

Computer Networks and Network Security: Lab

Assignment Five: Routing and Forwarding

22920202202763 陈增辉

P1

a. Suppose that routers were subjected to conditions that might cause them to fail fairly often. Would this argue in favor of a VC or datagram architecture? Why?

这个观点将更赞成使用 datagram architecture.

当路由比较容易出错时, 因为 virtual-circuit 在发送数据时使用分组交换, 但传输时数据包携带的是源地址到目的地址的虚电路信息, 传输时先进行连接, 当电路中一个数据包传输失败时, 这一连接后面的所有数据包全都无法传输, 需要重新建立虚电路。

而数据报形式的数据包依靠目的地址寻路, 传输时不建立连接, 每个分组独立选路, 当出错时只需让出错数据包重传。

所以, 当路由出错率高时, 选用数据报方式可以节省传输量, 甚至出现虚电路方式多次尝试都无法完整传输的优势。

b. Suppose that a source node and a destination require that a fixed amount of capacity always be available at all routers on the path between the source and destination node, for the exclusive use of traffic flowing between this source and destination node. Would this argue in favor of a VC or datagram architecture? Why?

这会支持 VC。

当使用 datagram 时, 每个包都会独立寻址, 而节点能控制发送的 packet, 无法控制接收的 packet (只能控制是否接收某个节点的 packet), 这样就会出现被选择多的节点在某一时刻收到的 packet 多, 而不受欢迎的节点收到的少, 难以控制每个节点剩余的容量。

而 VC 在进行连接时, 可以占用全部带宽, 传输之前了解虚电路状态, 从而控制发送总量。

c. Suppose that the links and routers in the network never fail and that routing paths used between all source/destination pairs remains constant. In this scenario, does a VC or datagram architecture have more control traffic overhead? Why?

datagram architecture 会有更多的控制连接开销。

在这种情况下, VC 不需要改变任何数据包的虚电路信息, 也不会遇到传输失败需要重传从而重新寻找虚电路的情况, 所以 control traffic overhead 是非常少的。

而 datagram 需要照常为每个包寻址, 花销不变。

P2

distance table entries

z	x	v	u	y	
	2	6	big	big	initially
	2	5	6	5	1 exchange

z	x	v	u	y	
	2	5	6	5	2 exchanges

P3

假设有 n 个节点， v 条边。

因为每个节点都需要与其他节点相连，所以边的数目至少为 $n - 1$ 条，此时考虑最坏的情况，即 $n - 1$ 条边全部只能找到一个新节点，此时需要 $n - 1$ 轮迭代。而当 $v \geq n$ 时，每增加一条边，迭代轮数至少减少1轮，故最大迭代轮数：

$n - 1 - (v - n + 1)$ 当 $n - 1 \leq v < 2n$
 2 当 $v \geq 2n$

P4

a. Give x's distance vector for destinations w, y, and u.

x	w	y	u	
	2	5	big	initial
	2	4	big	1 exchange
	2	4	7	2 exchanges

x -> w : x - w 2
 x -> y : x - w - y 4
 x -> u : x - w - ... - u 7

b. Give a link-cost change for either $c(x,w)$ or $c(x,y)$ such that x will inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

当修改 $c(x,w)$ 时， 改为 1， $c(x,u)$ 变为 6， $x - w - \dots - u$
 当修改 $c(x,y)$ 时， 只有改到 ≤ 0 时才会发生变化，故无法修改。

c. Give a link-cost change for either $c(x,w)$ or $c(x,y)$ such that x will not inform its neighbors of a new minimum-cost path to u as a result of executing the distance-vector algorithm.

