# شبكات الحواسيب

# Computer Networks

Dr.-Eng. Samer Sulaiman

2023-2024

# 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

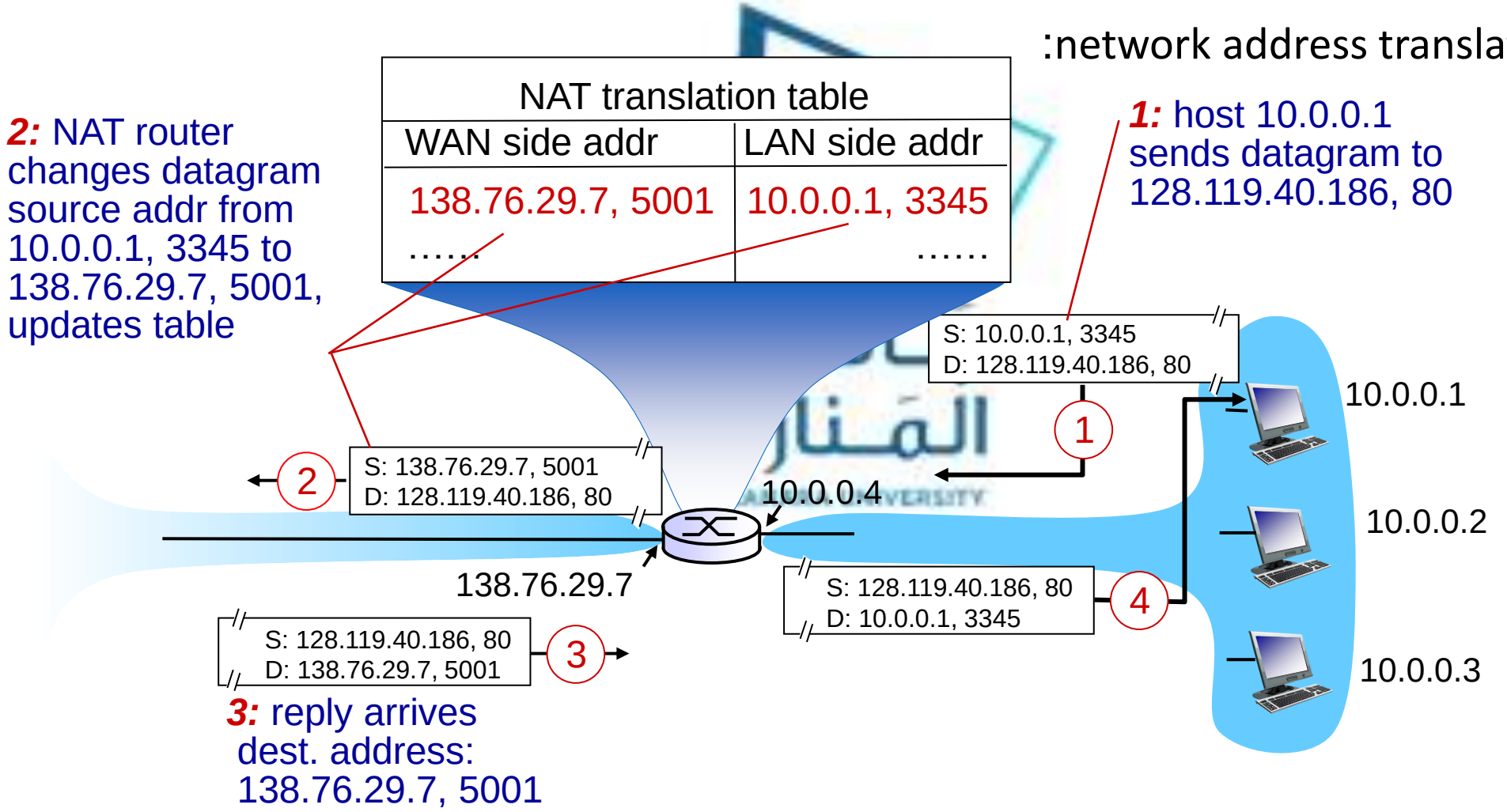
جامعة المنارة
MANARA UNIVERSITY

# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- طبقة الشبكة:
  - network address translation (NAT):
    - الهدف: إمكانية استخدام الشبكة المحلية لعنوان IP واحدًا فقط متصل بالعالم الخارجي:
      - نطاق من العناوين غير مطلوب من ISP: عنوان IP عام واحد فقط لجميع الأجهزة
      - يمكن تغيير عناوين الأجهزة في الشبكة المحلية دون إخطار العالم الخارجي
      - يمكن تغيير مزود خدمة الإنترنت دون تغيير عناوين الأجهزة في الشبكة المحلية
      - الأجهزة الموجودة داخل الشبكة المحلية غير قابلة للعنونة بشكل صريح ، ومرئية من قبل العالم الخارجي (أمان إضافي)
    - التنفيذ: يجب على جهاز توجيه الـ NAT:
      - حزم البيانات المرسلة: استبدال (عنوان IP المصدر ، المنفذ #) من كل حزم بيانات المرسل إلى (عنوان IP NAT، المنفذ الجديد #)
      - سيستجيب العملاء / الخوادم البعيدة باستخدام (عنوان IP NAT، المنفذ الجديد #) كعنوان هدف
    - ملاحظة: (في جدول الـ NAT)يتألف كل سجل من زوج المعلومات (عنوان IP المصدر ، المنفذ #) إلى (عنوان IP NAT، المنفذ الجديد #)
