


知识点一 IP分片

IP为什么要分片？不同链路的MTU(Maximum Transmission Unit)（填空）

M T U for diverse Links

Protocol	MTU(Byte)	
Hyperchannel	65535	
Token Ring(16 Mbit/s)	17914	理论上限
Token Ring (4 Mbit/s)	4464	特别强调
FDDI	4352	局域网所能承载的data部分
1500 Ethernet		即IP数据报总长度
X.25	576	
532 PPP		

IP如何分片和重组？(填空)

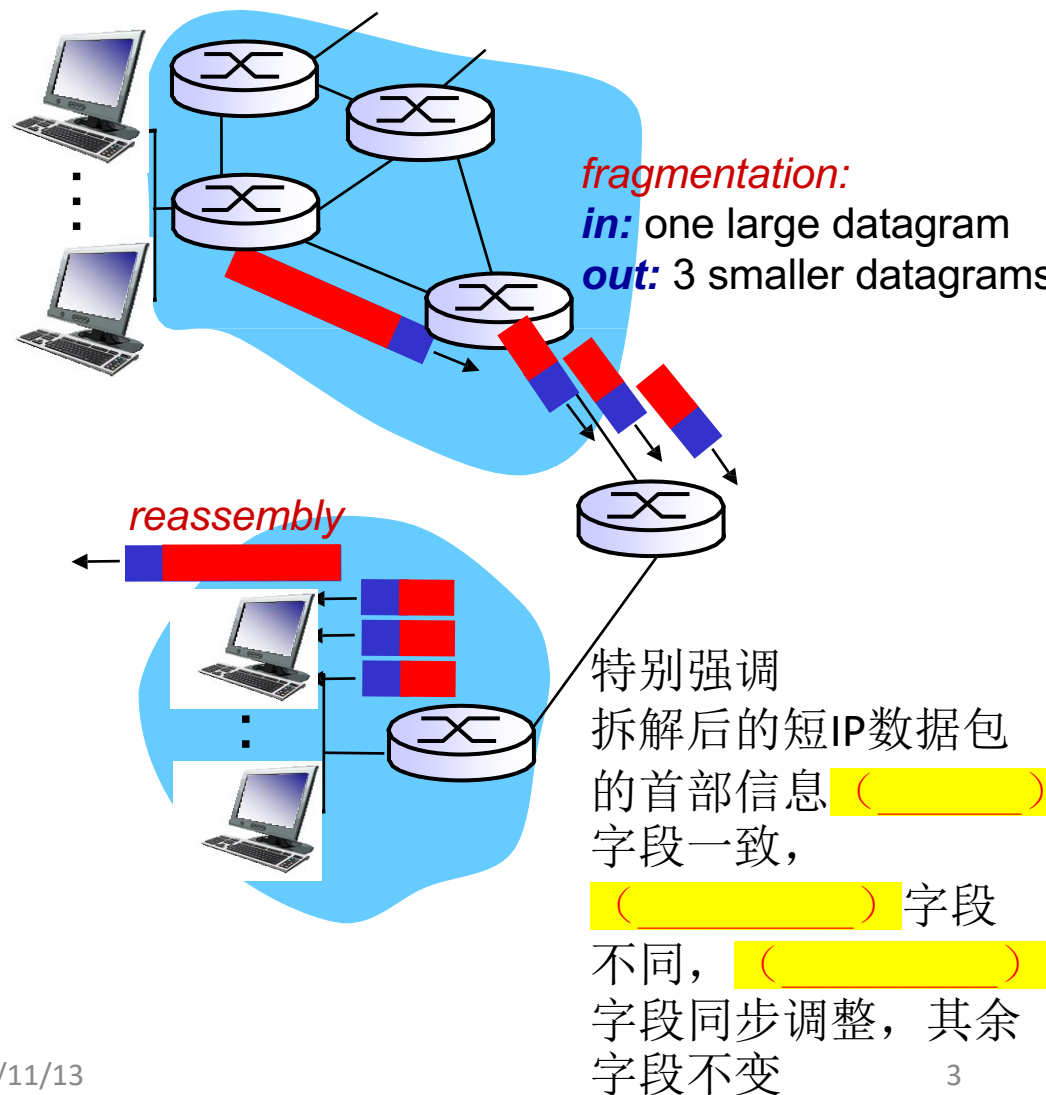


- 网络链路层有**MTU**限制 (max. transm. unit) - 最大可能的链路层帧能承载的数据量

- 不同的链路类型，不同 MTUs

- 大的**IP**数据报在网络中被划分为小“片”

