

慕课网
imooc.com

慕课网
imooc.com

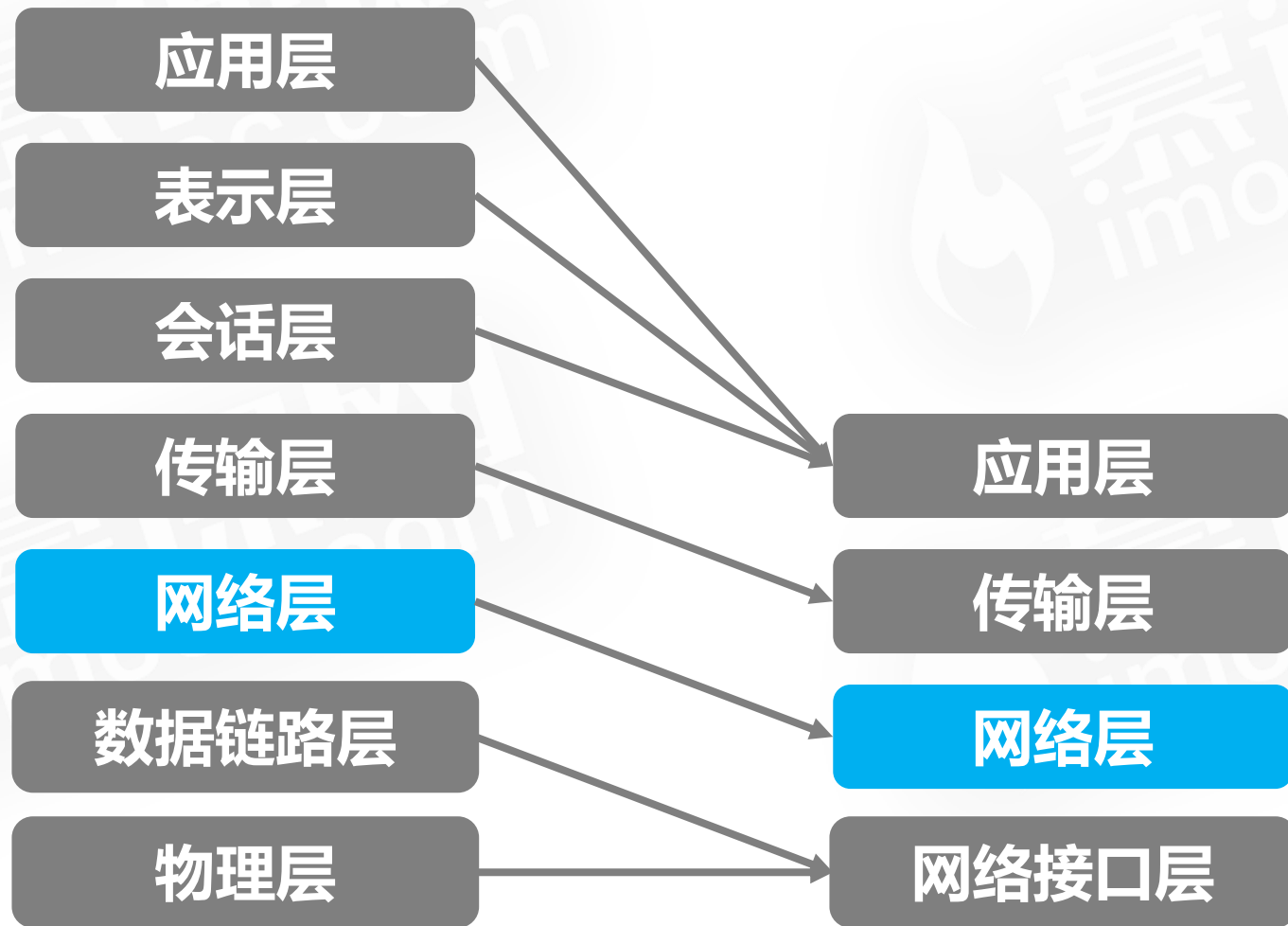
慕课网
imooc.com

慕课网
imooc.com

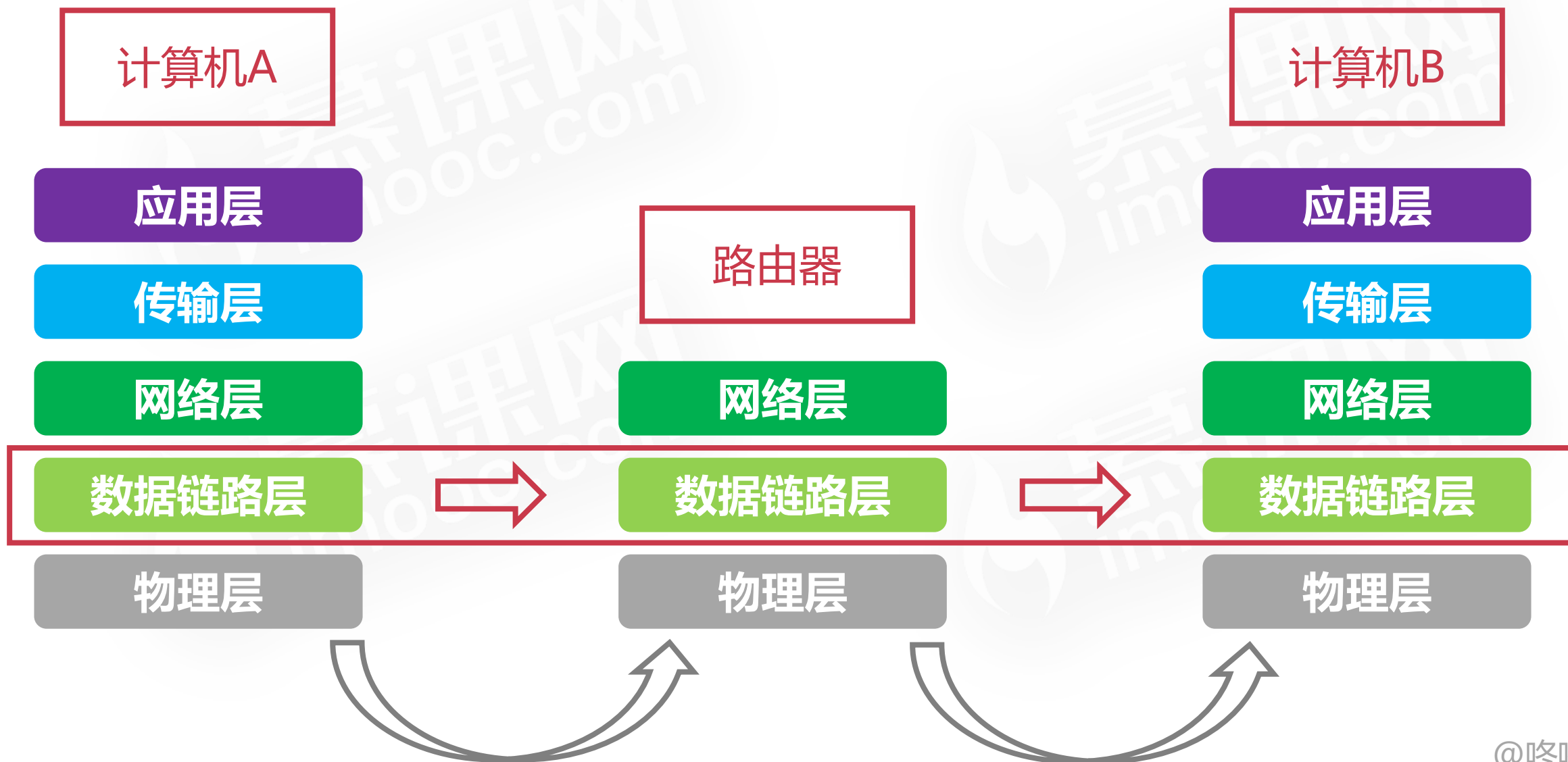
章节导学



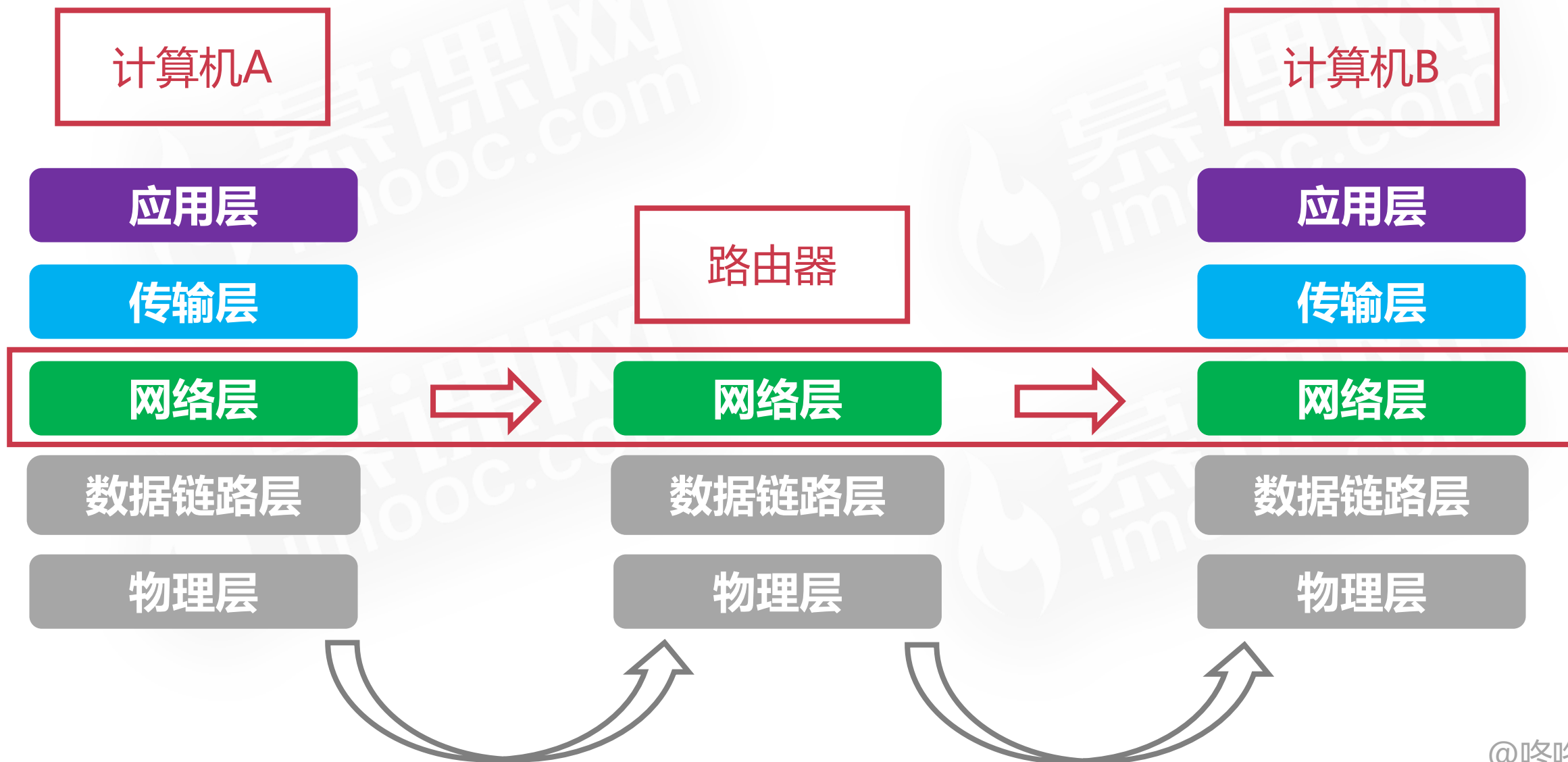
网络层概述



网络层概述