    - حزم البيانات المستقبلة: استبدل (عنوان IP NAT، المنفذ الجديد #) في حقول الاستقبال لكل حزم بيانات المستقبلة مع (عنوان IP المصدر، المنفذ #) المخزن في جدول NAT

# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- طبقة الشبكة:
  - network address translation (NAT):



# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- طبقة الشبكة:
  - بروتوكولات التوجيه:
    - هدف بروتوكول التوجيه: تحديد المسارات "الجيدة" (بالتساوي ، المسارات)، من المرسل إلى المستقبل، عبر شبكة أجهزة التوجيه
    - المسار(path): انتقال تسلسل حزم بيانات الموجهات من مرسل معين إلى مستقبل معين
    - "جيد" (good): أقل "تكلفة" ، "أسرع" ، "أقل ازدحامًا" ، أقل تكلفة عملية
    - التوجيه (routing): يعتبر من المواضيع المهم في الشبكات
  - خوارزمية التوجيه
  - التصنيف:
    - يمكن تصنيفها اعتماداً على السؤال التالي: هل المعلومات عامة أم لا مركزية؟
      - عامة (global):
        - تحتوي جميع الموجهات على مخطط كامل ومعلومات عن تكلفة الاتصال
        - خوارزميات "link state"
      - لامركزية (decentralized):
        - يحتوي الموجه على معلومات عن الجيران المتصلين معه فيزيائياً، وتكاليف الاتصال بالجيران
        - عملية تكرارية للحساب وتبادل المعلومات مع الجيران
        - خوارزميات "distance vector"

جامعة المنارة
MANARA UNIVERSITY

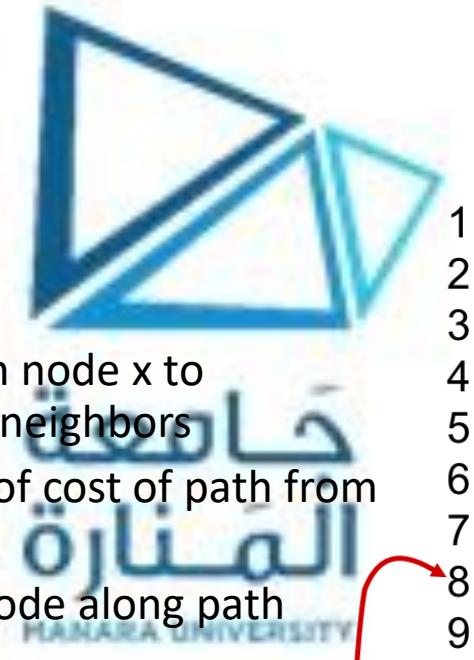
# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

