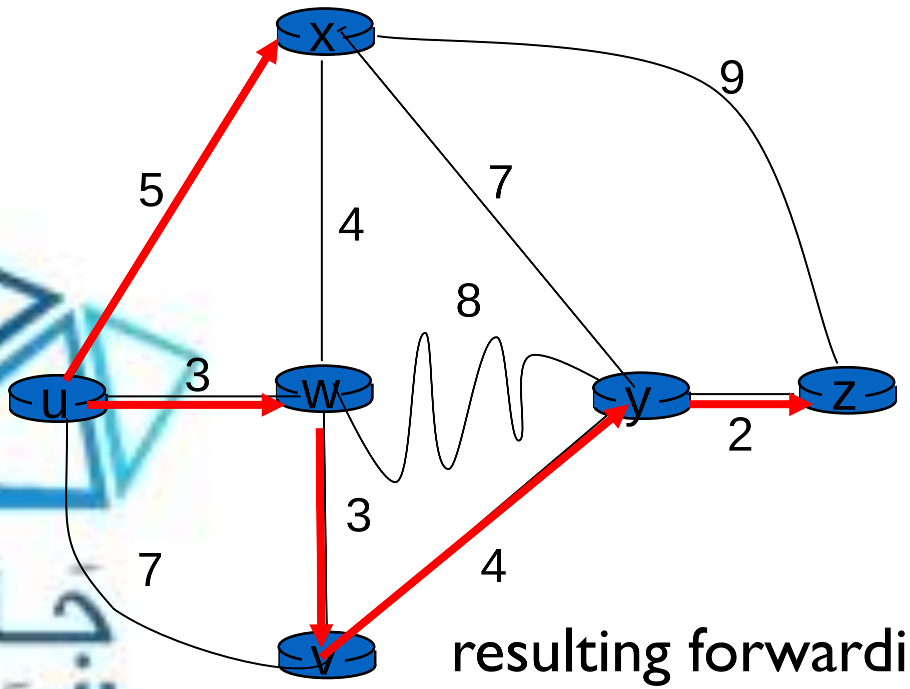
- Network layer:
  - Routing algorithm
    - A link-state routing algorithm
      - Dijkstra’s algorithm
        - notation:
          - c(x,y): link cost from node x to y; = ∞ if not direct neighbors
          - D(v): current value of cost of path from source to dest. v
          - p(v): predecessor node along path from source to v
          - N': set of nodes whose least cost path definitively known

```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞

 Loop
   find w not in N' such that D(w) is a minimum
    add w to N'
    update D(v) for all v adjacent to w and not in N' :
       D(v) = min( D(v), D(w) + c(w,v) )
    /* new cost to v is either old cost to v or known
     shortest path cost to w plus cost from w to v */
 until all nodes in N'
```

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Routing algorithm
    - A link-state routing algorithm
      - Dijkstra’s algorithm
        - Example:



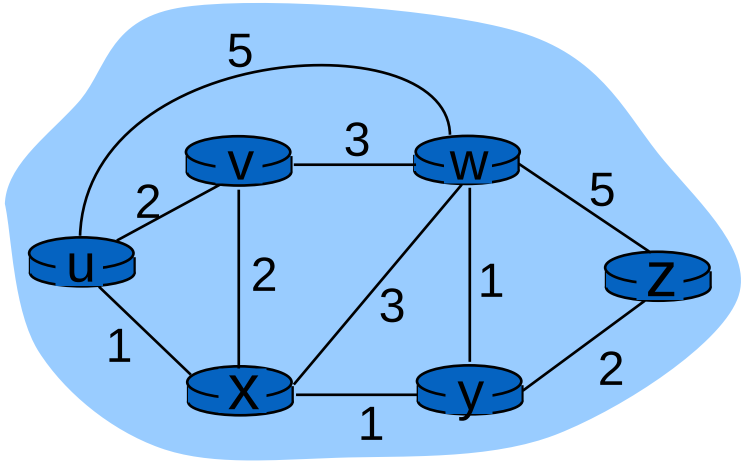
Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

resulting forwarding table in u:

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

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

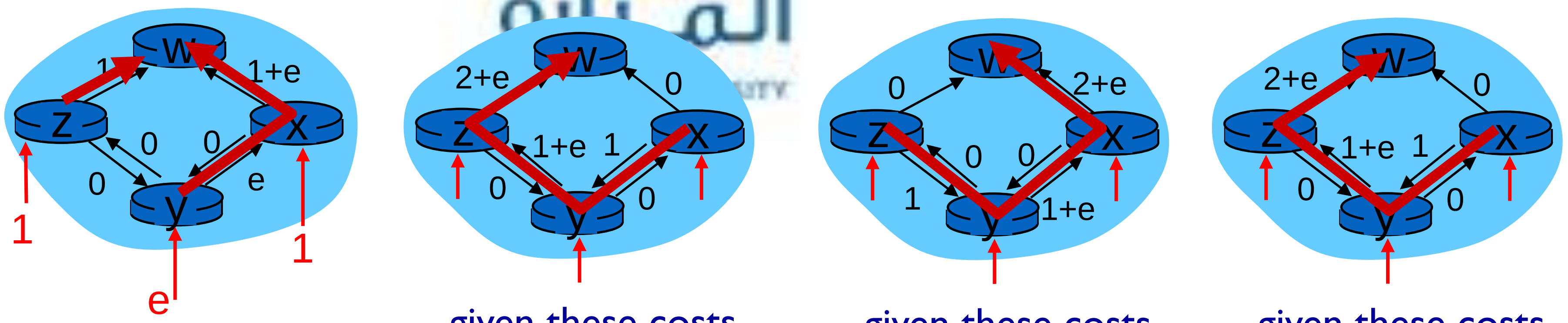
- Network layer:
  - Routing algorithm
    - A link-state routing algorithm
      - Dijkstra's algorithm
        - Example2:



Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Routing algorithm
    - A link-state routing algorithm
      - Dijkstra’s algorithm
        - oscillations possible:
          - e.g., support link cost equals amount of carried traffic:
        - solutions
          - Prevent using load as a metric of cost!
          - Prevent routers from running LS algorithm at the same time
          - randomize the time it sends out a link advertisement by routers



initially

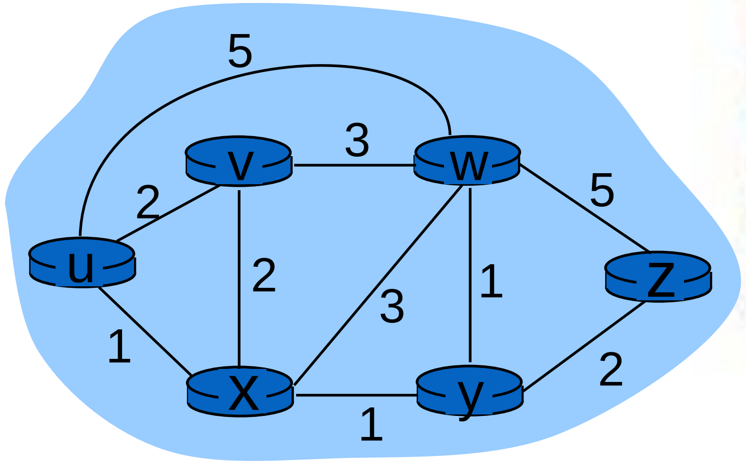
given these costs, find new routing…. resulting in new costs

given these costs, find new routing…. resulting in new costs

given these costs, find new routing…. resulting in new costs

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Routing algorithm
    - Distance vector algorithm
      - Bellman-Ford example



clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

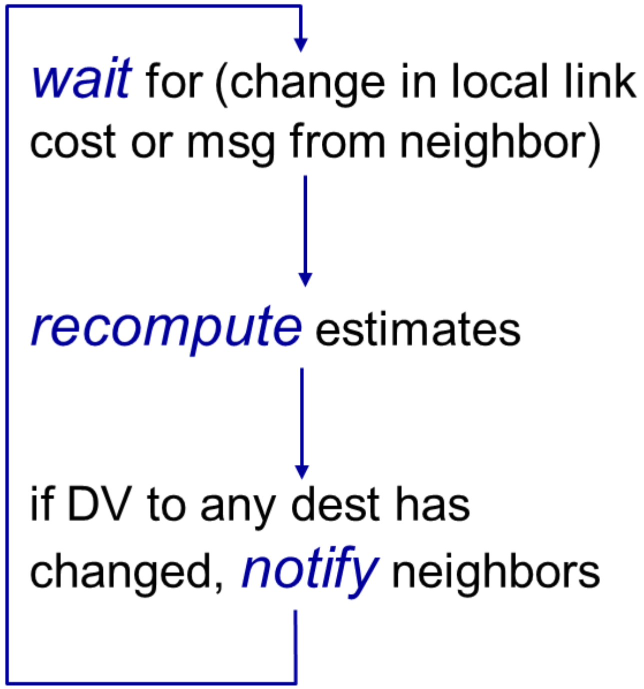
$$
\begin{aligned}
d_u(z) &= \min \{ c(u,v) + d_v(z), \\
&\qquad c(u,x) + d_x(z), \\
&\qquad c(u,w) + d_w(z) \} \\
&= \min \{2 + 5, \\
&\qquad 1 + 3, \\
&\qquad 5 + 3\} = 4
\end{aligned}
$$

node achieving minimum is next
hop in shortest path, used in forwarding table

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

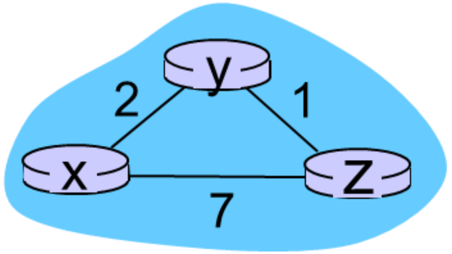
- Network layer:
  - Routing algorithm
    - Distance vector algorithm
      - key idea:
        - from time-to-time, each node sends its own distance vector estimate to neighbors
        - when x receives new DV estimate from neighbor, it updates its own DV using B-F equation:
          - $D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\}$ *for each node* $y \in N$
      - iterative, asynchronous: each local iteration caused by:
        - local link cost change
        - DV update message from neighbor
      - distributed:
        - each node notifies neighbors only when its DV changes
        - neighbors then notify their neighbors if necessary