网络层概述

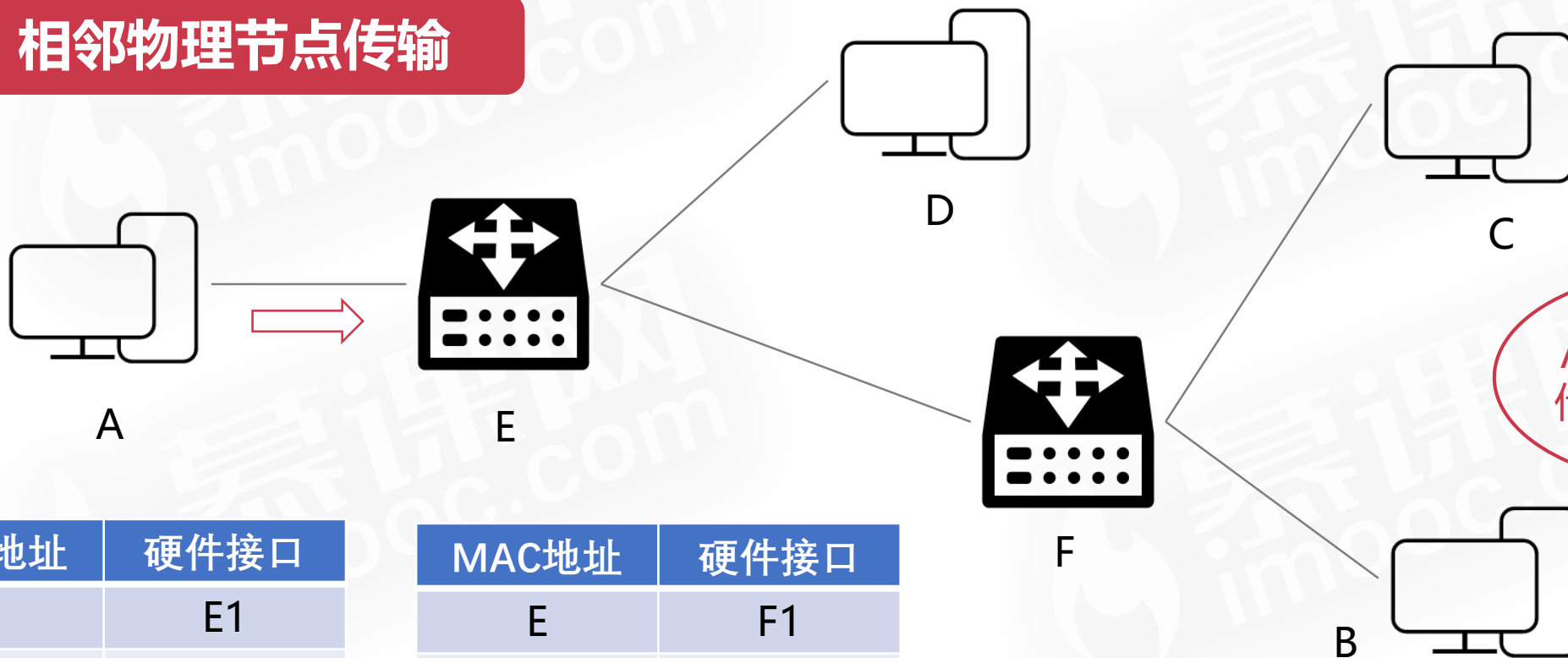


网络层概述



网络层概述

相邻物理节点传输

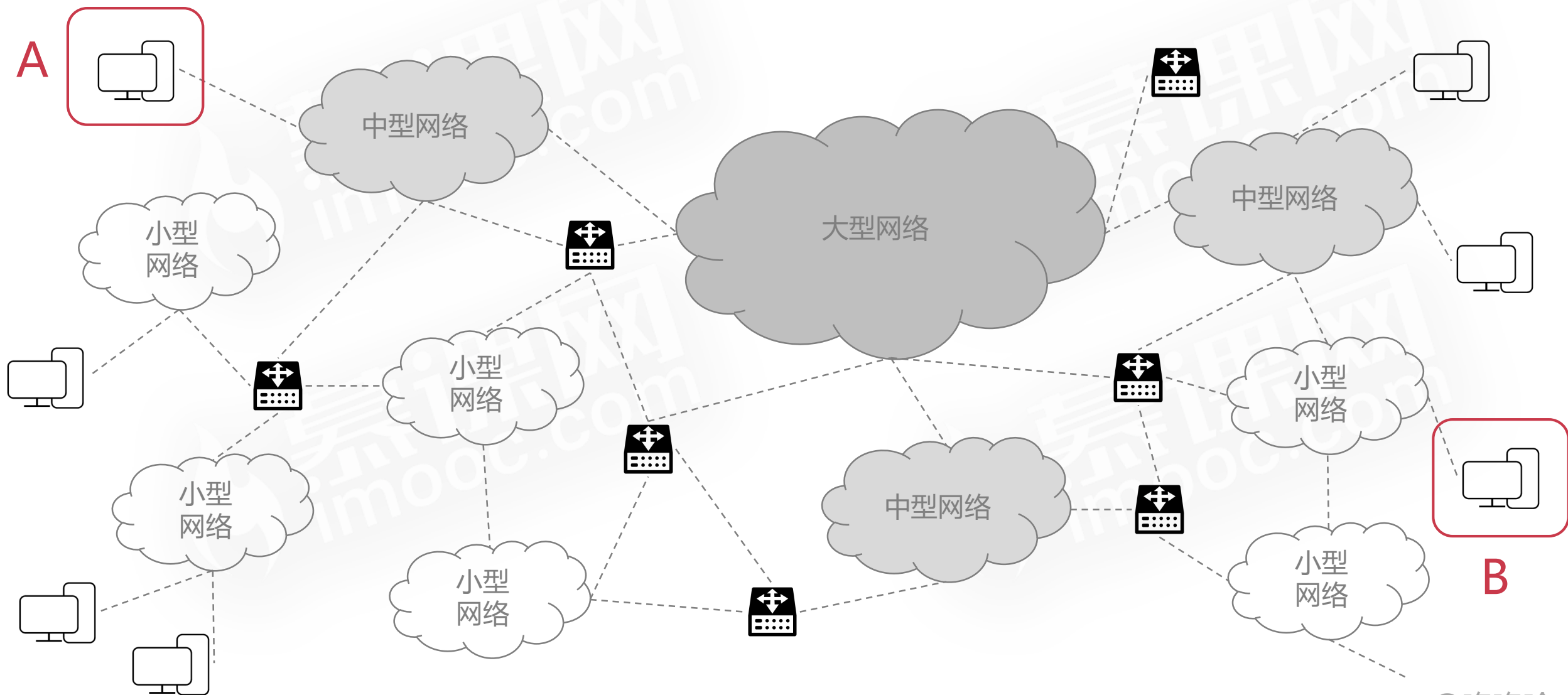


MAC地址	硬件接口
A	E1
D	E2
F	E3
...	...

MAC地址	硬件接口
E	F1
B	F2
C	F3
...	...

A怎么跨设备
传输数据到C?