- طبقة الشبكة:
  - خوارزمية التوجيه
    - التصنيف:
      - يمكن تصنيفها اعتماداً على السؤال التالي: هل هي ثابتة أم ديناميكية؟
      - ثابتة (static):
        - تتغير المسارات ببطء مع مرور الزمن
      - ديناميكية (dynamic):
        - تتغير المسارات بسرعة أكبر
        - التحديث الدوري
        - استجابةً لتغيرات كلفة الاتصال (الوصلة)
    - A link-state routing algorithm
      - Dijkstra’s algorithm
        - طوبولوجيا الشبكة ، تكاليف الاتصال يجب أن تكون معروفة لجميع العقد
        - يتم إنجازها بواسطة ”link state broadcast“
        - جميع العقد لديها نفس المعلومات
        - تحسب المسارات الأقل تكلفة من عقدة معينة ("المصدر") إلى جميع العقد الأخرى
        - يعطي جدول التمرير (forwarding table) لتلك العقدة
        - تكرارية: بعد تكرارات k، يتم التعرف على المسارات الأقل تكلفة لـ k مستقبل

جامعة المنارة

# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

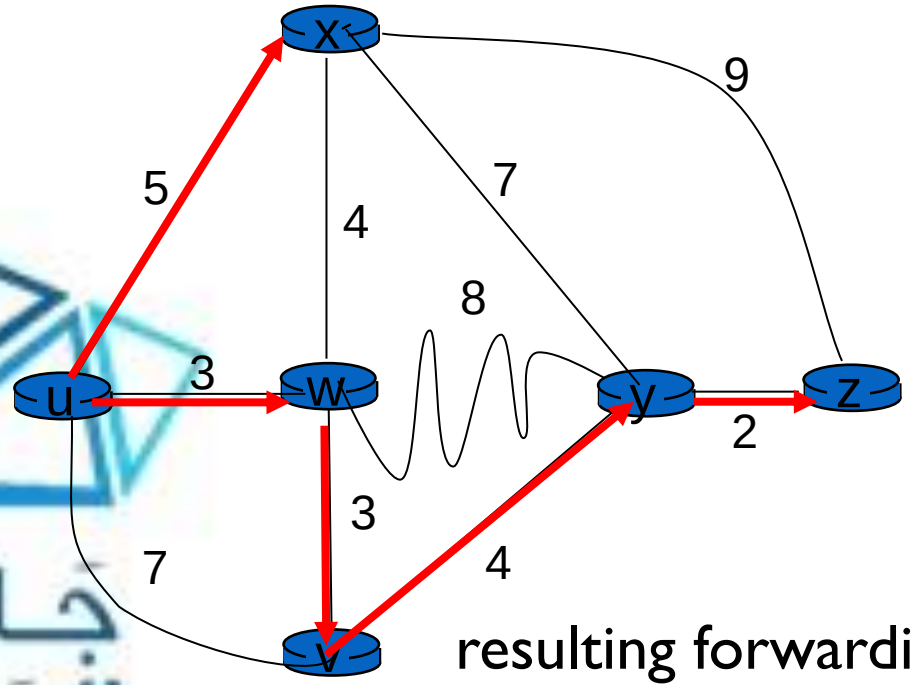
- Network layer:
  - Routing algorithm
    - A link-state routing algorithm
      - Dijkstra's algorithm
        - notation:
          - c(x,y): link cost from node x to y; = ∞ if not direct neighbors
          - D(v): current value of cost of path from source to dest. v
          - p(v): predecessor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




```
1  Initialization:
2   N' = {u}
3   for all nodes v
4     if v adjacent to u
5         then D(v) = c(u,v)
6     else D(v) = ∞
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 cost to v is either old cost to v or known
14    shortest path cost to w plus cost from w to v */
15 until all nodes in N'
```

# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

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



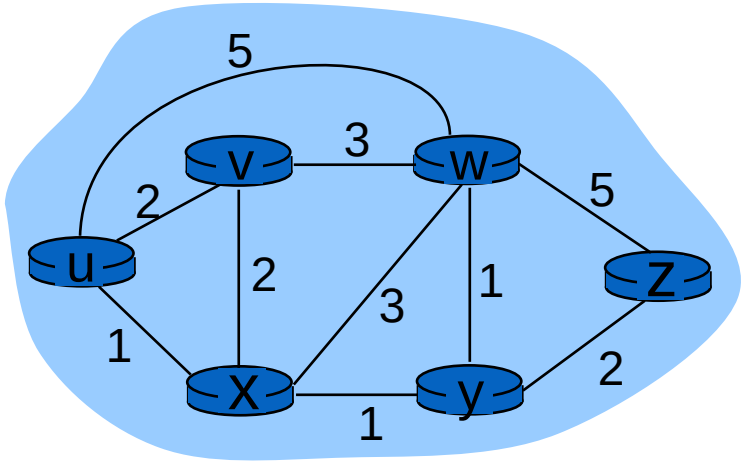

| Step | N' | D(v) p(v) | D(w) p(w) | D(x) p(x) | D(y) p(y) | D(z) p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 7,u | 3,u | 5,u | ∞ | ∞ |
| 1 | uw | 6,w | | 5,u | 11,w | ∞ |
| 2 | uwx | 6,w | | | 11,w | 14,x |
| 3 | uwxv | | | | 10,v | 14,x |
| 4 | uwxvy | | | | | 12,y |
| 5 | uwxvyz | | | | | |

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)

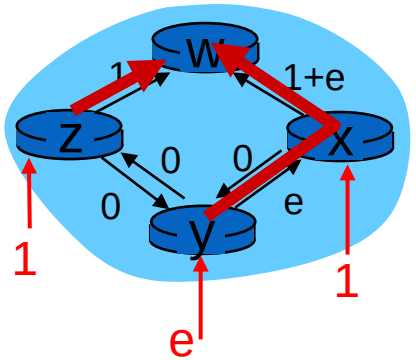
- Network layer:
  - Routing algorithm
    - A link-state routing algorithm
      - Dijkstra's algorithm
        - Example2:



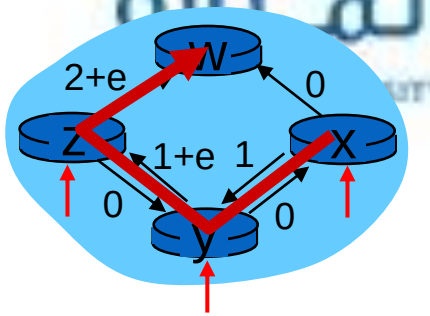

| Step | N' | D(v),p(v) | D(w),p(w) | D(x),p(x) | D(y),p(y) | D(z),p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 2,u | 5,u | 1,u | ∞ | ∞ |
| 1 | ux | 2,u | 4,x | | 2,x | ∞ |
| 2 | uxy | 2,u | 3,y | | | 4,y |
| 3 | uxyv | | 3,y | | | 4,y |
| 4 | uxyvw | | | | | 4,y |
| 5 | uxyvwz | | | | | |

# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

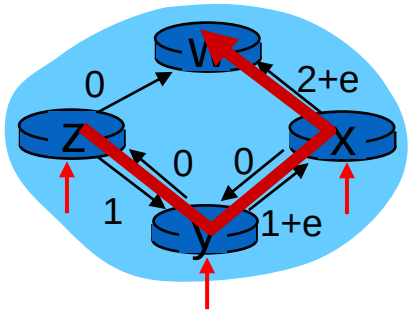
- Network layer:
  - Routing algorithm
    - A link-state routing algorithm
      - Dijkstra's algorithm
        - oscillations possible:
          - e.g., support link cost equals amount of carried traffic:
        - solutions
          - Prevent using load as a metric of cost!
          - Prevent routers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



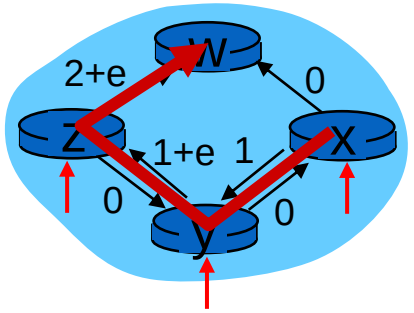

initially




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



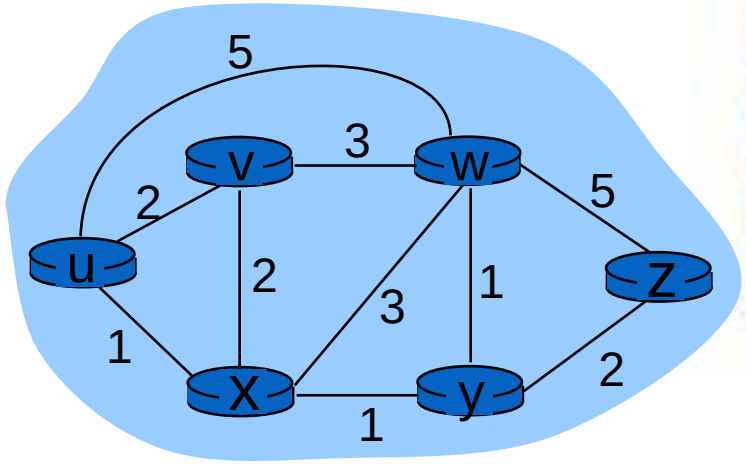

given these costs, find new routing…. resulting in new costs




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:

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

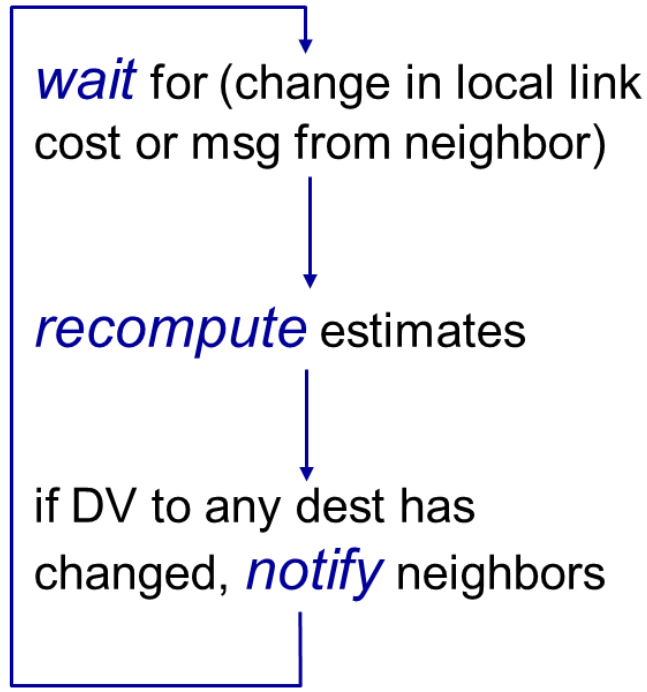
node achieving minimum is next
hop in shortest path, used in forwarding table

# 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

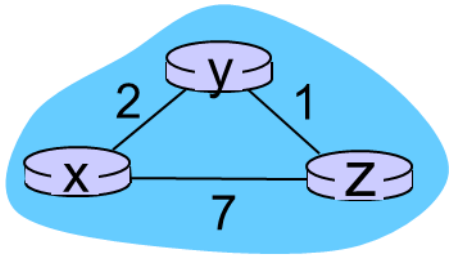
جامعة المنارة
MANARA UNIVERSITY

*each node:*



# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

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




$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$

**node x table**

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | 0 | 2 | 7 |
| y | ∞ | ∞ | ∞ |
| z | ∞ | ∞ | ∞ |

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | 0 | 2 | 3 |
| y | 2 | 0 | 1 |
| z | 7 | 1 | 0 |

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | 0 | 2 | 3 |
| y | 2 | 0 | 1 |
| z | 3 | 1 | 0 |

**node y table**

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | ∞ | ∞ | ∞ |
| y | 2 | 0 | 1 |
| z | ∞ | ∞ | ∞ |

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | 0 | 2 | 7 |
| y | 2 | 0 | 1 |
| z | 7 | 1 | 0 |

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | 0 | 2 | 3 |
| y | 2 | 0 | 1 |
| z | 3 | 1 | 0 |

**node z table**

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | ∞ | ∞ | ∞ |
| y | ∞ | ∞ | ∞ |
| z | 7 | 1 | 0 |