- 一个数据报变为几个数据报
- 仅在 ()
- IP** 报头标识符字段用来对相关的片进行识别、排序





IP 分片和重组示例（填空并写出求解过程）

length	ID	fragflag	offset	
=4000	=x	=0	=0	

把一个大的数据报分为几个更小的数据报

length	ID	fragflag	offset	
=1500	=x	()	()	

length	ID	fragflag	offset	
=1500	=x	()	()	

length	ID	fragflag	offset	
=1040	=x	()	()	

知识点二 子网和超网

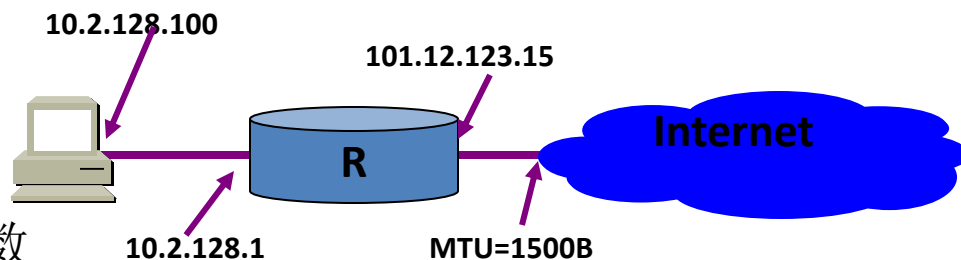
*

在划分子网的情况下路由器转发分组的算法

- (1) 从收到的分组的首部提取 () D 。
- (2) 先用连接到该路由器的各网络的 () 和 D 逐位相“()”，看是否和该路由器所直连的网络地址匹配。若匹配，则将分组 () 交付。
否则就是 () 交付，执行(3)。
- (3) 若路由表中有 D 的 ~~特定~~ 主机路由，则将分组传送给指明的下一跳路由器；否则，执行(4)。
- (4) 对路由表中的每一行的 () 和 D 逐位相“()”，若其结果与该行的目的网络地址匹配，则将分组传送给该行指明的 ~~下一跳~~ 路由器；否则，执行(5)。
- (5) 若路由表中有一个 () 路由，则将分组传送给路由表
中所指明的该路由器；否则，执行(6)。
- (6) 报告转发分组出错

应用题

- 某主机的MAC地址为00-15-C5-C1-5E-28，IP地址为10.2.128.100（私有地址）。网络拓扑如图1，图2是该主机进行Web请求的1个以太网数据帧前80个字节的十六进制及ASCII码内容



数据帧的字节计数

一行为16个字节

图1 网络拓扑

0000	00 21	27 21	51 ee	00 15	c5 c1	5e 28	08 00	45 00
0010	01 ef	11 3b	40 00	80 06	ba 9d	0a 02	80 64	40 aa
0020	62 20	04 ff	00 50	e0 e2	00 fa	7b f9	f8 05	50 18
0030	fa f0	1a c4	00 00	47 45	54 20	2f 72	66 63	2e 68
0040	74 6d	6c 20	48 54	54 50	2f 31	2e 31	0d 0a	41 63

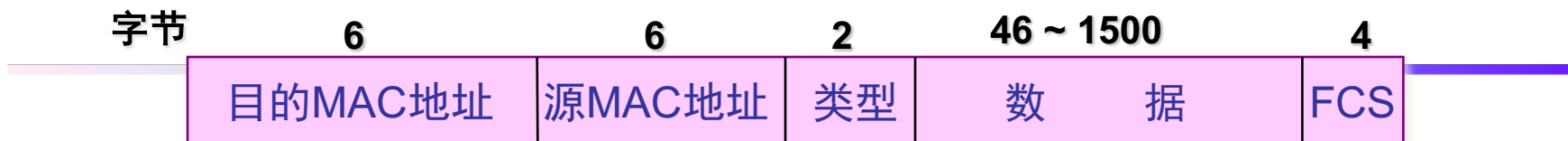


图3 以太网数据帧结构



图 4 IP分组头首部

-
- 参考图中的数据回答问题：
 - （1）**Web**服务器的**IP**地址是什么？该主机的默认网关的**MAC**地址是什么
 - （2）封装该协议请求报文的以太网帧的目的**MAC**地址是什么？
 - （3）该帧所封装的**IP**分组经过路由器**R**转发时，需修改**IP**分组头中的哪些字段？