网络层概述



网络层概述



路由器

章节的主要内容

网络层IP协议相关



网络层其他协议



IP的路由算法

IP协议

子网划分

简单路由过程

ARP协议与RARP协议

ICMP协议

路由的概述

内部网关路由协议

外部网关路由协议

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

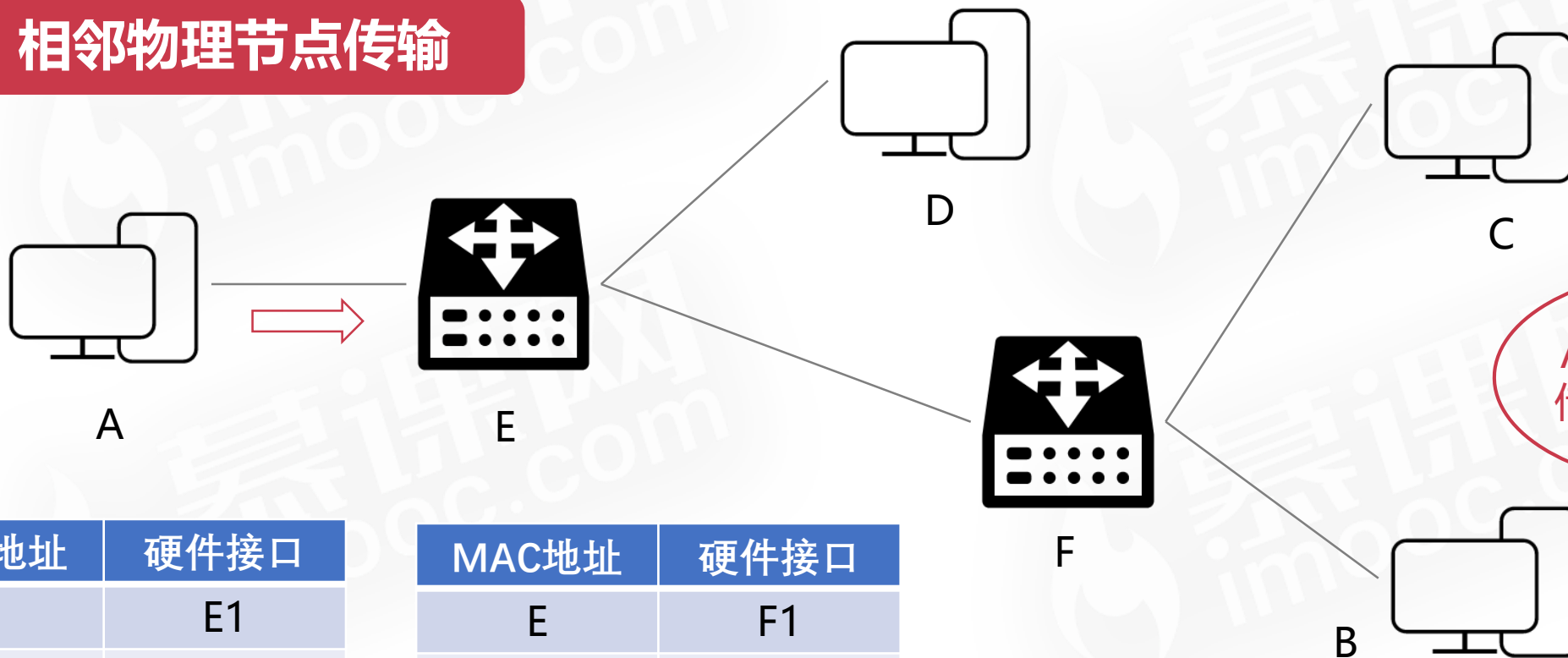
慕课网
imooc.com

IP协议详解

- ◆ 虚拟互连网络
- ◆ IP协议

虚拟互连网络

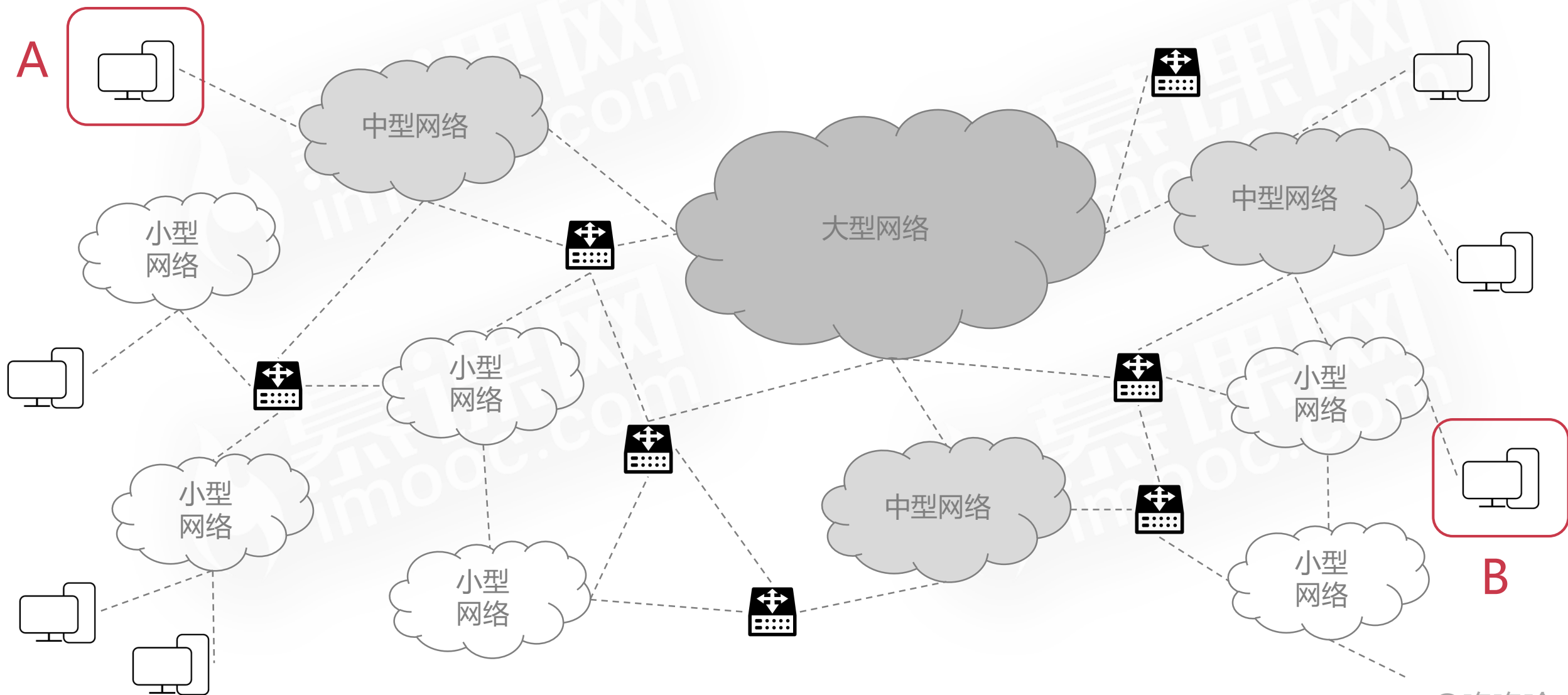
相邻物理节点传输



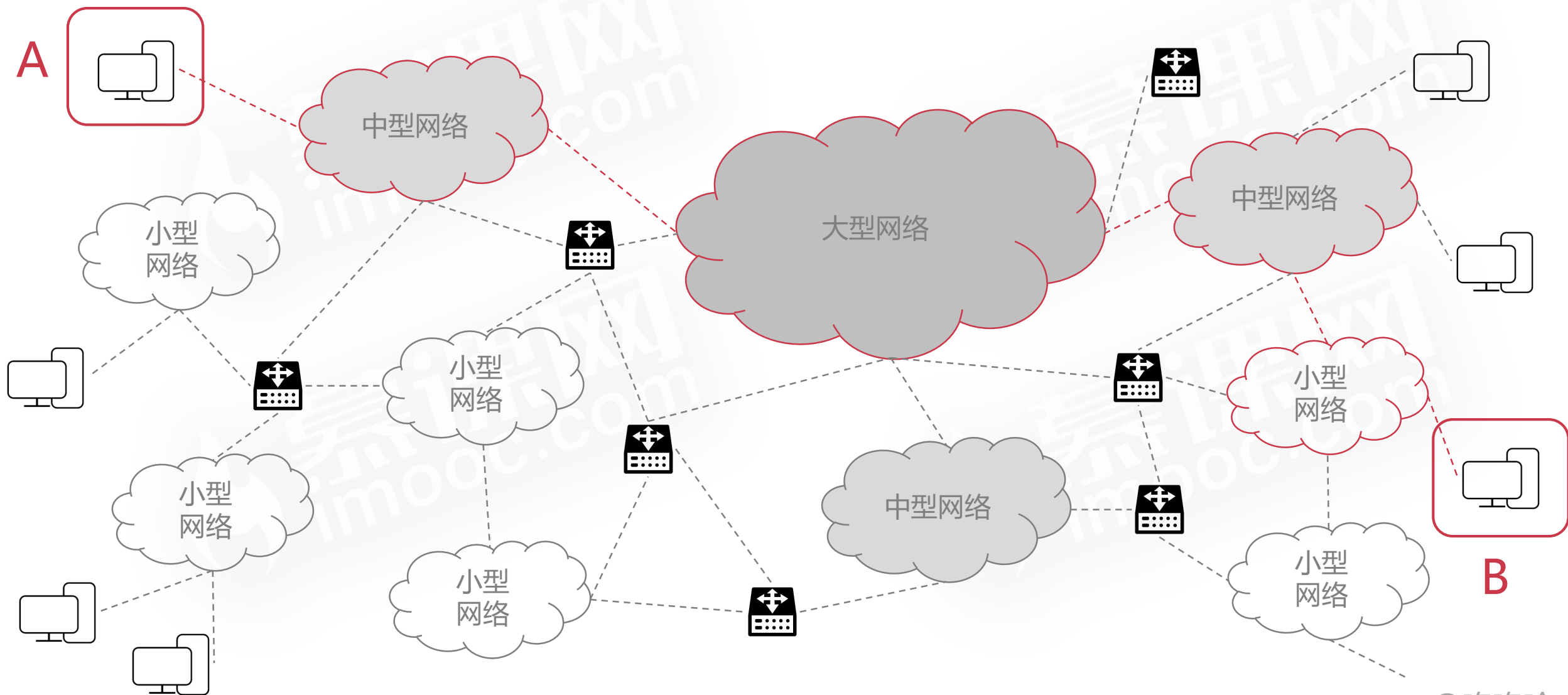
MAC地址	硬件接口
A	E1
D	E2
F	E3
...	...

MAC地址	硬件接口
E	F1
B	F2
C	F3
...	...

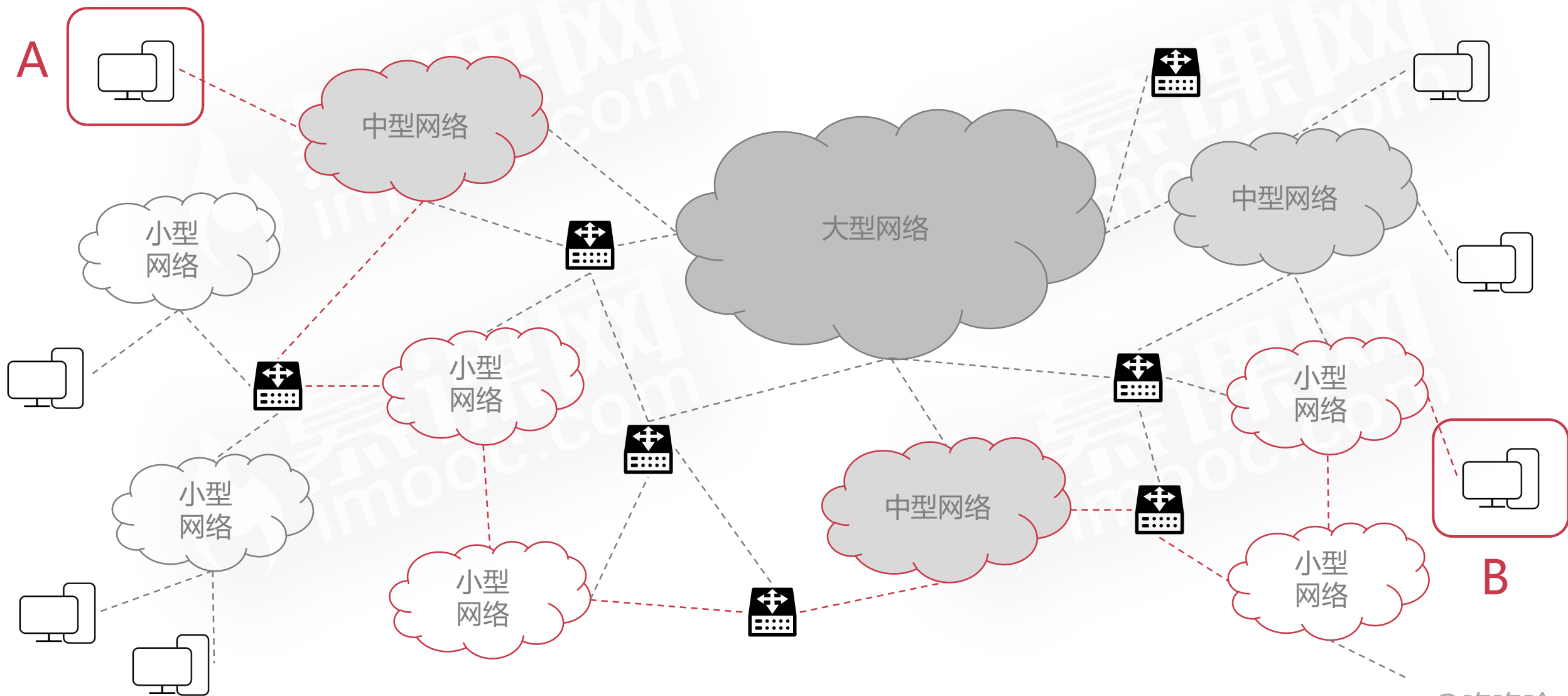
网络层概述



网络层概述



网络层概述



虚拟互连网络

- ◆ 实际的计算机网络是错综复杂的
- ◆ 物理设备通过使用IP协议，屏蔽了物理网络之间的差异
- ◆ 当网络中的主机使用IP协议连接时，则无需关注网络细节

虚拟互连网络

虚拟互连网络



虚拟互连网络

- ◆ IP协议使得复杂的实际网络变为一个虚拟互连的网络
- ◆ IP协议使得网络层可以屏蔽底层细节而专注网络层的数据转发
- ◆ IP协议解决了在虚拟网络中数据报传输路径的问题

IP协议详解

- ◆ 虚拟互连网络
- ◆ IP协议

IP协议

MAC地址: 30-B4-9E-ED-85-CA

00110000-11000020-11101101-01000101-11001010

IP地址: 192.168.11.11

11000000.10101000.00001011.00001011

点分十进制

IP协议

- ◆ IP地址长度为32位，常分成4个8位
- ◆ IP地址常使用点分十进制来表示(0~255.0~255.0~255.0~255)

114.114.114.114

1.1.1.1

$2^{32} = 4294967296$

8.8.8.8

70.12.34.34

255.255.255.255

111.111.111.111

IP协议



001010101001110110101010101010...1010101010101010101010101010101

IP协议

4位版本	4位首部长度	8位服务类型(TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)	8位协议		16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options（若有）				
IP数据				

IP协议

版本：占4位，指的是IP协议的版本，通信双方的版本必须一致，当前主流版本是4，即IPv4，也有IPv6

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议

首部位长度：占4位，最大数值为15，表示的是IP首部长度，单位是“32位字”（4个字节），也即是IP首部最大长度为60字节

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议

总长度： 占16位，最大数值为65535，表示的是IP数据报总长度（IP首部+IP数据）

MTU

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)	8位协议		16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议

标识

标志

片偏移

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议

TTL：占8位，表明IP数据报文在网络中的寿命，每经过一个设备，TTL减1，当TTL=0时，网络设备必须丢弃该报文

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议

协议：占8位，表明IP数据所携带的具体数据是什么协议的
(如：TCP、UDP等)

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议

协议： 占8位，表明IP数据所携带的具体数据是什么协议的（如：TCP、UDP等）

协议名	ICMP	IGMP	IP	TCP	UDP	OSPF	...
字段值	1	2	4	6	17	89	...

