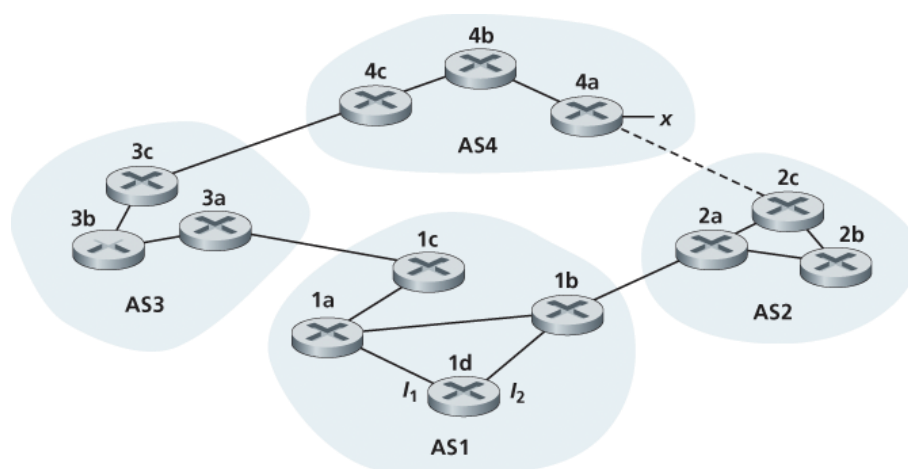


## 计算机网络及应用（2020 秋）第十周作业

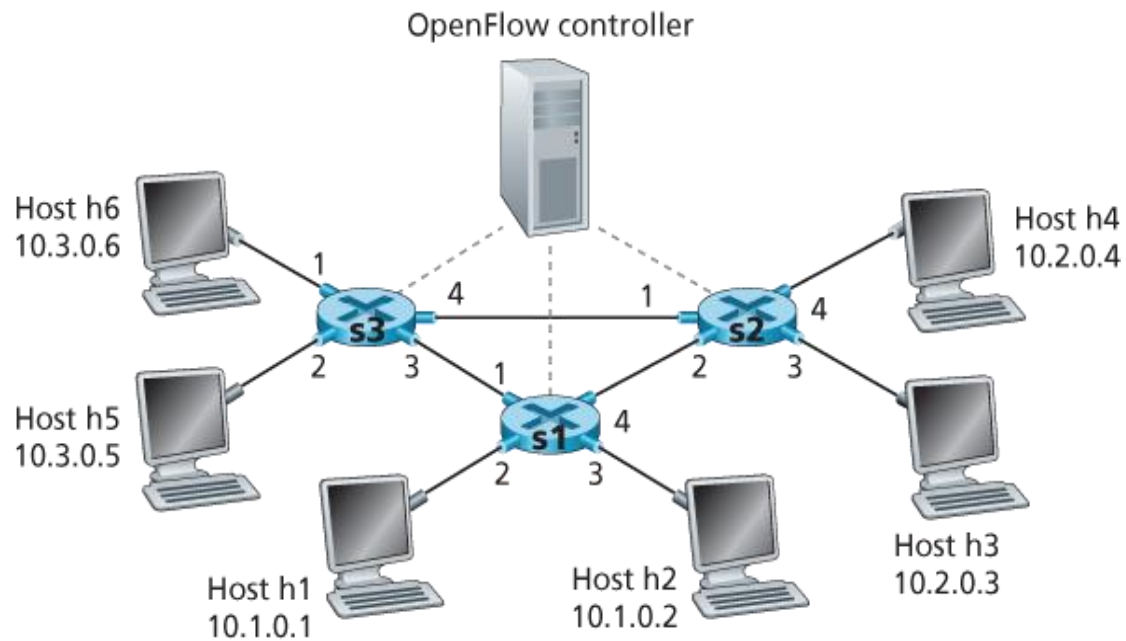
请大家通过网络学堂以 pdf 格式提交，命名为：学号\_姓名\_班.pdf。

作业纪律：禁止抄袭，抄袭双方都记 0 分。迟交作业最终分数=卷面评分\*(1-0.1\*迟交天数)。

1. 考虑下图所示的网络。假定所有的 AS 正在运行 OSPF 作为其 AS 内部路由选择协议，任意两个节点间直接的链路连接开销都是 1。假定 AS 间路由选择协议使用的是 eBGP 和 iBGP。假定最初在 AS2 和 AS4 之间不存在物理链路。一旦路由器 1d 知道了 x 的情况, 它将一个表项 (x, I) 放入它的转发表中。



- a) 对这个表项而言, I 将等于  $I_1$  还是  $I_2$ ? 请说明理由。  
 $I_1$ 。因为 1d 只能通过 1c 到达 x, 而  $I_1$  在 1d 到 1c 的最短路径上。
  - b) 现在假定在 AS2 和 AS4 之间有一条物理链路, 显示为图中的虚线。假定路由器 1d 知道经 AS2 以及经 AS3 能够访问到 x。I 将设置为  $I_1$  还是  $I_2$ ? 请说明理由。  
 $I_2$ 。按照“热土豆路由选择”, 1d 会选择域内开销最小的本地网关, 也就是 1b, 而  $I_2$  就在这条路径上。
2. 现有 SDN 网络, S1、S2、S3 为 SDN 交换机, 如图所示, 假设以跳数作为链路距离的度量。