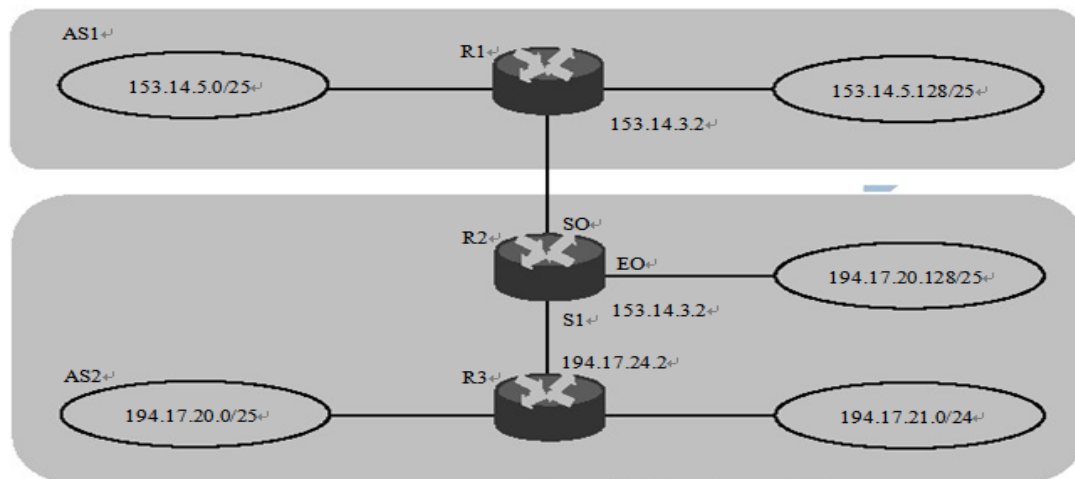
计算题（请给出计算过程）

- 某主机的IP地址是**180.80.77.55**，子网掩码是**255.255.252.0**。若该主机向其所在子网发送广播分组，则目的地址可以是？

知识点三 无分类编址最长前缀匹配

应用题

假设 Internet 的两个自治系统构成的网络如下图所示，自治系统 AS1 由路由器 R1 连接两个子网构成；自治系统 AS2 由路由器 R2、R3 互联并连接 3 个子网构成。各子网地址、R2 的接口名、R1、R2 与 R3 的部分接口 IP 地址如下图所示。



□请回答下列问题。

(1) 假设路由表结构如下表所示。请利用路由聚合技术，给出 R2 的路由表，要求包括到达题图中所有子网的路由，且路由表中的路由项尽可能少

目的网络	下一跳	接口
------	-----	----

(2) 若 R2 收到一个目的 IP 地址为 194.17.20.200 的 IP 分组，R2 会通过哪个接口转发该 IP 分组？

P5. Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	2
otherwise	3

- Provide a forwarding table that has five entries, uses longest prefix matching, and forwards packets to the correct link interfaces.
- Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

11100001 01000000 11000011 00111100
11100001 10000000 00010001 01110111

知识点四 ICMP协议

Traceroute 与 ICMP

*

- 源主机发送一系列 () 到目的主机
 - 第一个 TTL = ()
 - 第二个 TTL = (), 依次类推
 - 发送给不可达的端口号 (端口号 () 的 () 报文)
- 当第n个数据报到达第n个路由器时:
 - 路由器 () 数据报
 - 发送回源主机一个 ICMP 报文 (type 11, code 0) 表明因 () 终止
 - 报文包含路由器的名字和IP地址

- 源主机收到 ICMP 报文后, 源主机计算 RTT
- Traceroute 重复 3 次这样的过程

停止判据

- UDP 报文段最终到达目的主机
- 目的主机返回 ICMP " () " 分组 (type 3, code 3)
- 当源主机收到这样的 ICMP, 停止

知识点五 路由选择算法

A weighted undirected graph with 7 nodes (u, v, w, x, y, z) and 10 edges. The edges and their weights are: (u, x) weight 5, (u, w) weight 3, (u, v) weight 7, (w, x) weight 4, (w, y) weight 3, (w, z) weight 8, (x, y) weight 7, (x, z) weight 9, (v, y) weight 4, and (y, z) weight 2. A path of red edges is highlighted: u to x, u to w, w to v, and v to y.