IP协议

首部校验和：占16位，校验IP
首部是否有出错

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议

源IP地址

目的IP地址

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

IP协议详解

- ◆ 虚拟互连网络
- ◆ IP协议

慕课网
imooc.com

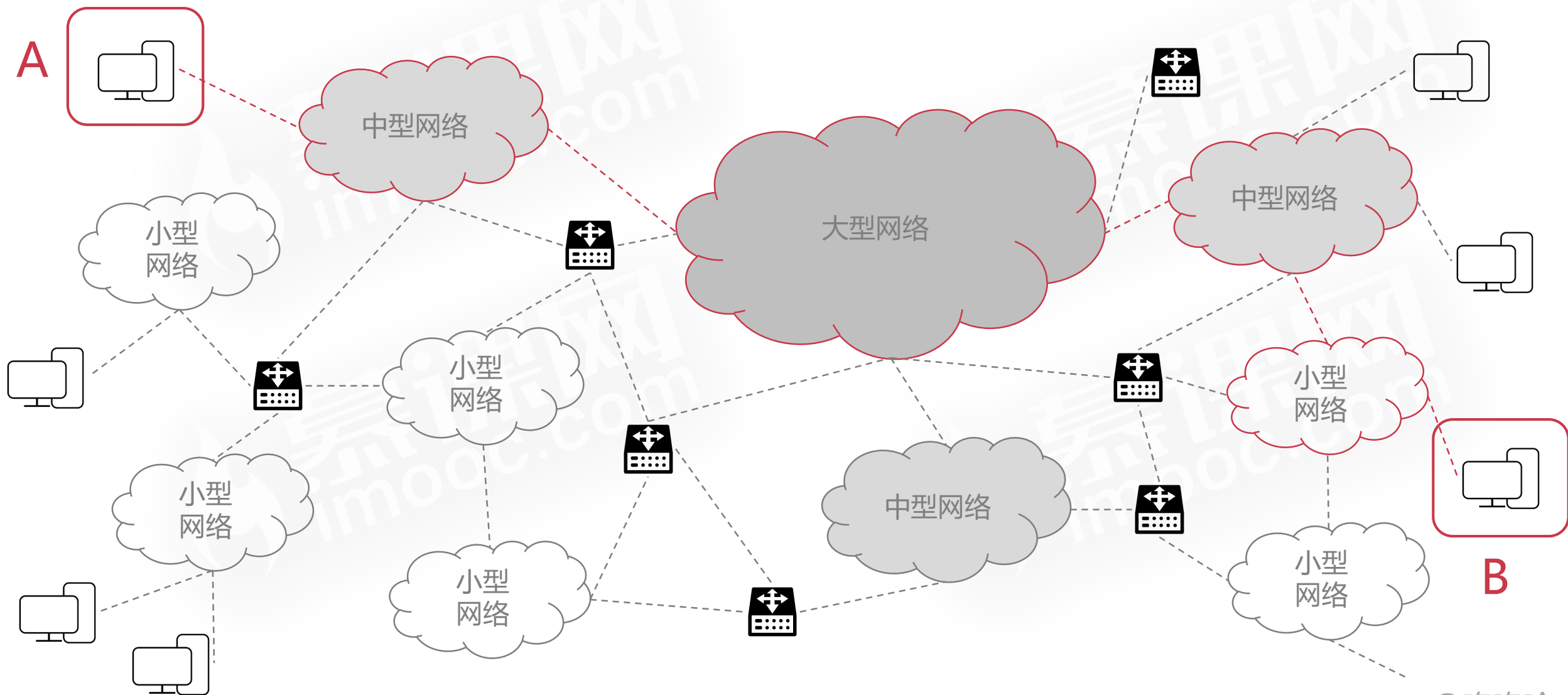
慕课网
imooc.com

慕课网
imooc.com

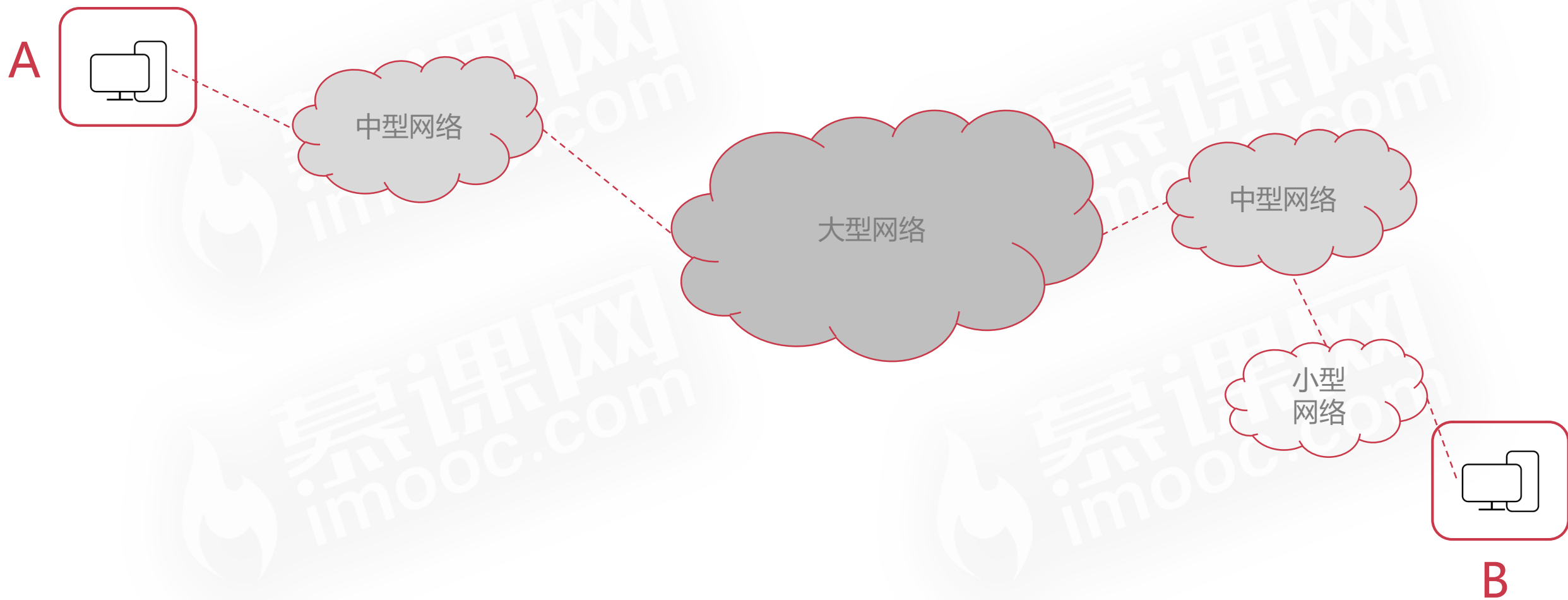
慕课网
imooc.com

IP协议的转发流程

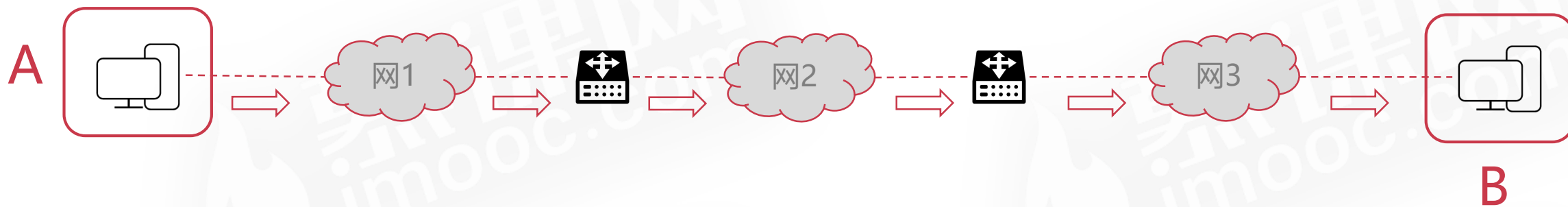
IP协议的转发流程



IP协议的转发流程



IP协议的转发流程

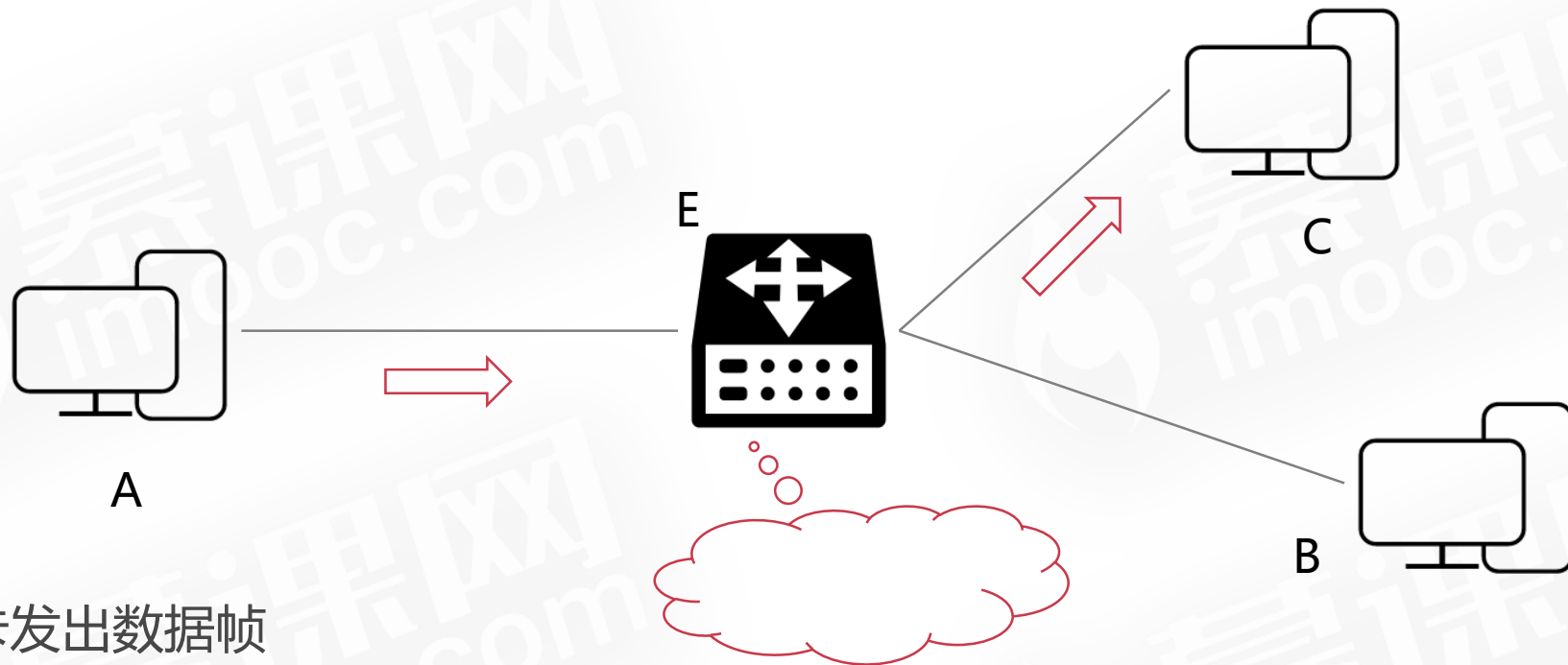


逐跳(hop-by-hop)

IP协议的转发流程

- ◆ 路由表简介
- ◆ IP协议的转发流程

路由表简介



- ◆ A通过网卡发出数据帧
- ◆ 数据帧到达路由器，路由器取出前6字节
- ◆ 路由器匹配MAC地址表，找到对应的网络接口
- ◆ 路由器往该网络接口发送数据帧

MAC地址	硬件接口
A	E1
B	E2
C	E3
...	...

路由表简介

MAC地址	硬件接口
A	E1
B	E2
C	E3
...	...

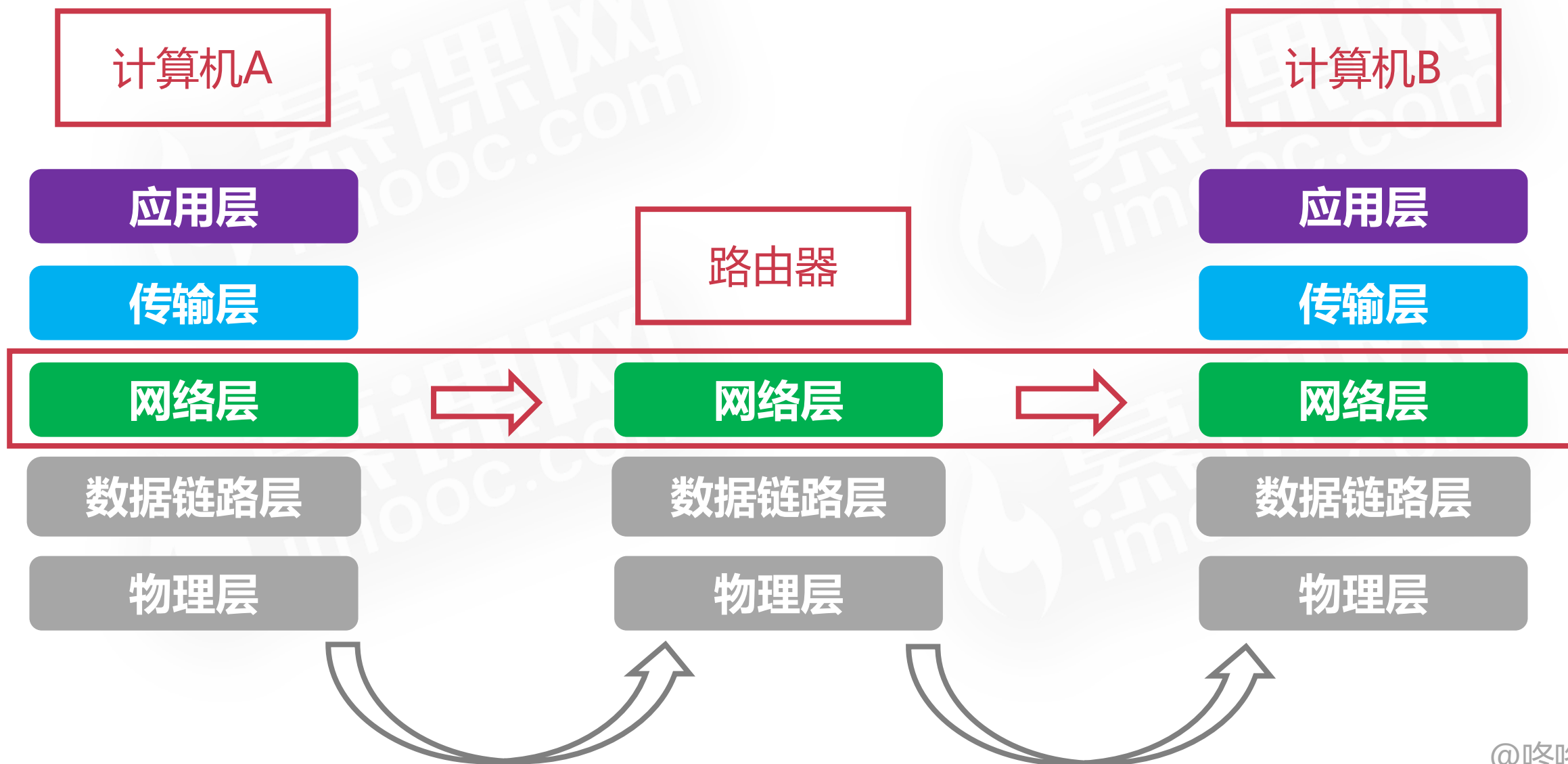
目的IP地址	下一跳IP地址
IP1	IP4
IP2	IP5
IP3	IP6
...	...