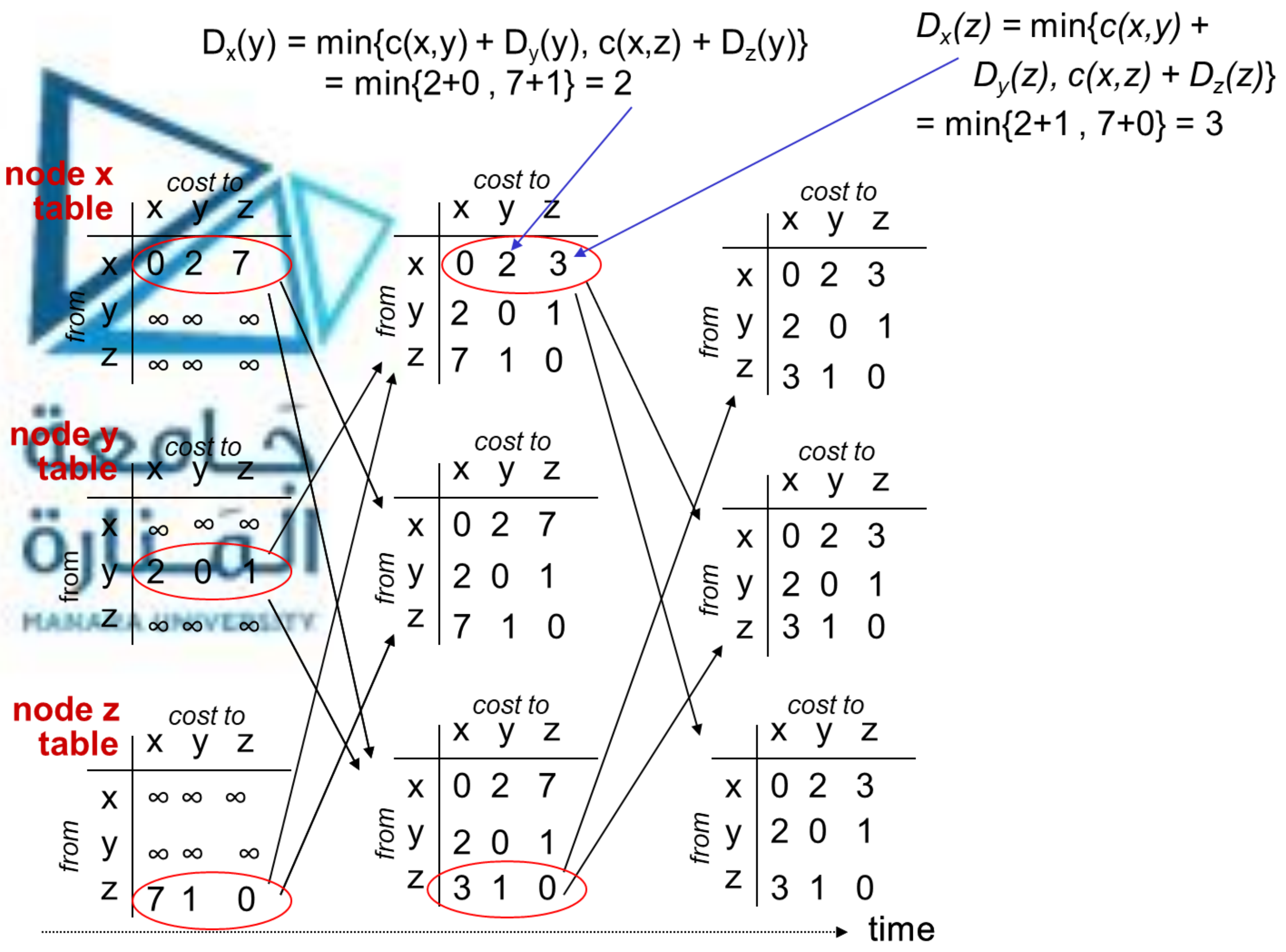
each node:



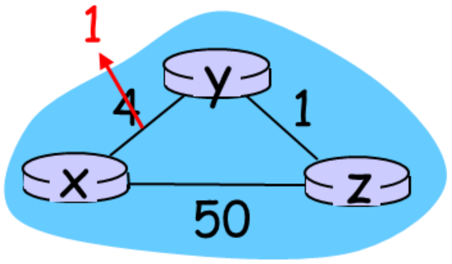
طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Routing algorithm
    - Distance vector algorithm
      - Example:



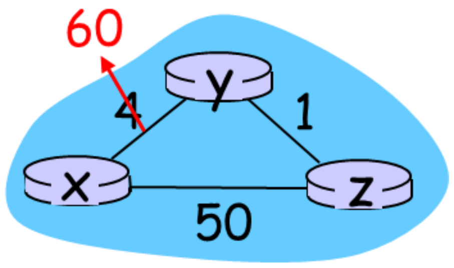


- Network layer:
  - Routing algorithm
    - Distance vector algorithm
      - link cost changes:
        - node detects local link cost change
        - updates routing info, recalculates distance vector
        - if DV changes, notify neighbors
        - Time-steps:
          - t0 : y detects link-cost change, updates its DV, informs its neighbors.
          - t1 : z receives update from y, updates its table, computes new least cost to x , sends its neighbors its DV.
          - t2 : y receives z’s update, updates its distance table. y’s least costs do not change, so y does not send a message to z.