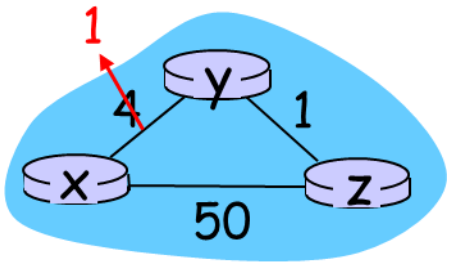
| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | 0 | 2 | 7 |
| y | 2 | 0 | 1 |
| z | 3 | 1 | 0 |

| from | cost to x | cost to y | cost to z |
|---|---|---|---|
| x | 0 | 2 | 3 |
| y | 2 | 0 | 1 |
| z | 3 | 1 | 0 |

time

# طبقات شبكات الحواسيب (الشبكة، النقل، التطبيقات)

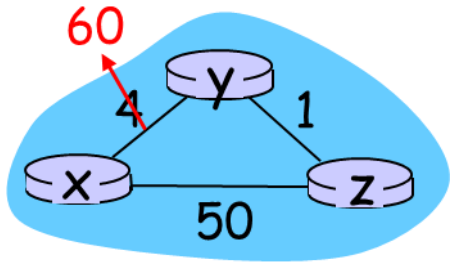
- Network layer:
  - Routing algorithm
    - Distance vector algorithm
      - link cost changes:
        - node detects local link cost change
        - updates routing info, recalculates distance vector
        - if DV changes, notify neighbors
        - Time-steps:
          - t0 : y detects link-cost 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.