计算机或者路由器都拥有路由表

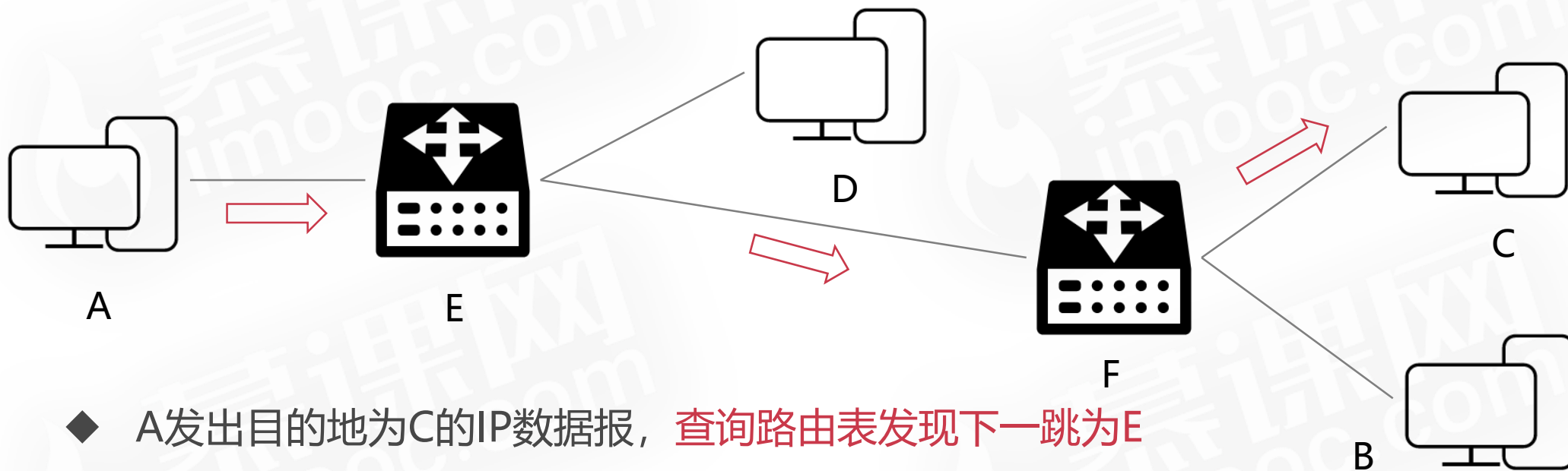
IP协议的转发流程

- ◆ 路由表简介
- ◆ IP协议的转发流程

IP协议的转发流程

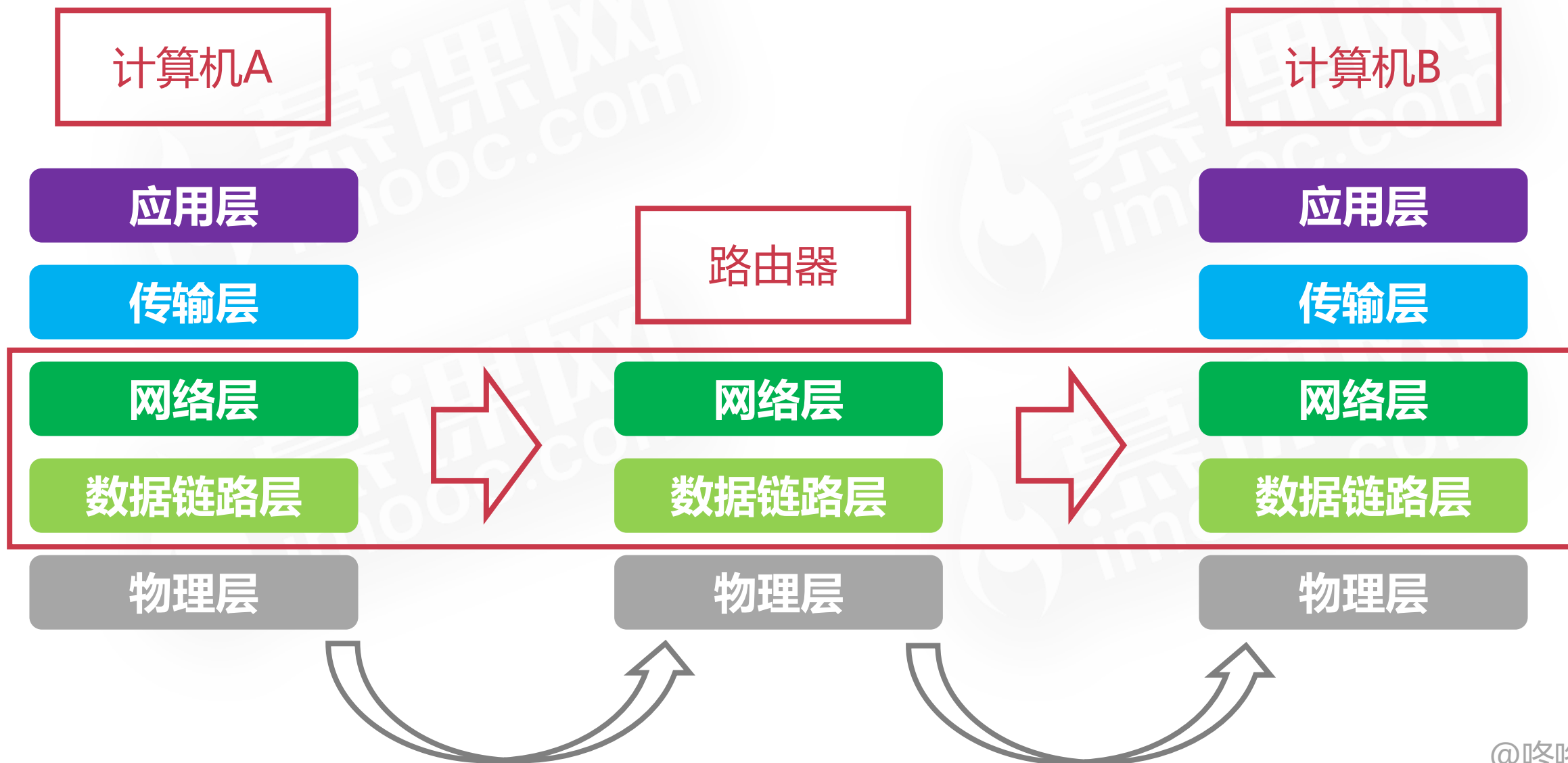


IP协议的转发流程

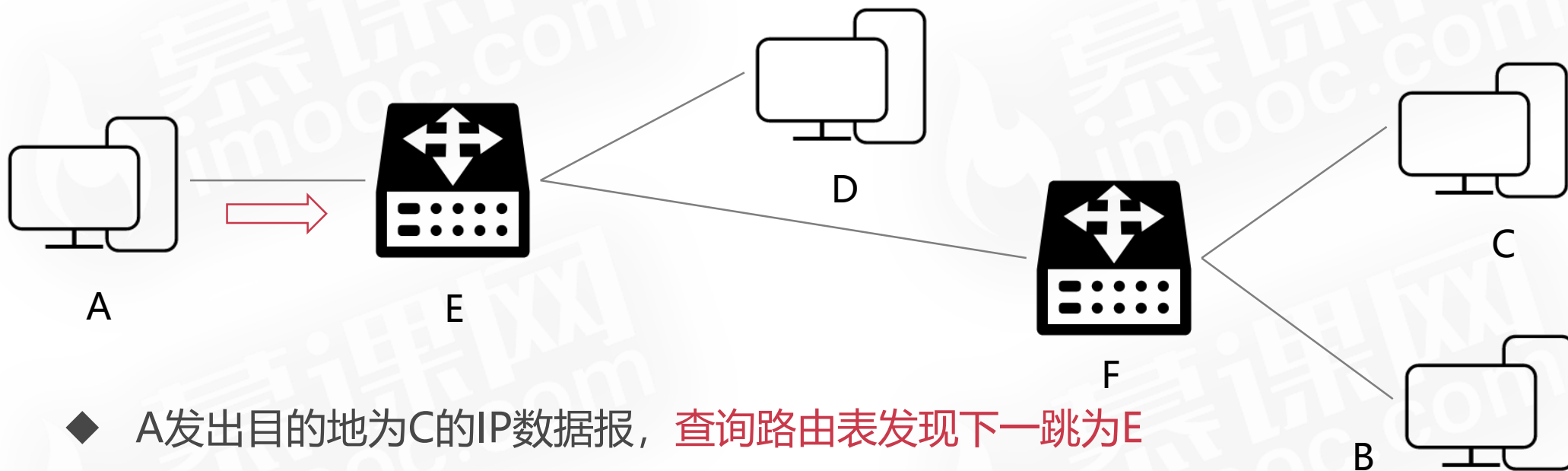


- ◆ A发出目的地为C的IP数据报，**查询路由表发现下一跳为E**
- ◆ A将数据报发送给E
- ◆ E**查询路由表发现下一跳为F**，将数据报发送给F
- ◆ F**查询路由表发现目的地C直接连接**，将数据报发送给C

IP协议的转发流程



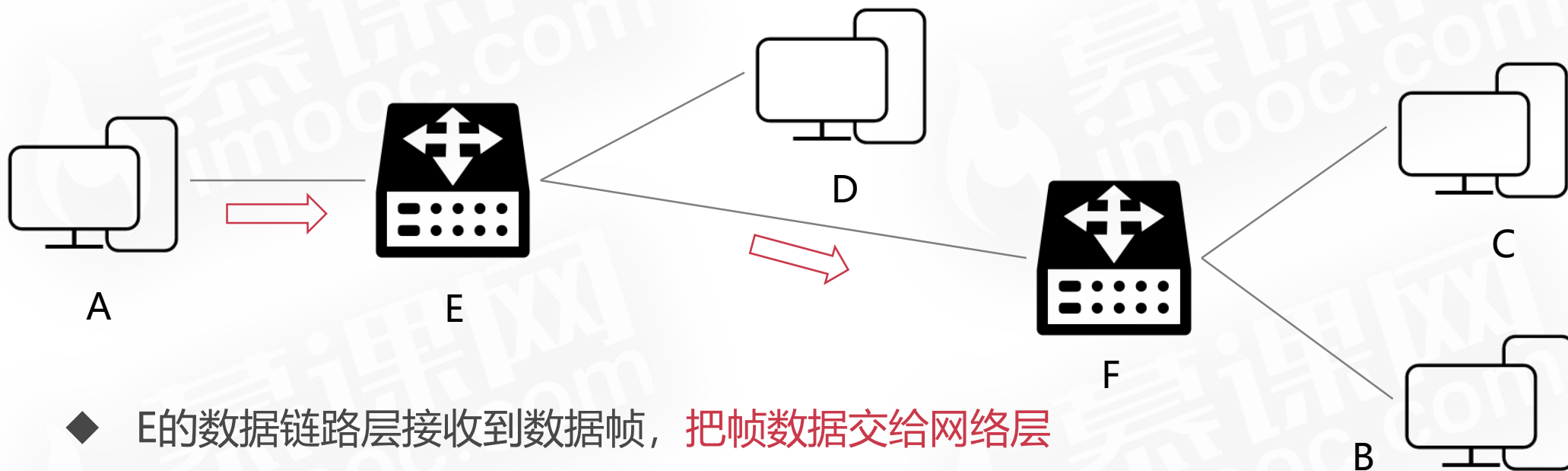
IP协议的转发流程



- ◆ A发出目的地为C的IP数据报，**查询路由表发现下一跳为E**
- ◆ A将IP数据报**交给数据链路层**，并告知目的MAC地址是E
- ◆ 数据链路层**填充源MAC地址A和目的MAC地址E**
- ◆ 数据链路层通过物理层将数据发送给E

路由表

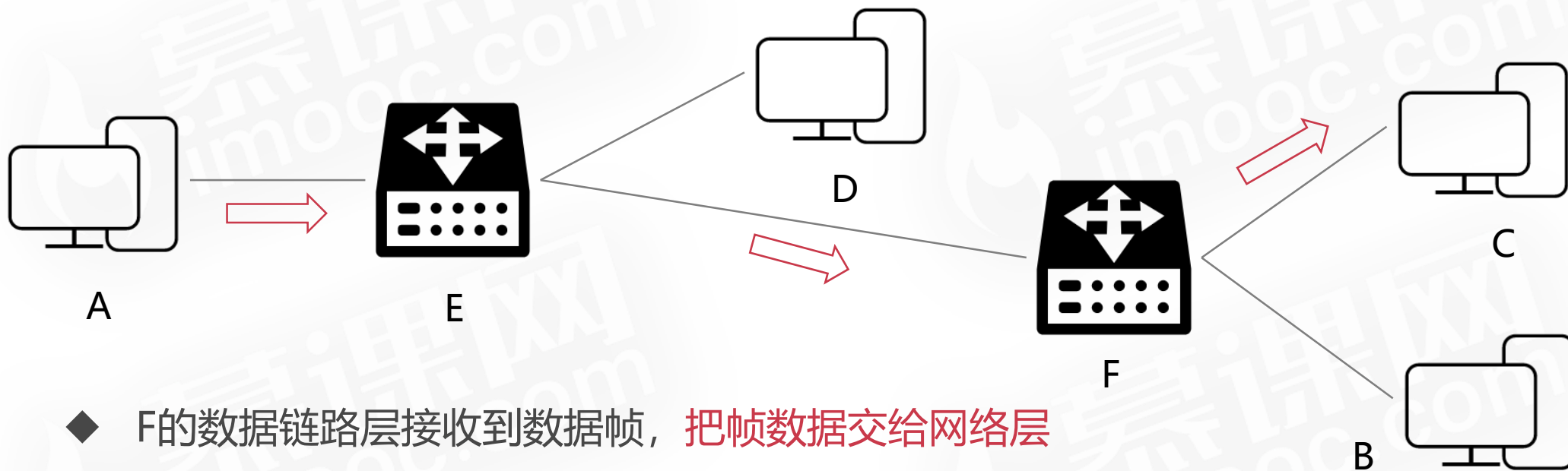
IP协议的转发流程



- ◆ E的数据链路层接收到数据帧，把帧数据交给网络层
- ◆ E查询路由表，发现下一跳为F
- ◆ E把数据报交给数据链路层，并告知目的MAC地址为F
- ◆ E的数据链路层封装数据帧并发送

路由表

IP协议的转发流程



- ◆ F的数据链路层接收到数据帧，把帧数据交给网络层
- ◆ F查询路由表，发现下一跳为C
- ◆ F把数据报交给数据链路层，并告知目的MAC地址为C
- ◆ F的数据链路层封装数据帧并发送

路由表

IP协议的转发流程

- ◆ 数据帧**每一跳**的MAC地址都在变化
- ◆ IP数据报**每一跳**的IP地址始终不变

IP协议的转发流程

- ◆ 路由表简介
- ◆ IP协议的转发流程

IP协议的转发流程

慕课网
imooc.com

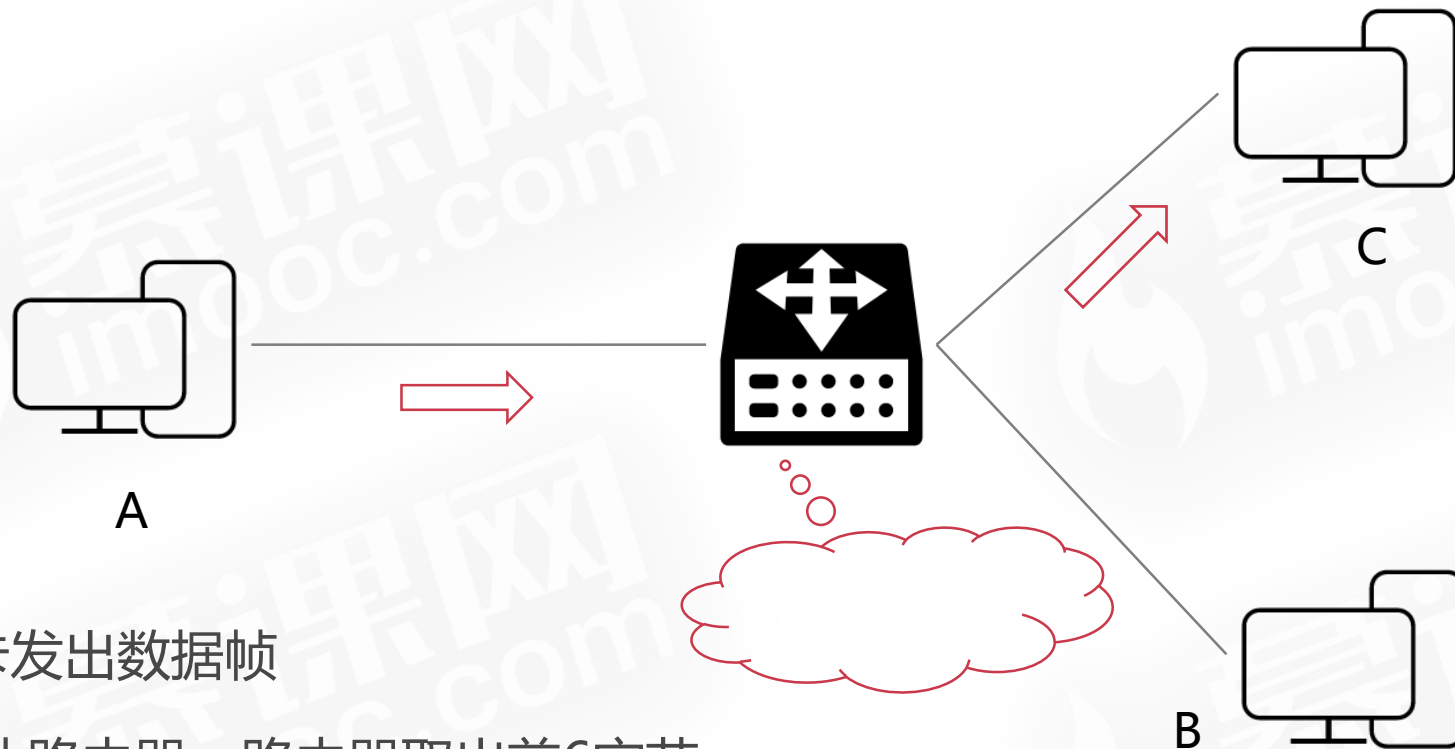
慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