如 b，将 $c(x,y)$ 改为 > 0 或将 $c(x,w)$ 改为 > 2 的任意值均可满足要求。

P5

count-to-infinity problem 是由拓扑中的路由环路导致的，出现环路时，寻路时可能在环路中无限绕圈，导致 cost 无限增大。当我们减短一个链接的 cost 时，这个链接会立刻通知它的邻居更新链路，而新的链路只有可能比原来的链路短，邻居接受到信息后，只有 cost 小于原 cost 才会更改，不会出现如 $c(x,y) = c(y,x)$ 时，x 和 y 选择不同路径的情况。新增一条链路相当于将无穷大减小为有限，同理。

只有将 cost 增大，x 先更新到 y 的 cost，y 更新到 x 的 cost 时选择还未更新的过期的节点提供的路径时会有 count-to-infinity。

P6

a. When the distance vector routing is stabilized, router w, y, and z inform their distances to x to each other. What distance values do they tell each other?

```
router z: => w d(z,x) = big, => y d(z,x) = 6
router w: => y d(w,x) = big, => z d(w,x) = 5
router y: => w d(y,x) = 4, => z d(y,x) = 4
```

b. Now suppose that the link cost between x and y increases to 60. Will there be a count-to-infinity problem even if poisoned reverse is used? Why or why not? If there is a count-to-infinity problem, then how many iterations are needed for the distance-vector routing to reach a stable state again? Justify your answer.

当 $c(x,y)$ 由 4 变到 60 时，y 到 x 的最短路径不可用，开始改变，而 z 告知 y 的 $d(z,x) = 6$ ，y 就会选择经过 z 到达 x，将 $d(y,x)$ 改为 9 并告知 z w，z 和 w 得知后也进行修改，就进入无限计数。

iteration 1: $d(y,x) \Rightarrow 9$, $d(w,x) \Rightarrow 10$, $d(z,x) \Rightarrow 11$

iteration 2: $d(y,x) \Rightarrow 14$, $d(w,x) \Rightarrow 15$, $d(z,x) \Rightarrow 16$

可以看出，每轮 $d(z,x)$ 增加 5，当 $d(z,x) > 50$ 时，z 会选择 50 结束迭代， $(50-6) / 5 = 8 \dots 4$ ，共需要 9 轮迭代。

c. How do you modify $c(y,z)$ such that there is no count-to-infinity problem at all if $c(y,x)$ changes from 4 to 60?

在 y 和 w 进行寻路时，都没有用到 $c(y,z)$ ，而在 y 因为问题 b 中的原因利用 $c(y,z)$ 寻路时，只需将 $c(y,z)$ 设置为 $c(y,z) + c(y,x) > 60$ ，则 y 不会选用，此时 y 将 $d(y,x)$ 改为 60，w 改为 61，z 改为 50，立刻稳定。

即 $c(y,z) > 10$ 。

P7

a. Router 3c learns about prefix x from which routing protocol: OSPF, RIP, eBGP, or iBGP?

3c 从 4c 得知，而前缀 x 由 iBGP 或 eBGP 得知，同一 AS 用 iBGP，不同 AS 用 eBGP。所以，本题为 eBGP。

b. Router 3a learns about x from which routing protocol?

3a 从 3b 得知，同上，为 iBGP。

c. Router 1c learns about x from which routing protocol?

1c 从 3a 得知，同上，为 eBGP。

d. Router 1d learns about x from which routing protocol?

1d 从 1a 得知，同上，为 iBGP。

P8

a. Will I be equal to I1 or I2 for this entry? Explain why in one sentence.

Be equal to I1.

在使用 RIP 的 AS1 里面选择 I1 使得跳数或者跃点数最少。

b. Now suppose that there is a physical link between AS2 and AS4, shown by the dotted line. Suppose router 1d learns that x is accessible via AS2 as well as via AS3. Will I be set to I1 or I2? Explain why in one sentence.

I2.

这种情况下选择 I2 在经过的 AS 数量相同的情况下距离下一个 AS 的跳数最少。

c. Now suppose there is another AS, called AS5, which lies on the path between AS2 and AS4 (not shown in diagram). Suppose router 1d learns that x is accessible via AS2 AS5 AS4 as well as via AS3 AS4. Will I be set to I1 or I2? Explain why in one sentence.

I1.