- a) 请写出 S3 中主机 h5、h6 向主机 h1、h2、h3、h4 发送数据的流表部分。
    - src = 10.3.0.\*, dest=10.2.0.\*->forward(4)
    - src = 10.3.0.\*, dest=10.1.0.\*->forward(3)
  - b) 如果 S3 与 S2 之间的直接连接的链路断开, 请写出更新后 S3 中主机 h5、h6 向主机 h1、h2、h3、h4 发送数据的流表部分。
    - src = 10.3.0.\*, dest=10.2.0.\*->forward(3)
    - src = 10.3.0.\*, dest=10.1.0.\*->forward(3)
  - c) 在 b 从链路断开到流表完成更新的过程中, 请结合 PPT, 写出哪些报文流跨越 SDN 控制器的北向 API, 哪些报文流跨越 SDN 控制器的南向 API。
    - 1、S3 经历了链路故障, 使用 **OpenFlow 端口状态报文**向 SDN 控制器通报链路状态的更新, **跨越了南向 API**。
    - 2、SDN 控制器接收 OpenFlow 报文, 更新链接状态信息, **跨越了南向 API**。
    - 3、实现 Dijkstra 链路状态路由选择的应用程序先前进行了注册, 在任何链路状态改变时都将得到通告。 **跨越了北向 API**。
    - 4、Dijkstra 路由算法访问网络图信息、控制器中的链路状态信息, 从而计算新的路由。 **跨越了北向 API**。
    - 5、链接状态路由选择应用与流表管理器交互, 流表管理器决定更新的流表。 **跨越了北向 API**。
    - 6、流表管理器使用 OpenFlow 协议在需要更新的交换机中安装新表。 **跨越了南向 API**。
3. 考虑在管理服务器和被管设备之间发生通信的两种方式: 请求响应方式和陷阱方式。从以下方面考虑这两种方式的优缺点: ①开销; ②当异常事件出现时通知的时间; ③对于管理实体和设备之间丢失报文的健壮性。
- 开销: 请求响应方式的开销更大, 原因之一是这一方式中既有管理服务器向被管设备的请求, 也有反过来的相应; 而陷阱方式中只有单向通信。而且请求响应方式需要轮询, 而大部分情况下并不会异常事件, 所以频繁的轮询会浪费资源。

当异常事件出现时通知的时间：陷阱方式中，异常事件一旦出现就会通报给管理服务器。但是请求响应方式中，需要等到下一次轮询，管理服务器才能够得知异常事件。

对于管理实体和设备之间丢失报文的健壮性：陷阱方式中，报文丢失后被管设备不会重发；而请求响应方式中，如果请求或者响应报文丢失，管理服务器能够察觉到，并重新请求。

#### 4. 针对二维奇偶校验。

a) 说明(举一个不同于 PPT 的例子)能够纠正和检测单比特差错。

原报文：

1	1	1	1	0
1	0	1	1	1
1	1	0	0	0
1	0	0	1	0
0	0	0	1	1

出错的报文：

1	1	1	1	0
1	1	1	1	1
1	1	0	0	0
1	0	0	1	0
0	0	0	1	1

通过第二行的校验码，可以确定第二行有个单比特差错；通过第二列的校验码，可以确定第二列有个单比特差错。这就能找到第二行、第二列处出错的单比特了。

b) 说明某些双比特差错能够被检测但不能纠正。

原报文：

1	1	1	1	0
1	0	1	1	1
1	1	0	0	0
1	0	0	1	0
0	0	0	1	1

出错的报文：

1	1	1	1	0
1	1	0	1	1
1	1	0	0	0
1	0	0	1	0
0	0	0	1	1

通过第二列的校验码，可以确定第二列有个单比特差错；通过第三列的校验码，可以确定第三列有个单比特差错。但是不能确定更多信息，无法锁定出错的具体位置，也就无法纠正了。

c) 说明某些偶数个差错不能够被检测。

原报文：

1	1	1	1	0
1	0	1	1	1

1	1	0	0	0
1	0	0	1	0
0	0	0	1	1

出错的报文:

1	1	1	1	0
1	1	0	1	1
1	0	1	0	0
1	0	0	1	0
0	0	0	1	1

报文中有 4 个差错，但是校验码都是正常的，从校验码中无法检测出差错。

5. 考虑 5 比特生成多项式,  $G=10101$ , 并且假设  $D$  的值为下列情况,  $R$  的值是什么?

1010000010

0010

$$\begin{array}{r}
 \begin{array}{c} 10101 \end{array} \overline{) \begin{array}{c} 10001010 \\ 100000000000 \end{array}} \\
 \underline{10101} \phantom{000000000000} \\
 10001 \phantom{000000000000} \\
 \underline{10101} \phantom{000000000000} \\
 10000 \phantom{000000000000} \\
 \underline{10101} \phantom{000000000000} \\
 10100 \phantom{000000000000} \\
 \underline{10101} \phantom{000000000000} \\
 \underline{0010} = R
 \end{array}$$

1000011101

0000

$$\begin{array}{r}
 \begin{array}{c} 10101 \end{array} \overline{) \begin{array}{c} 1000011101 \\ 100001110000 \end{array}} \\
 \underline{10101} \phantom{000000000000} \\
 10111 \phantom{000000000000} \\
 \underline{10101} \phantom{000000000000} \\
 10101 \phantom{000000000000} \\
 \underline{10101} \phantom{000000000000} \\
 \underline{00000} = R
 \end{array}$$

0101111010

0000

$$\begin{array}{r}
 100/00000 \\
 101 \overline{) 0101110/00000} \\
 \underline{1010} \phantom{00} \\
 10101 \phantom{00} \\
 \underline{10101} \phantom{00} \\
 \hline
 000000 = R
 \end{array}$$

0110101100  
0000

$$\begin{array}{r}
 11/000000 \\
 101 \overline{) 0110101/000000} \\
 \underline{1010} \phantom{00} \\
 1111 \phantom{00} \\
 \underline{1010} \phantom{00} \\
 10101 \phantom{00} \\
 \underline{10101} \phantom{00} \\
 000000 = R
 \end{array}$$