ARP协议与RARP协议

ARP协议与RARP协议

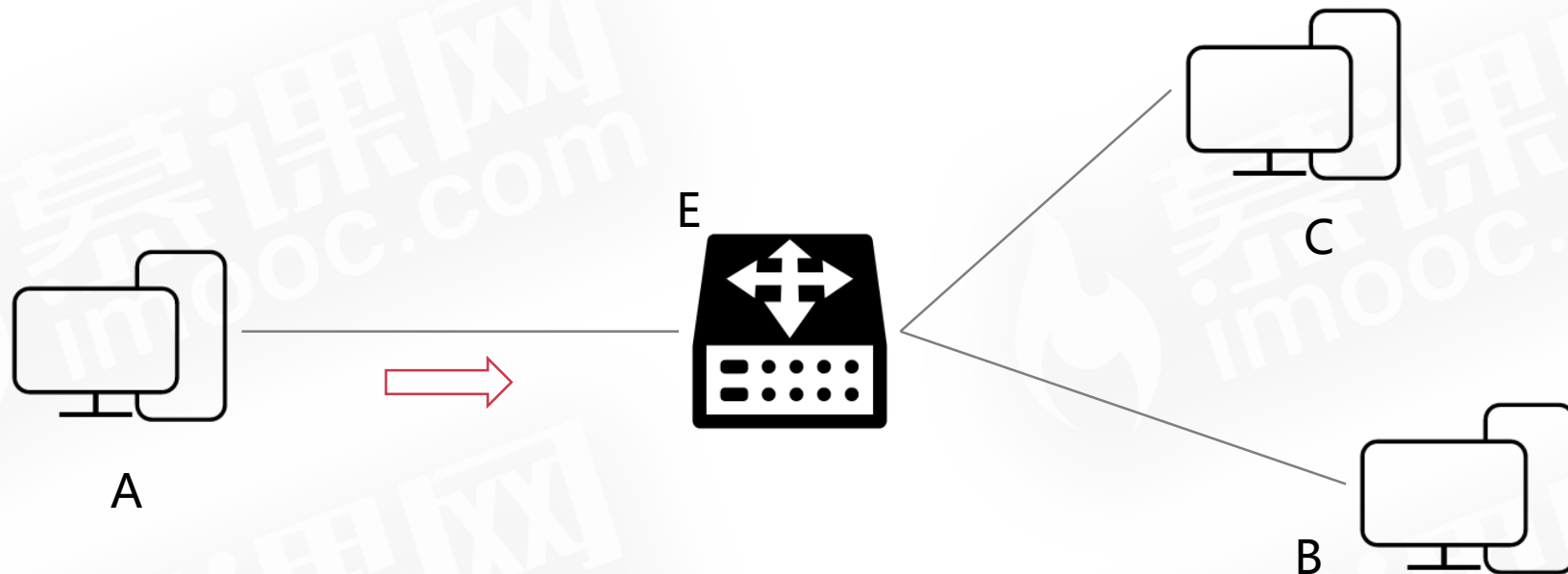


- ◆ A通过网卡发出数据帧
- ◆ 数据帧到达路由器，路由器取出前6字节
- ◆ 路由器匹配MAC地址表，找到对应的网络接口
- ◆ 路由器往该网络接口发送数据帧

路由器不知道
C的MAC地址?



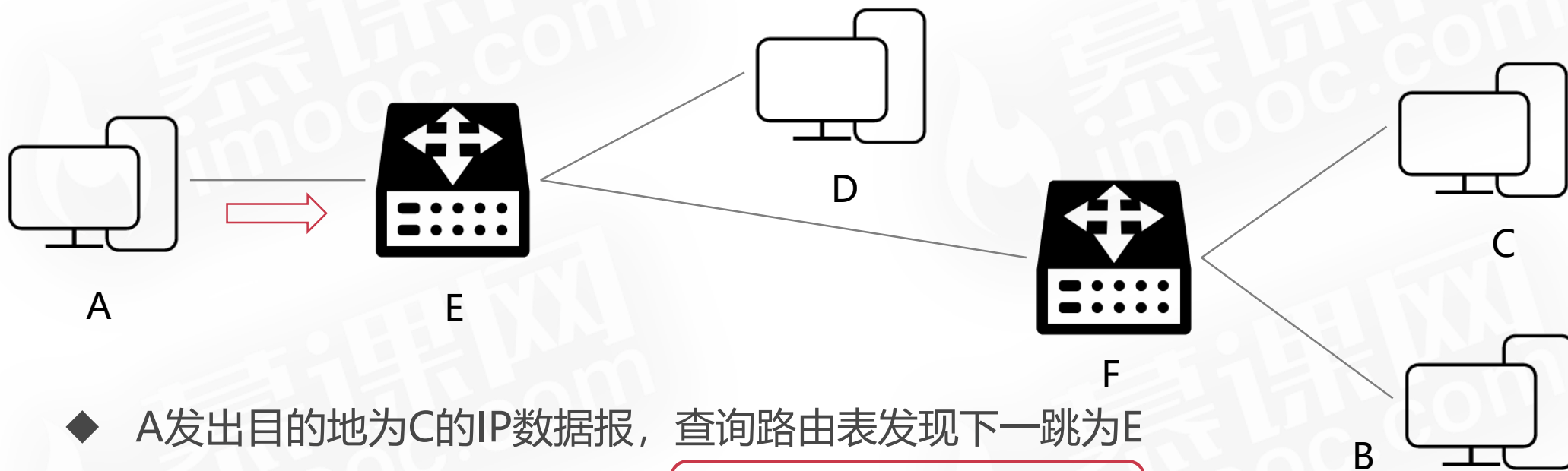
ARP协议与RARP协议



- ◆ E检查MAC地址表，发现没有C的信息
- ◆ E将广播A的数据包到除A以外的端口
- ◆ E将收到来自B、C的回应，并将地址记录

MAC地址	硬件接口
A	E1
B	E2
C	未知
...	...

ARP协议与RARP协议

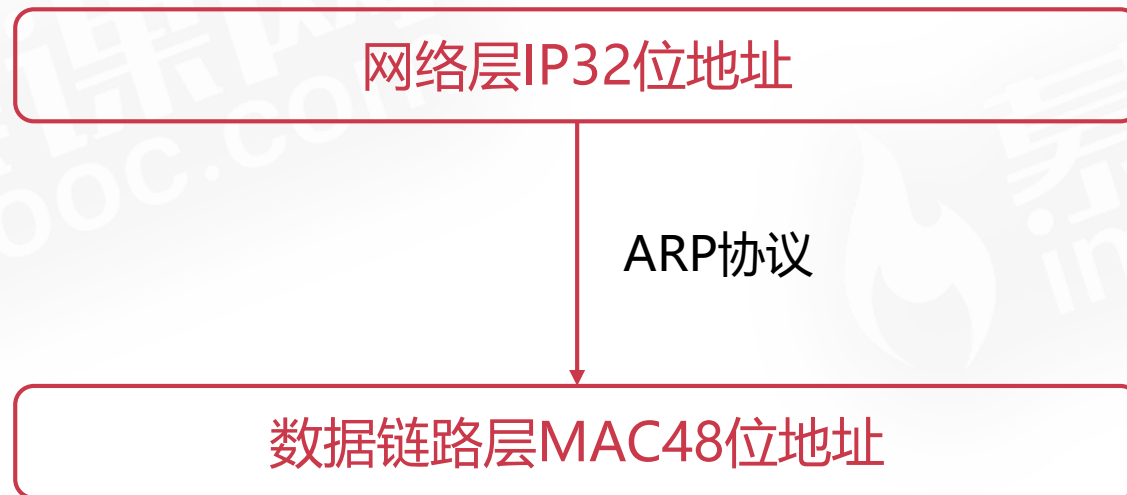


- ◆ A发出目的地为C的IP数据报，查询路由表发现下一跳为E
- ◆ A将IP数据报交给数据链路层，并告知目的MAC地址是E
- ◆ 数据链路层填充源MAC地址A和目的MAC地址E
- ◆ 数据链路层通过物理层将数据发送给E

路由表

ARP协议与RARP协议

◆ ARP(Address Resolution Protocol)地址解析协议



ARP协议与RARP协议

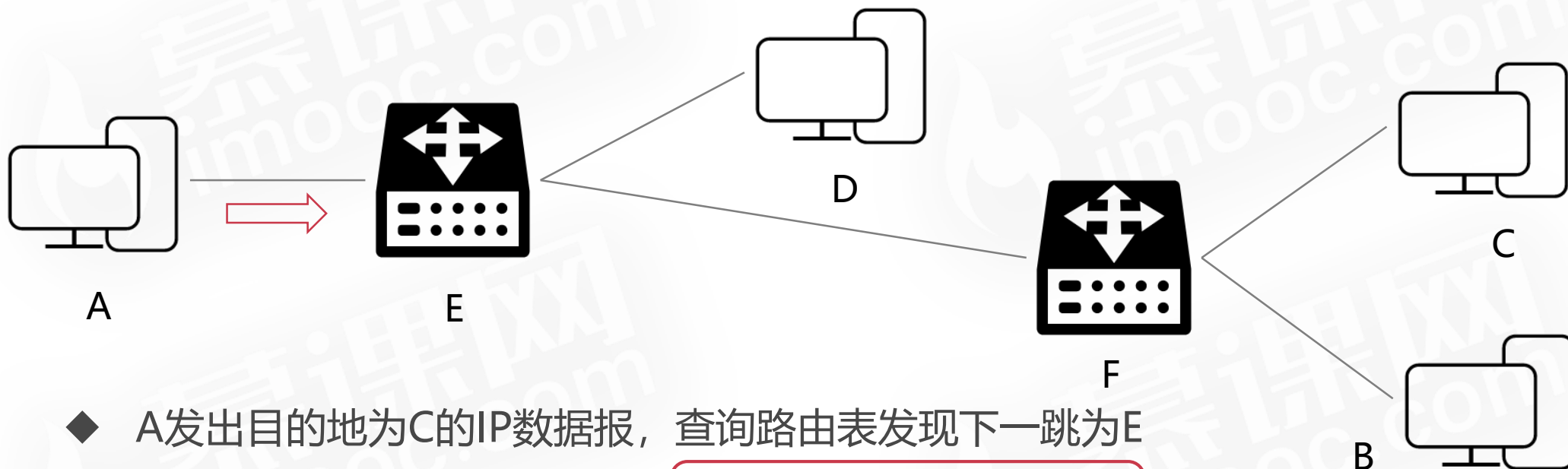
IP地址	MAC地址
192.168.83.254	00-50-56-e0-33-40
192.168.83.255	01-00-5e-00-00-16
224.0.0.251	01-00-5e-00-00-fc
239.1.2.3	01-00-5e-40-98-8f
255.255.255.255	01-00-5e-7f-ff-fa