这种情况下选择 I1 经过的 AS 数少。

P9

在网络层中每个 AS 拥有其自治性。ISP C 在向 ISP B 发送它的路径信息时，可以只在 its East Coast peering point 的边缘节点向 neighbor 广播它到 D 的 path，而在 West Coast 不广播。这样 ISP B 就仅仅收到通过 C 的 East Coast peering point 到达 D 的 BGP 信息，只能选择 B - East C - D。

P10

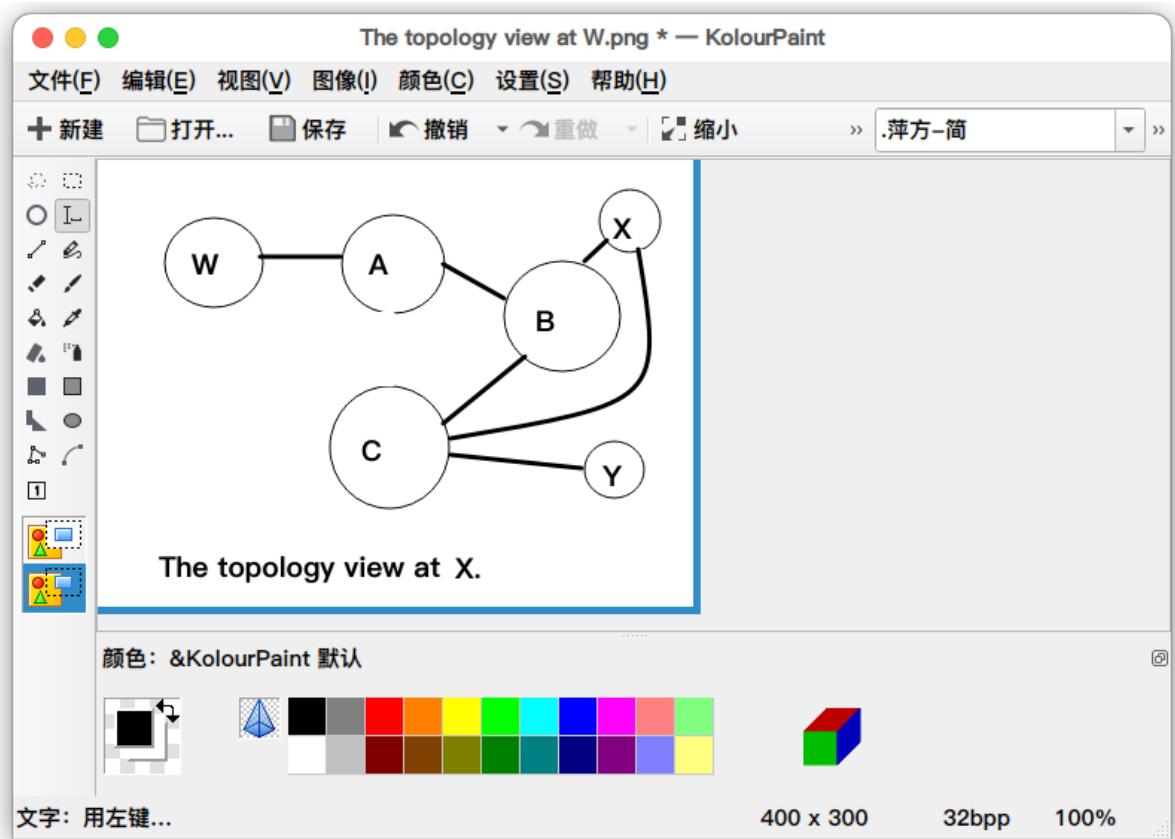
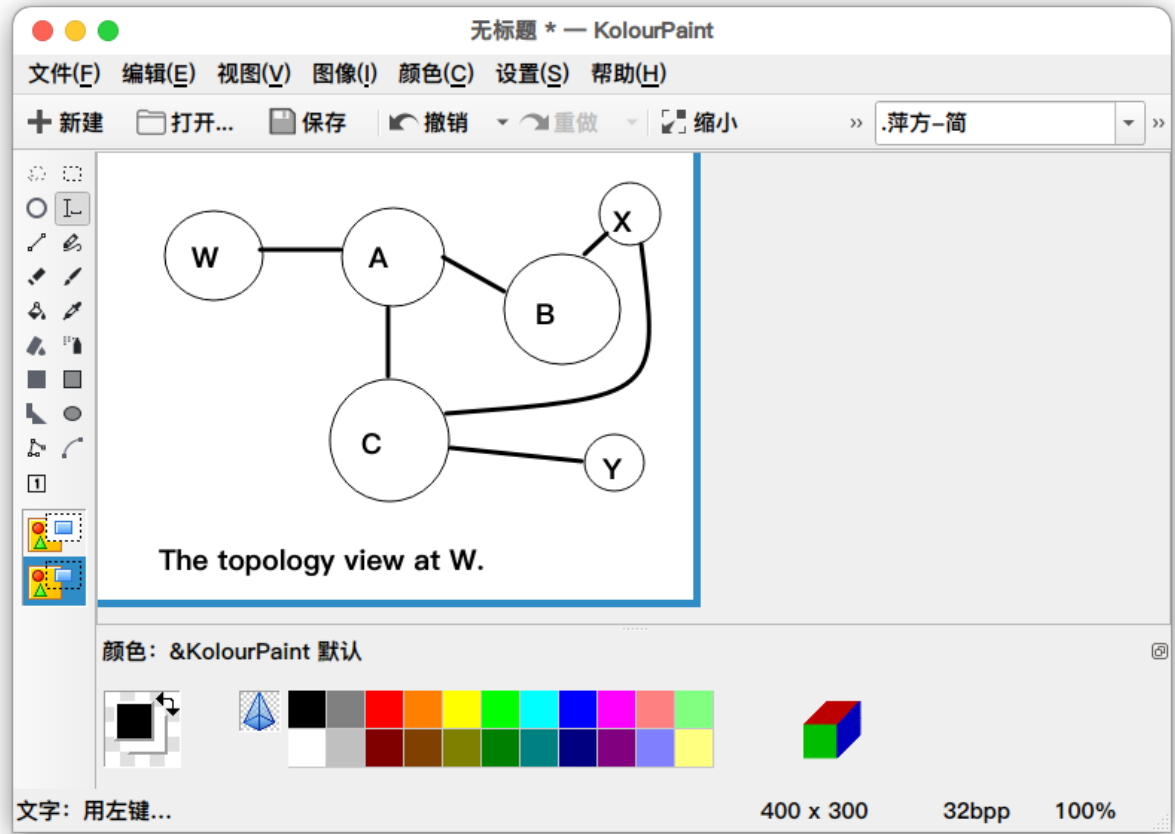
在此问题中，A、B、C 分别作为 W、X、Y 的 provider，相互之间为 peer 关系。