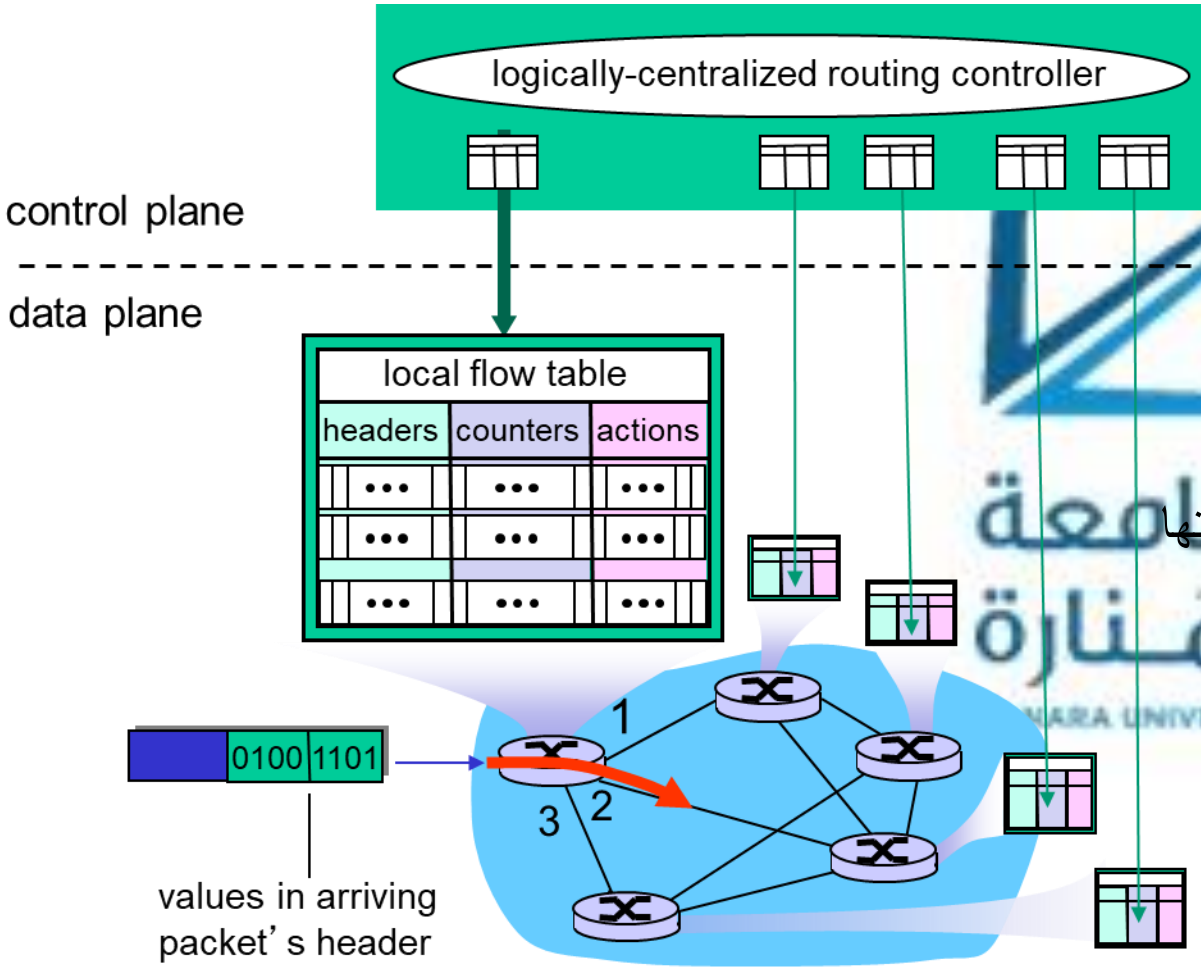

جامعة المنارة

# 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)

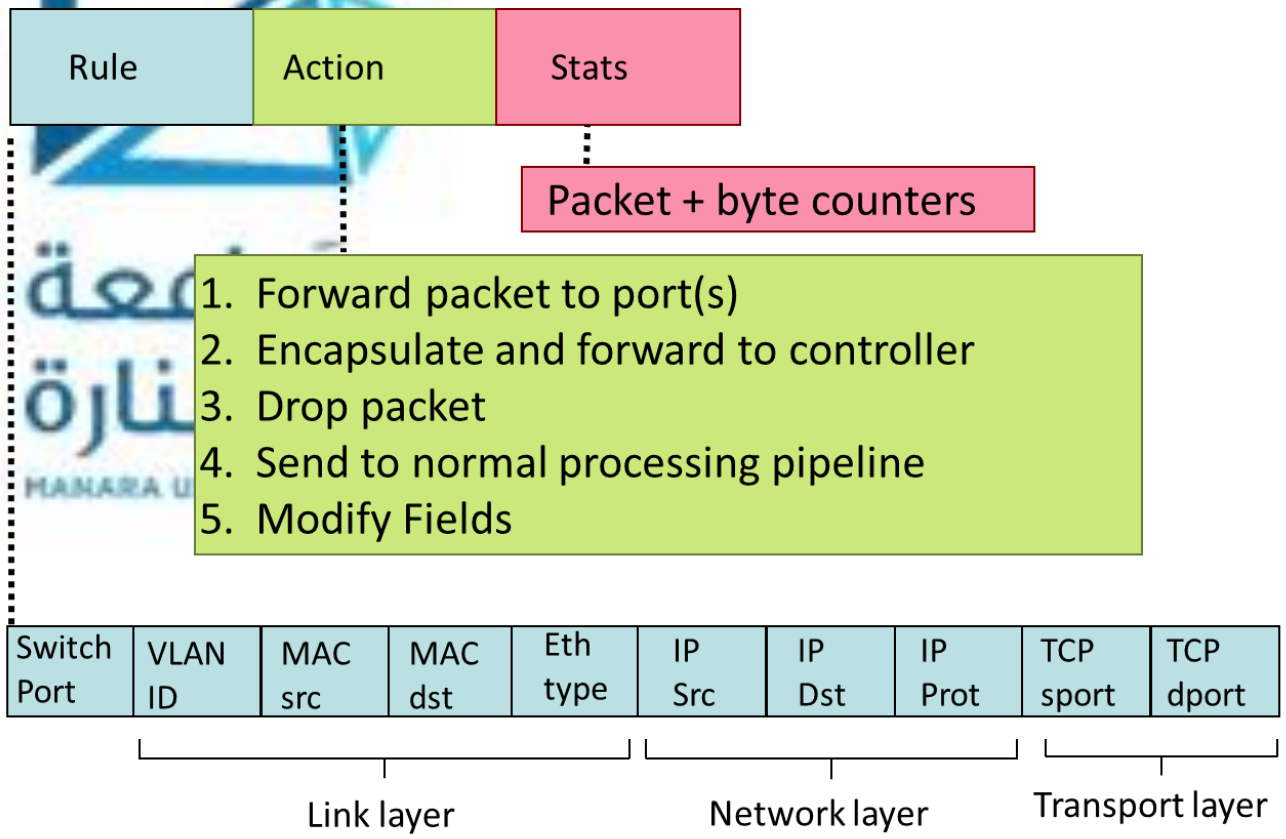



- طبقة الشبكة:
  - Generalized Forwarding and SDN:
    - يحتوي كل موجه على جدول تدفق يتم حسابه وتوزيعه بواسطة وحدة تحكم توجيه مركزية
    - generalized forwarding:
    - قواعد بسيطة للتعامل مع الحزم
    - Match (rule): حقول رأس الحزمة المطلوب مطابقتها:
    - الإجراءات (Actions) للحزمة المتطابقة:
      - drop, forward, modify, matched packet or send matched packet to controller