ARP缓存表

ARP协议与RARP协议

- ◆ ARP缓存表缓存有IP地址和MAC地址的映射关系
- ◆ ARP缓存表没有缓存IP地址和MAC地址的映射关系

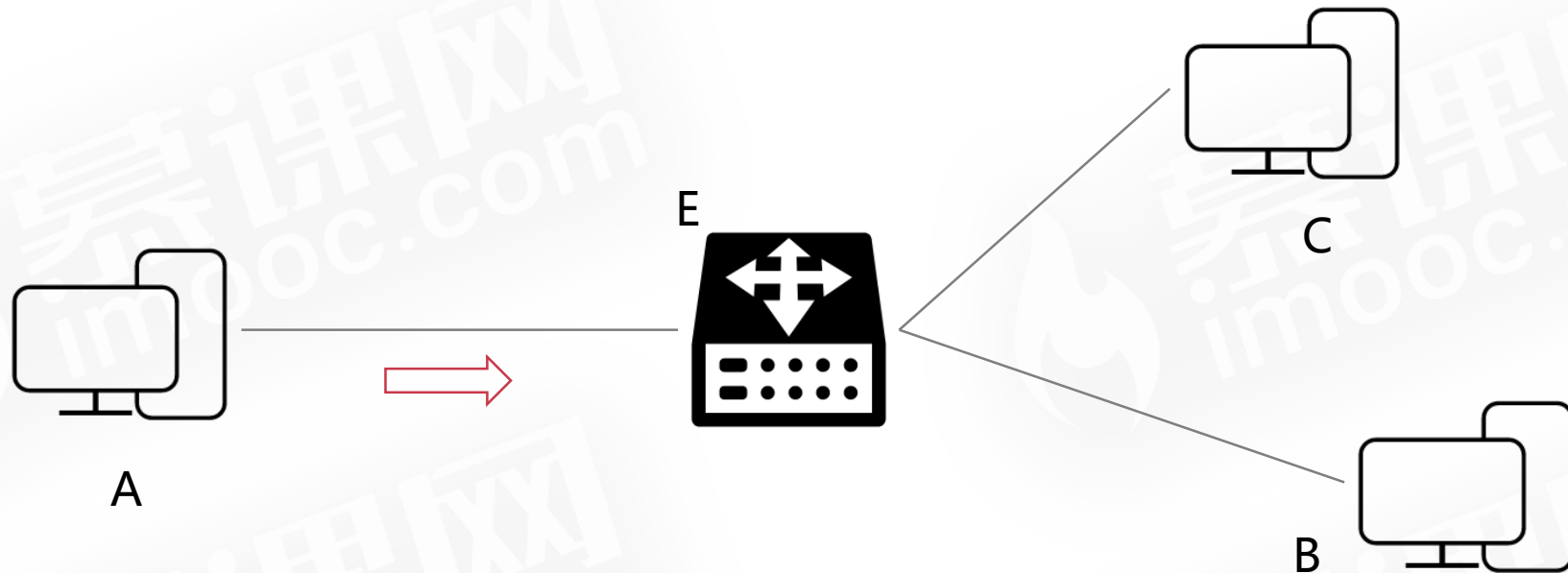
ARP协议与RARP协议



- ◆ A发出目的地为C的IP数据报，查询路由表发现下一跳为E
- ◆ A将IP数据报交给数据链路层，并告知目的MAC地址是E
- ◆ 数据链路层填充源MAC地址A和目的MAC地址E
- ◆ 数据链路层通过物理层将数据发送给E

路由表

ARP协议与RARP协议



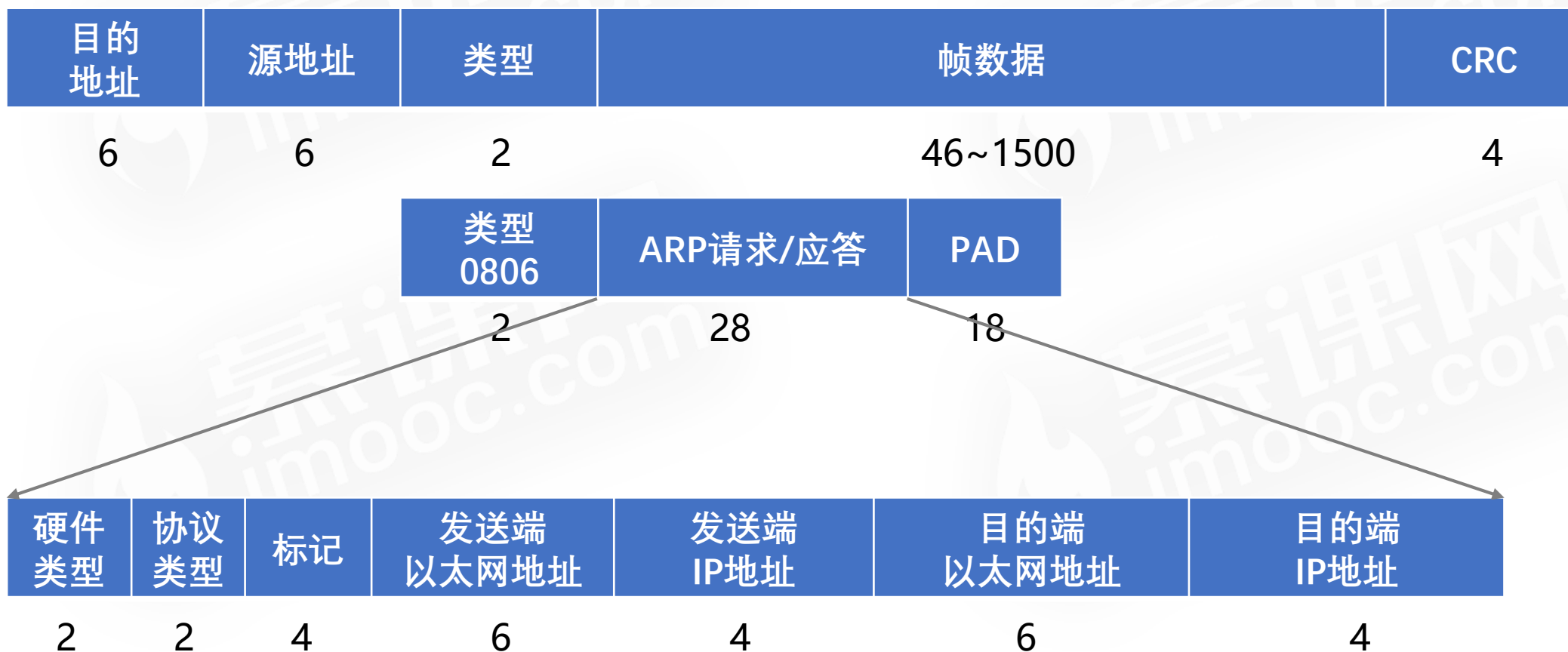
- ◆ E检查MAC地址表，发现没有C的信息
- ◆ E将广播A的数据包到除A以外的端口
- ◆ E将收到来自B、C的回应，并将地址记录

ARP协议与RARP协议

- ◆ ARP缓存表是ARP协议和RARP协议运行的关键
- ◆ ARP缓存表缓存了IP地址到硬件地址之间的映射关系
- ◆ ARP缓存表中的记录并不是永久有效的，有一定的期限

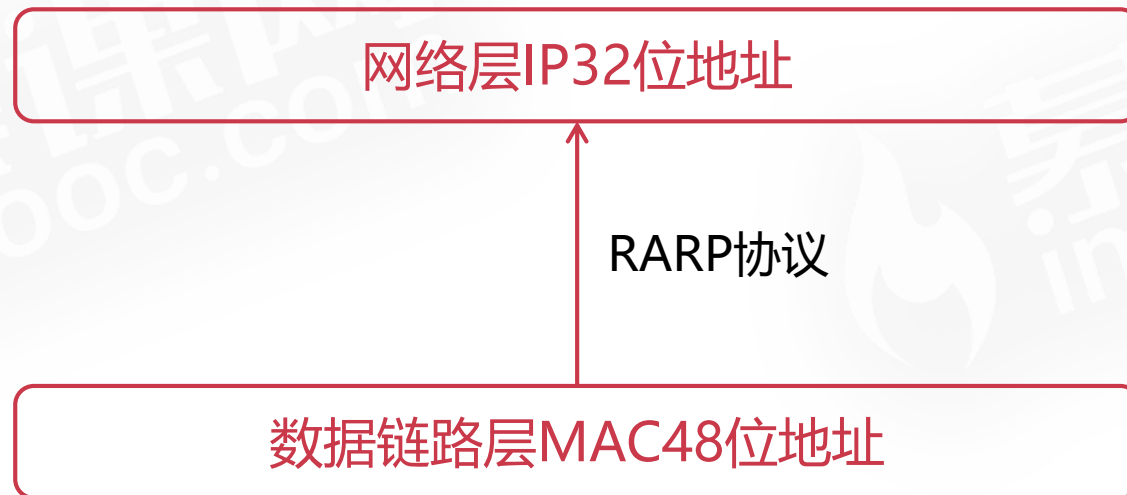
[查看ARP缓存表→](#)

ARP协议与RARP协议



ARP协议与RARP协议

- ◆ RARP(Reverse Address Resolution Protocol)逆地址解析协议

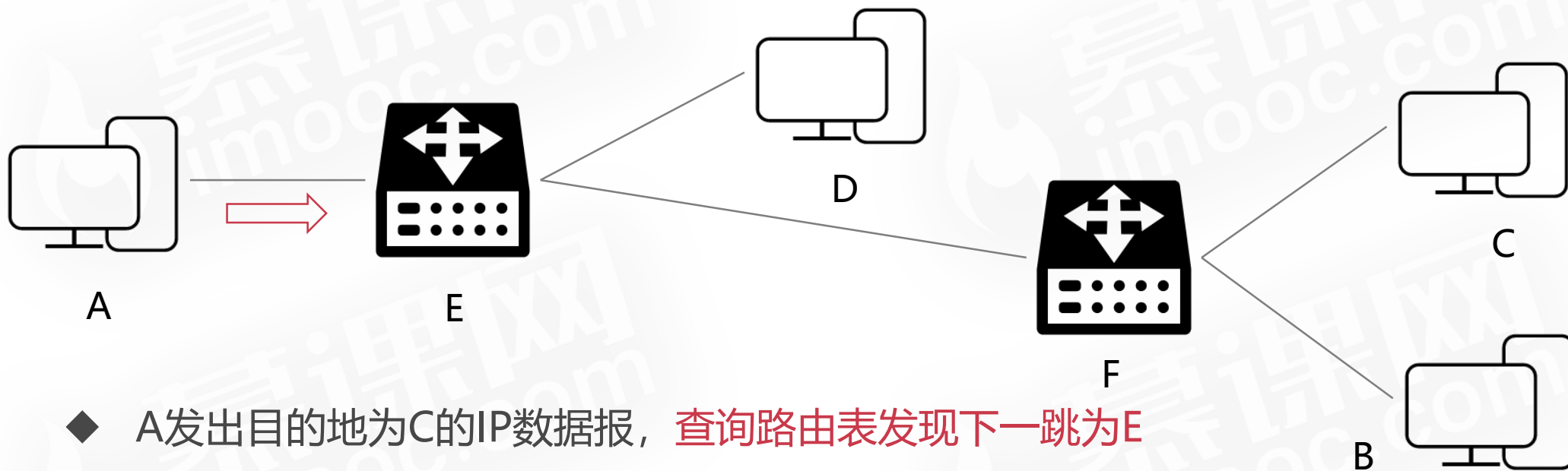


ARP协议与RARP协议

目的地址	源地址	类型	帧数据	CRC
6	6	2	46~1500	4

类型 0806	ARP请求/应答	PAD
2	28	18
类型 8035	RARP请求/应答	PAD
2	28	18

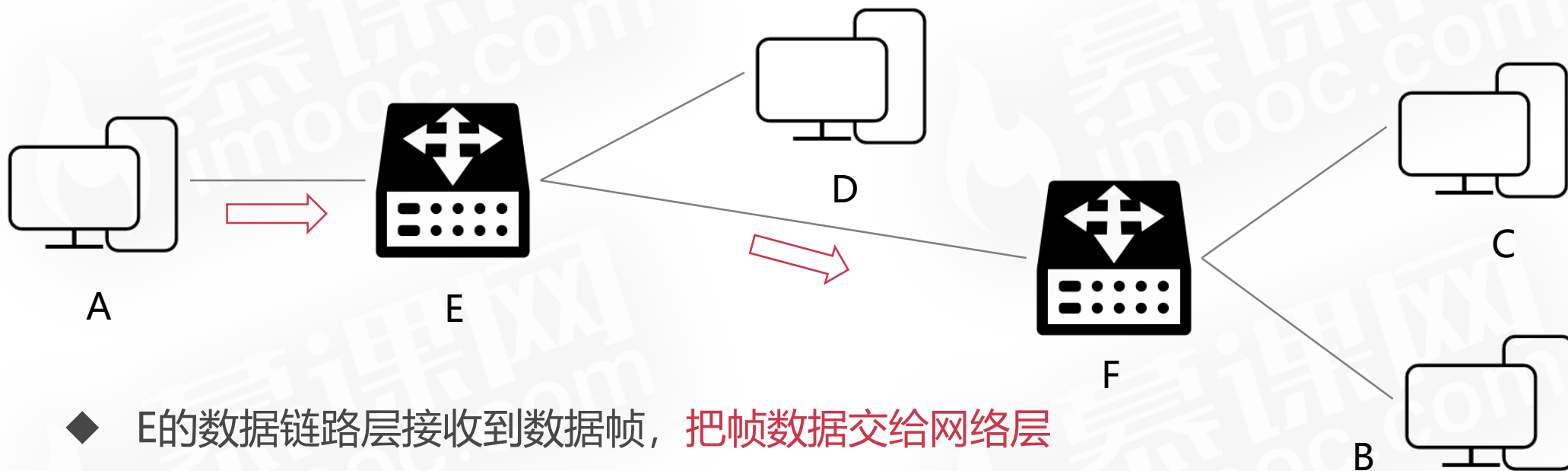
ARP协议与RARP协议



- ◆ A发出目的地为C的IP数据报，查询路由表发现下一跳为E
- ◆ A将IP数据报交给数据链路层，并告知目的MAC地址是E
- ◆ 数据链路层填充源MAC地址A和目的MAC地址E
- ◆ 数据链路层通过物理层将数据发送给E