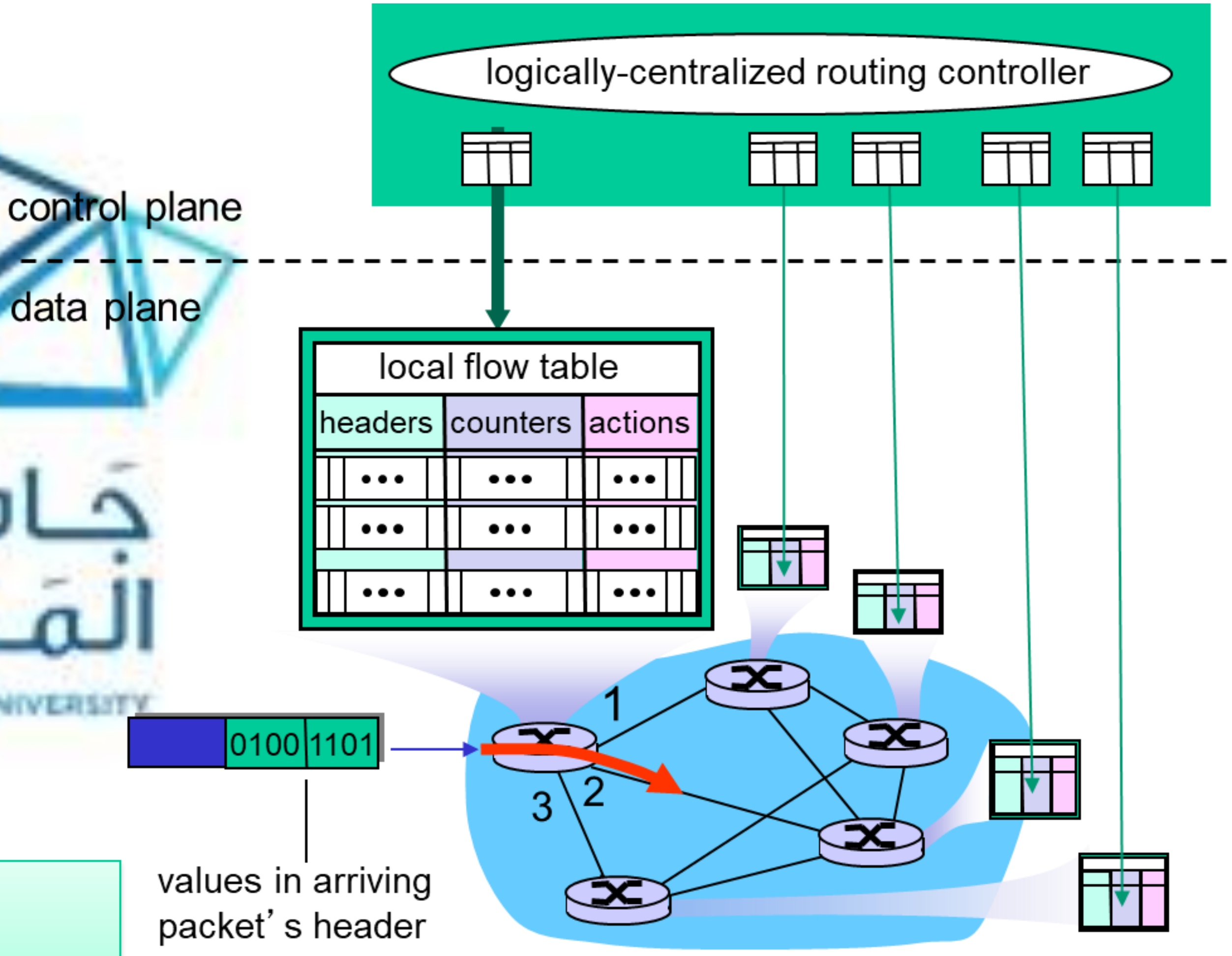
طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Routing algorithm
    - Distance vector algorithm
      - link cost changes:
        - node detects local link cost change
        - bad news travels slow - “count to infinity” problem!
        - 44 iterations before algorithm stabilizes: see text
        - Solution: add poisoned reverse
          - If Z routes through Y to get to X :
            - Z tells Y its (Z’s) distance to X is infinite (so Y won’t route to X via Z)
          - will this completely solve count to infinity problem?



طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

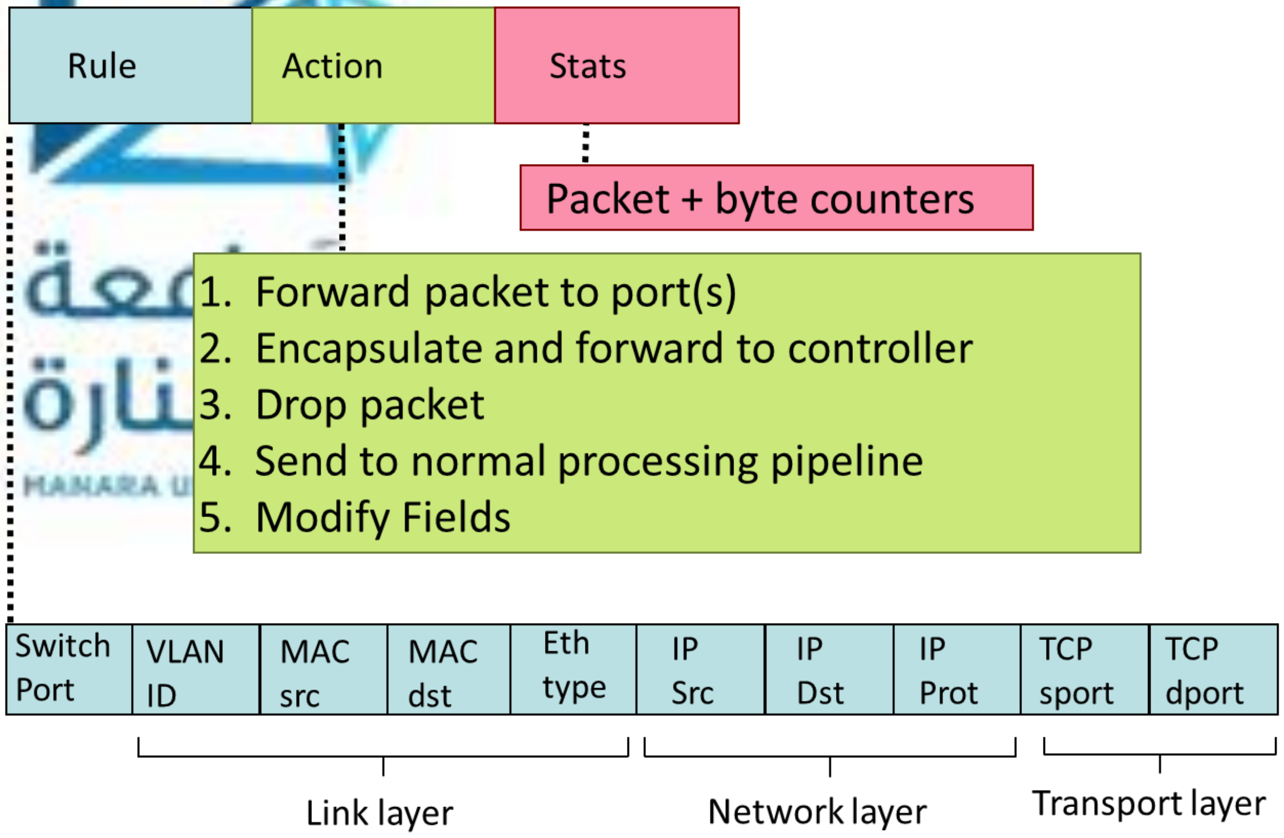
- Network layer:
  - Generalized Forwarding and SDN:
    - Each router contains a flow table that is computed and distributed by a logically centralized routing controller
    - generalized forwarding:
      - simple packet-handling rules
      - Match (rule): packet header fields to be matched
      - Actions: for matched packet: drop, forward, modify, matched packet or send matched packet to controller
      - Counters: #bytes and #packets



1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Generalized Forwarding and SDN:
    - OpenFlow: Flow Table Entries



طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Generalized Forwarding and SDN:
    - OpenFlow: Flow Table Entries
      - Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	port3

layer 2 frames from MAC address 22:A7:23:11:E1:02 should be forwarded to output port 3