对 Y，从 $X - B - C$ 、 $W - A - B - C$ 的路径，B 不会提供给 C；

在 W，以 Y 举例，Provider C 收到 Y 提供的路径 C-Y，将其发送给所有邻居 A、B、X，为了防止其他 peer 利用自己的流量进行无法收费的转发，B 仅将这一路径发给 X，A 仅将这一路径发给 W。X-B、X-C 同理。

X 同理。

如下图所示：



P11

Bittorrent 文件共享采用了这种方式。

在首次打开某些磁力下载软件时，会出现类似“使用本协议即代表您同意相关的文件共享法规，请不要在软件中传输违规数据”的提示，这是因为磁力链接是一种在对等点间共享数据的传输方式，例如有3个 peer — W、X、Y，数据块可以先从 W 发到 X，再从 X 转发给 Y，实现题目中描述的方式。

P12

a. Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.

Link Interface 0: 224.0.0.0/10
Link Interface 1: 224.64.0.0/16
Link Interface 2: 224.65.0.0/7
Link Interface 3: otherwise

Forwarding Table
Destination NextHop Flag Interface TunnelID
225.128.17.119 225.64.195.60 HU 1 0x0

b. Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses

nextHop 的地址为 11100001 01000000 11000011 00111100
进行最长前缀匹配：
Link Interface 0: 匹配10位，不符合
Link Interface 1: 匹配16位，符合
由此选定 Link Interface 1，继续匹配，最终匹配地址为：225.64.195.60/16