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 |

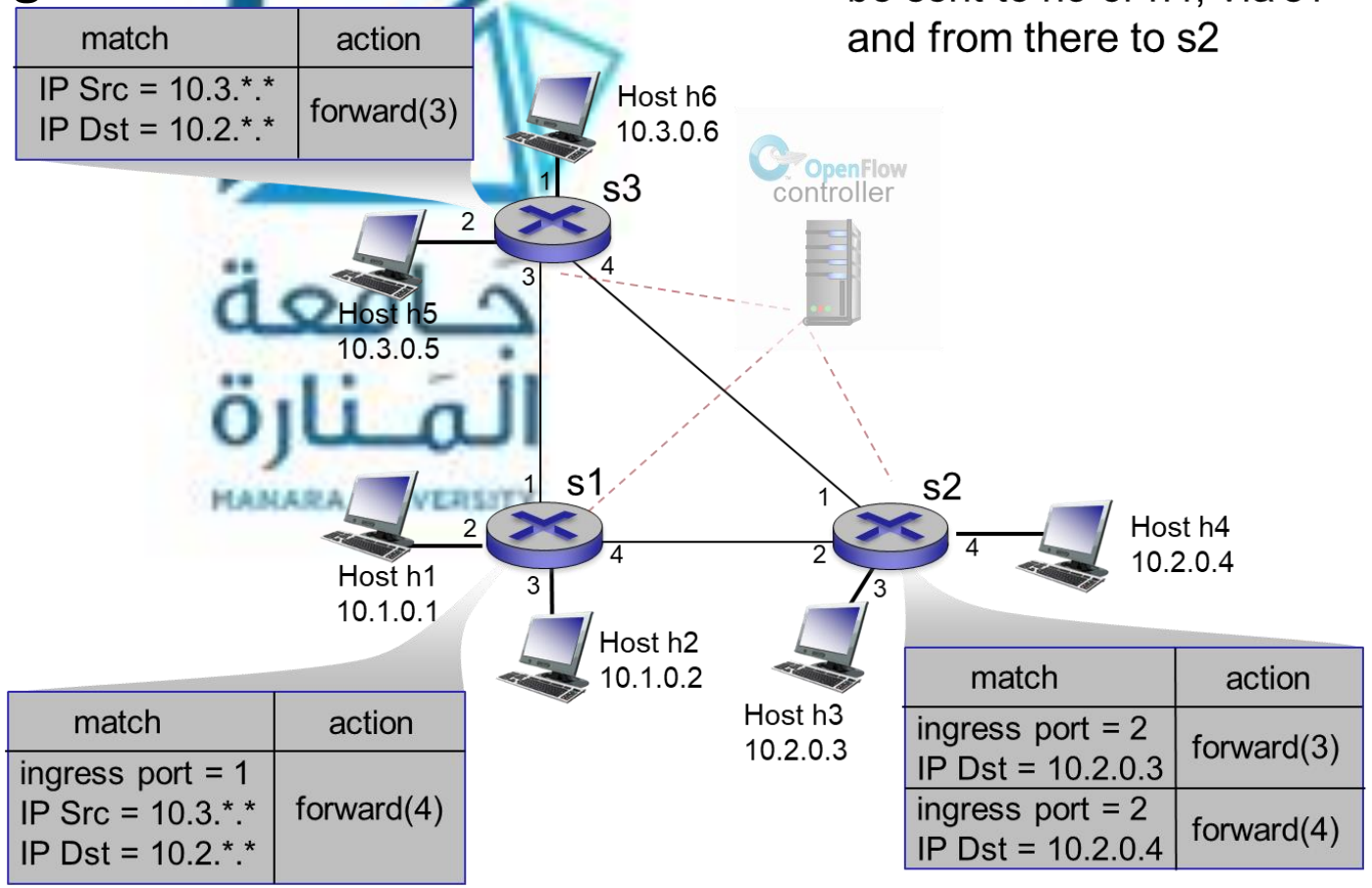
*do not forward (block) all datagrams destined to TCP port 22*

| 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