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Generalized Forwarding and SDN:
    - OpenFlow: Flow Table Entries
      - Examples

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

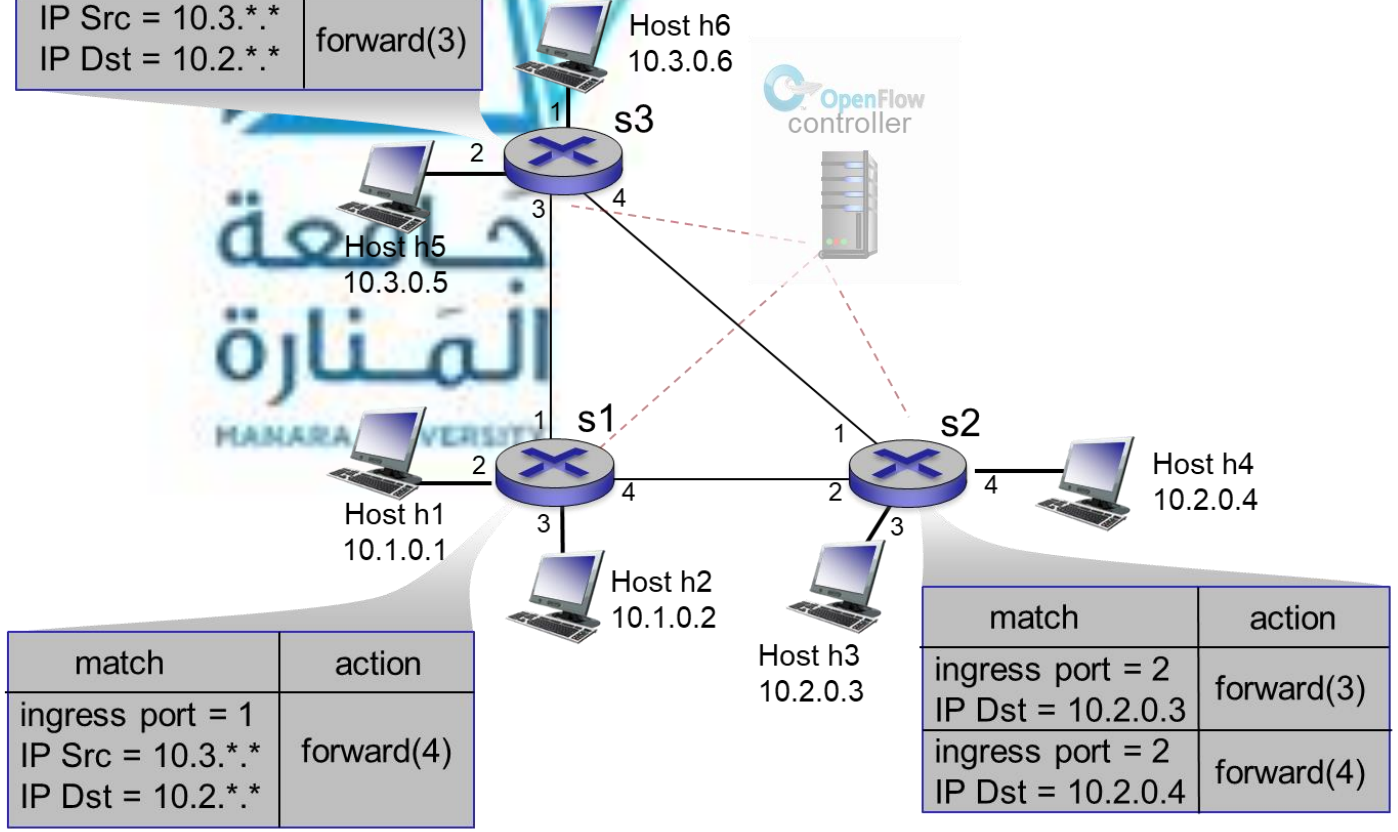
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- Network layer:
  - Generalized Forwarding and SDN:
    - OpenFlow Example:

Example: datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2



match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)

match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)