

2050865 黄秀铭
- 1. 802.11

自组织网络 基础结构型网络

2. 时分复用, 时分复用, 时分复用, 时分复用

3. ~~192.168.3.0~~, 192.168.3.0, 192.168.3.31

4. 单播, 组播, 广播,

192.168.1.1

~~192.168.1.1~~ 244.0.0.0

192.168.1.254

5. 广播, ~~单播~~

二. 1. ✓ 2. ✓ 3. ✓ 4. ✓ 5. X
6. ✓ 7. ✓ 8. X 9. ✓ 10. X
11. X 12. X 13. ✓ 14. ✓

三. 1. ACD 2. AD 3. AC 4. AC 5. AD
6. ABCD 7. BC 8. ACD 9. AB 10. C
11. AB 12. AB 13. AC 14. CD



四. 4.1

1. 答: CSMA/CD为载波侦听多路访问/冲突检测, 是一种随机介质接入协议, 过程如下:

(1) 若有一节点想要发送数据, 则要先进行侦听。确认信道上没有能量时再进行传输。

(2) 在传输数据时, 发送方要持续检测在发送过程中有没有发生冲突。若无冲突, 则将传输进行完; 若有冲突, 则~~将传输~~立即中止传输。并广播一个~~阻塞~~阻塞信号。

(3) 若因冲突中止传输, 采用二进制指数退避算法。

等待 $R \times 2^i \times D$ 之后, 再重新侦听, 至合适时机重传。

2. 答: 不能。无线局域网可能有隐藏站问题, 一个节点未必能监测到所有冲突。

3. 答: 是 CSMA/CA。

(1) 发送站先给接收节点发送一个控制报文。

(2) 接收节点广播一个接收宣告。

(3) 发送节点给接收节点传输数据。

(4) 接收节点广播一个传输结束的报文。



1. UDP采用无连接服务,传输开始前无需建立连接;
TCP采用面向连接服务,传输开始前要建立连接,传输结束后要断开连接。

2. 答:不能。因为UDP提供不可靠的服务,很可能丢包、乱序等等。传输层需要能提供可靠的服务。

3. 答: ~~当~~当发送方检测到丢包时,即有些包在规定时间内未收到ACK,则~~认为~~认为发生拥塞。此时主动减小拥塞窗口,限制发送方的发送速率的大小。



2080863 黃廣銘

解: generator 可表示为 10011, 计算出 CRC 为 0100.

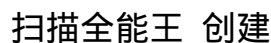
Handwritten binary long division:

$$\begin{array}{r}
 10011 \overline{) 1101101101100} \\
 \underline{10011} \\
 010000 \\
 \underline{10011} \\
 000111 \\
 \underline{00000} \\
 001110 \\
 \underline{00000} \\
 011101 \\
 \underline{10011} \\
 011101 \\
 \underline{10011} \\
 011100 \\
 \underline{10011} \\
 011111 \\
 \underline{10011} \\
 011001 \\
 \underline{10011} \\
 010101 \\
 \underline{10011} \\
 001100 \\
 \underline{00000} \\
 011001 \\
 \underline{10011} \\
 010101 \\
 \underline{10011} \\
 001100 \\
 \underline{00000} \\
 011001 \\
 \underline{10011} \\
 010101 \\
 \underline{10011} \\
 001100 \\
 \underline{00000} \\
 011001 \\
 \underline{10011} \\
 010101 \\
 \underline{10011} \\
 001101 \\
 \underline{00000} \\
 011010 \\
 \underline{10011} \\
 010010
 \end{array}$$

Remainder: 000100

Verification:

$$\begin{array}{r}
 010010 \\
 \times 10011 \\
 \hline
 000010 \\
 000000 \\
 \hline
 000100
 \end{array}$$



- 5.2. 2050865 黄雅雅
1. 解:
- ① 选取一个起点, 初始化其他点到起点的距离, 达不到的置为 ∞ .
 - ② 每次从剩余节点中挑选出一个距离起点距离最短的节点, 加入到已经确定的节点集合中。
 - ③ 每次从剩余节点中挑选一个点加入时, ~~检查~~检查剩余节点的每个点。如果形成了一条更短的路径, 则将该节点距离更新为“上一节点到起点距离 + 上一节点到该节点距离”。
 - ④ 不断重复②③, 直到所有节点的最短路径都被确定。

2. 解: ~~以~~以 ~~A(B, d)~~ $A(B, d)$ 表示目的地, 目的地的上一任节点, 起点到目的地的最短距离。

当前结点	剩余结点
5(-, d=0) 5(-, d=0)	1(-, d= ∞) 2(-, d= ∞) 3(5, d=6) 4(5, d=3) 6(5, d=8) 7(5, d=3) 8(-, d= ∞)
4(5, d=3)	1(-, d= ∞) 2(4, d=7) 3(4, d=5) 6(4, d=5) 7(5, d=3) 8(-, d= ∞)
7(5, d=3)	1(-, d= ∞) 2(4, d=7) 3(4, d=5) 6(4, d=5) 8(7, d=6)
3(4, d=5)	1(3, d=8) 2(3, d=6) 6(4, d=5) 8(7, d=6)
6(4, d=5)	1(3, d=8) 2(3, d=6) 8(7, d=6)
2(3, d=6)	1(3, d=8) 8(7, d=6)
8(7, d=6)	1(3, d=8)
1(3, d=8)	



所以, ~~路由表~~ 为节点5到其他节点的路径为:

2050865 黄乃铭

到节点1: $5 \rightarrow 4 \rightarrow 3 \rightarrow 1$

到2: $5 \rightarrow 4 \rightarrow 3 \rightarrow 2$

到3: $5 \rightarrow 4 \rightarrow 3$

到4: $5 \rightarrow 4$

到6: $5 \rightarrow 4 \rightarrow 6$

到7: $5 \rightarrow 7$

到8: $5 \rightarrow 7 \rightarrow 8$

所以, 节点5的路由表为:

目标节点	下一跳
1	4
2	4
3	4
4	4
6	4
7	7
8	7



3. 解:

2050865 黄彦铭

当前结点	剩余结点
5(-, d=0)	1(-, d=∞) 2(-, d=∞) 3(5, d=6) 6(5, d=8) 7(5, d=3) 8(-, d=∞)
7(5, d=3)	1(-, d=∞) 2(-, d=∞) 3(5, d=6) 6(7, d=7) 8(7, d=6)
3(5, d=6)	1(3, d=9) 2(3, d=7) 6(7, d=7) 8(7, d=6)
8(7, d=6)	1(3, d=9) 2(3, d=7) 6(7, d=7)
2(3, d=7)	1(3, d=9) 6(7, d=7)
6(7, d=7)	1(3, d=9)
1(3, d=9)	

所以5到各点路径:

到1: 5 → 3 → 1

到2: 5 → 3 → 2

~~到6: 5 → 7 → 6~~

~~到7: 5 → 7~~

到3: 5 → 3

到6: 5 → 7 → 6

到7: 5 → 7

到8: 5 → 7 → 8

所以,更新后的路由表为

目标节点	下一跳
1	3
2	3
3	3
6	7
7	7
8	7