距离向量选路算法（填空）

Distance Vector Routing Algorithm

- 所有节点和与它相邻的节点交换（ ）信息(DV)，更新DV信息，然后再把更新后的DV信息反馈给相邻节点
- ✓ () **Distributed**: 每个节点接受来自直接相邻节点的DV信息，迭代更新之后再把结果反馈回相邻节点
- ✓ () **Iterative**: 相邻节点的DV信息交换和DV更新持续进行，直到没有更多的信息可交换
- ✓ () **Asynchronous**: 不要求所有节点按时钟同步操作
- ✓ () **Self-terminated**: 没有要求停止的信号，但节点经过一定时间后自行停止

距离向量算法 (续)



Bellman-Ford方程 (动态规划中的最优性原理)

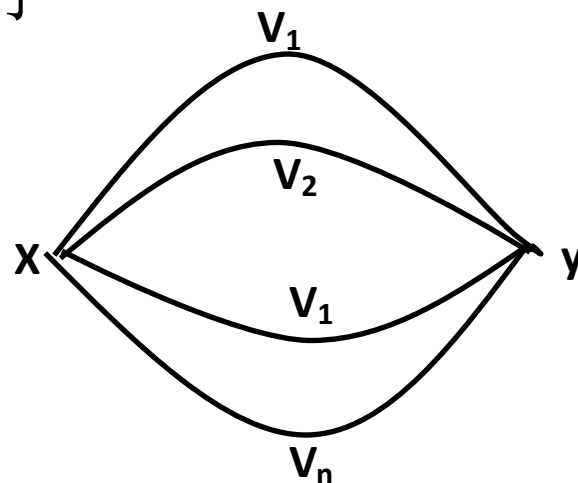
定义

$d_x(y) :=$ 从节点 x 到 y 的最短路径费用

Then

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

\min 遍历 x 的所有相邻节点 v
 $c(x,v)$ x 到相邻节点 v 的费用
 $d_v(y)$ v 到目的节点 y 的最短路径费用



距离向量算法-基本原理

核心思想:

- ❖ 每个节点根据需要向其邻居发送自己的距离向量
- ❖ 当节点x收到其邻居v发来的新的距离向量时, 使用B-F 方程更新自己的距离向量:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$

- ❖ 在适当条件下, 距离向量 $D_x(y)$ 最终将收敛至真实的最短路径费用 $d_x(y)$

分布式算法是学术界非常重要的研究方向

✓ 算法设计

✓ 性能分析(稳定性, 收敛性等)

(写出计算过程)

(写出计算过程)

node x
table

	cost to		
	x	y	z
from x.	0	2	7
from y.	∞	∞	∞
from z.	∞	∞	∞

node y
table

	cost to		
	x	y	z
from X.	∞	∞	∞
from y.	2	0	1
from Z.	∞	∞	∞

node z
table

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

	cost to		
	x	y	z
from x.	0	2	3
from y.	2	0	1
from z.	7	1	0

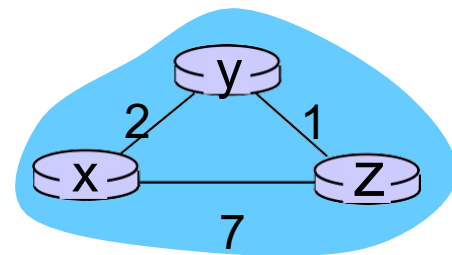
	cost to		
	x	y	z
from x.	0	2	7
from y.	2	0	1
from z.	7	1	0

	cost to		
	x	y	z
from x.	0	2	7
from y.	2	0	1
from z.	3	1	0

	cost to		
	x	y	z
from x.	0	2	3
from y.	2	0	1
from z.	3	1	0

	cost to		
	x	y	z
from x.	0	2	3
from y.	2	0	1
from z.	3	1	0

	cost to		
	x	y	z
from x.	0	2	3
from y.	2	0	1
from z.	3	1	0



距离向量
算法-举例
(续)

Time(时间轴)