路由表

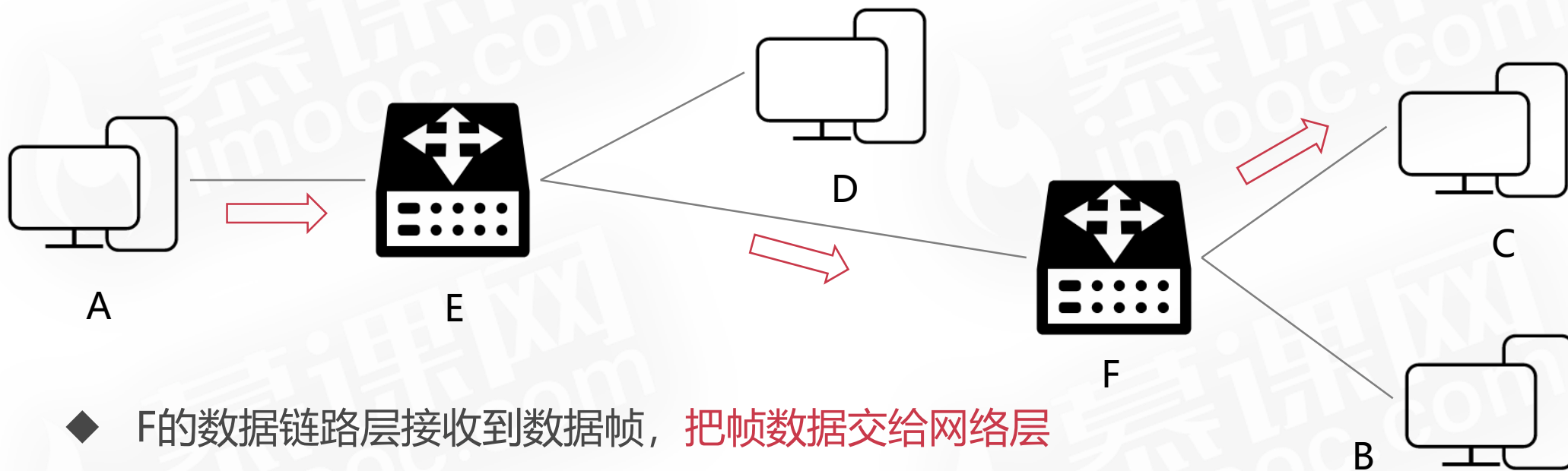
ARP协议与RARP协议



- ◆ E的数据链路层接收到数据帧，把帧数据交给网络层
- ◆ E查询路由表，发现下一跳为F
- ◆ E把数据报交给数据链路层，并告知目的MAC地址为F
- ◆ E的数据链路层封装数据帧并发送

路由表

ARP协议与RARP协议



- ◆ F的数据链路层接收到数据帧，把帧数据交给网络层
- ◆ F查询路由表，发现下一跳为C
- ◆ F把数据报交给数据链路层，并告知目的MAC地址为C
- ◆ F的数据链路层封装数据帧并发送

路由表

ARP协议与RARP协议

- ◆ (R)ARP协议是TCP/IP协议栈里面基础的协议
- ◆ ARP和RARP的操作对程序员是透明的
- ◆ 理解(R)ARP协议有助于理解网络分层的细节



IP地址的子网划分

IP地址的子网划分

- ◆ IP地址长度为32位，常分成4个8位
- ◆ IP地址常使用点分十进制来表示(0~255.0~255.0~255.0~255)

114.114.114.114

1.1.1.1

$2^{32} = 4294967296$

8.8.8.8

70.12.34.34

255.255.255.255

111.111.111.111

规划和分配IP地址非常麻烦

IP地址的子网划分

- ◆ 分类的IP地址
- ◆ 划分子网
- ◆ 无分类编址CIDR

分类的IP地址



分类的IP地址

	最小网络号	最大网络号	子网数量	最小主机号	最大主机号	主机数量
A	0(00000000)	127 (01111111)	2^7	0.0.0	255.255.255	2^{24}
B	128.0	191.255	2^{14}	0.0	255.255	2^{16}
C	192.0.0	223.255.255	2^{21}	0	255	2^8

分类的IP地址

特殊的主机号

- ◆ 主机号全0表示当前网络段，不可分配为特定主机
- ◆ 主机号为全1表示广播地址，向当前网络段所有主机发消息

1.2.3.4

1.0.0.0

1.255.255.255

分类的IP地址

特殊的网络号

- ◆ A类地址网络段全0(00000000)表示特殊网络
- ◆ A类地址网络段后7位全1(01111111:127)表示回环地址
- ◆ B类地址网络段(10000000.00000000:128.0)是不可使用的
- ◆ C类地址网络段(192.0.0)是不可使用的

分类的IP地址

	最小网络号	最大网络号	子网数量	最小主机号	最大主机号	主机数量
A	1	127 (01111111)	2^7-2	0.0.0	255.255.255	$2^{24}-2$
B	128.1	191.255	$2^{14}-1$	0.0	255.255	$2^{16}-2$
C	192.0.1	223.255.255	$2^{21}-1$	0	255	2^8-2

125.125.3.60

01111101

163.70.31.23

10100011

210.36.127.11

11010010

分类的IP地址

127.0.0.1, 通常被称为本地回环地址(Loopback Address), 不属于任何一个有类别地址类。它代表设备的本地虚拟接口, 所以默认被看作是永远不会宕掉的接口。在Windows操作系统中也有相似的定义, 所以通常在安装网卡前就可以ping通这个本地回环地址。一般都会用来检查本地网络协议、基本数据接口等是否正常的。

分类的IP地址

	最小网络号	最大网络号	子网数量	最小主机号	最大主机号	主机数量
A	1	127 (01111111)	2^7-2	0.0.0	255.255.255	$2^{24}-2$
B	128.1	191.255	$2^{14}-1$	0.0	255.255	$2^{16}-2$
C	192.0.1	223.255.255	$2^{21}-1$	0	255	2^8-2

D类地址:

1110.....

E类地址:

1111.....

IP地址的子网划分

- ◆ 分类的IP地址
- ◆ 划分子网

划分子网

某公司拥有100名员工，每人配备一个计算机，请问该公司应该申请哪种网络段？

	最小网络号	最大网络号	子网数量	最小主机号	最大主机号	主机数量
A	1	127 (01111111)	2^7-2	0.0.1	255.255.254	$2^{24}-2$
B	128.1	191.255	$2^{14}-1$	0.1	255.254	$2^{16}-2$
C	192.0.1	223.255.255	$2^{21}-1$	1	254	2^8-2

193.10.10.0

划分子网

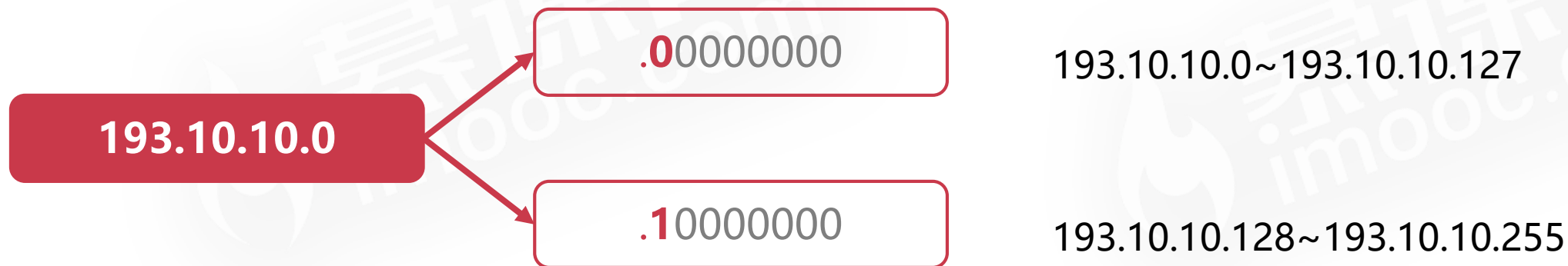
某公司拥有256名员工，每人配备一个计算机，请问该公司应该申请哪种网络段？

	最小网络号	最大网络号	子网数量	最小主机号	最大主机号	主机数量
A	1	127 (01111111)	2^7-2	0.0.1	255.255.254	$2^{24}-2$
B	128.1	191.255	$2^{14}-1$	0.1	255.254	$2^{16}-2$
C	192.0.1	223.255.255	$2^{21}-1$	1	254	2^8-2

190.17.0.0

造成了很大的地址空间浪费

划分子网



划分子网

某公司拥有100名员工，每人配备一个计算机，请问该公司应该申请哪种网络段？

193.10.10.128

划分子网



193.10.10.0

.00000000

193.10.10.0~193.10.10.127

.10000000

193.10.10.128~193.10.10.255

子网掩码

子网号这么多，
有没有办法快速
判断某个IP的网络
号？



@咚咚哈

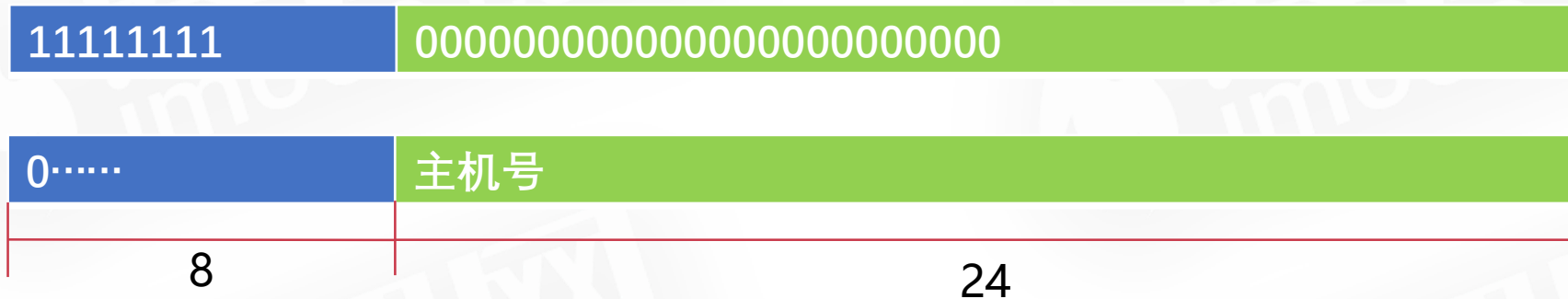
划分子网

子网掩码

- ◆ 子网掩码和IP地址一样，都是32位
- ◆ 子网掩码由连续的1和连续的0组成
- ◆ 某一个子网的子网掩码具备网络号位数个连续的1

划分子网

A类地址:



B类地址:



255.0.0.0

255.255.0.0

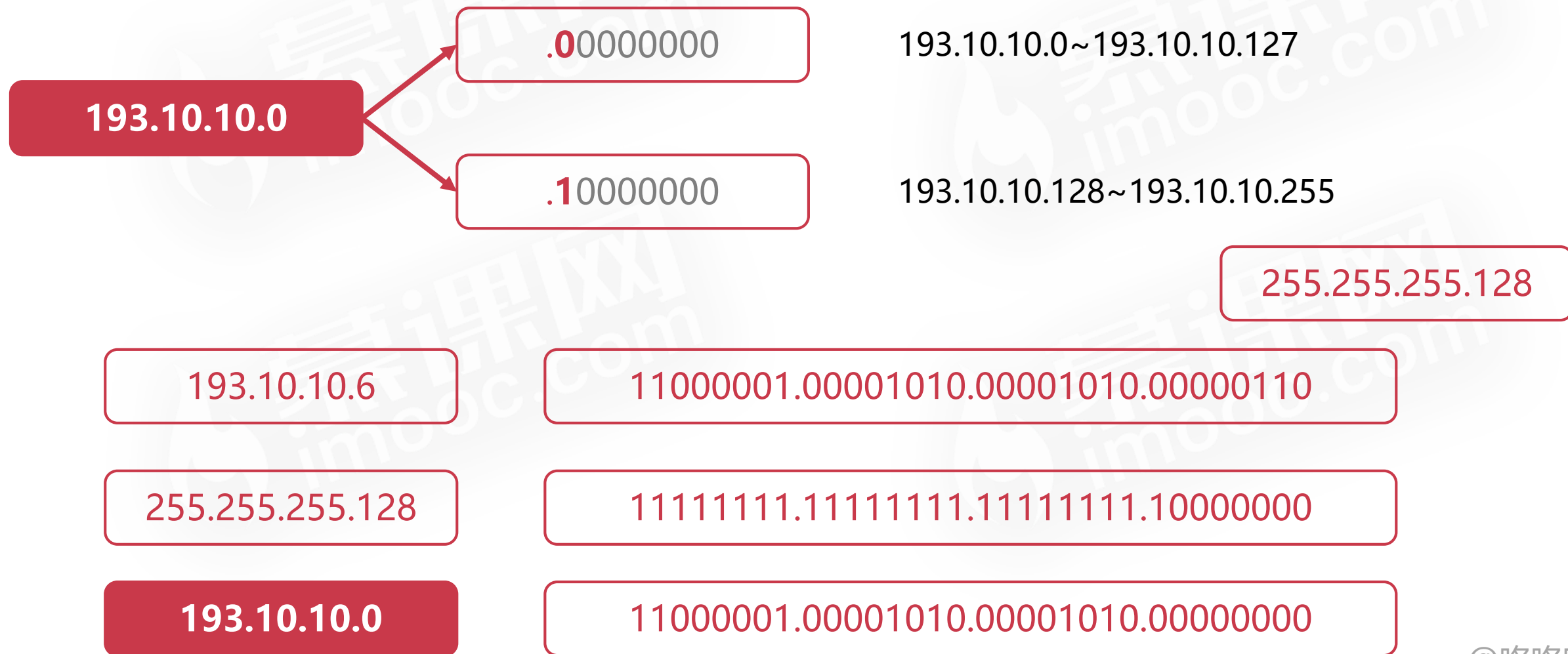
255.255.255.0

划分子网

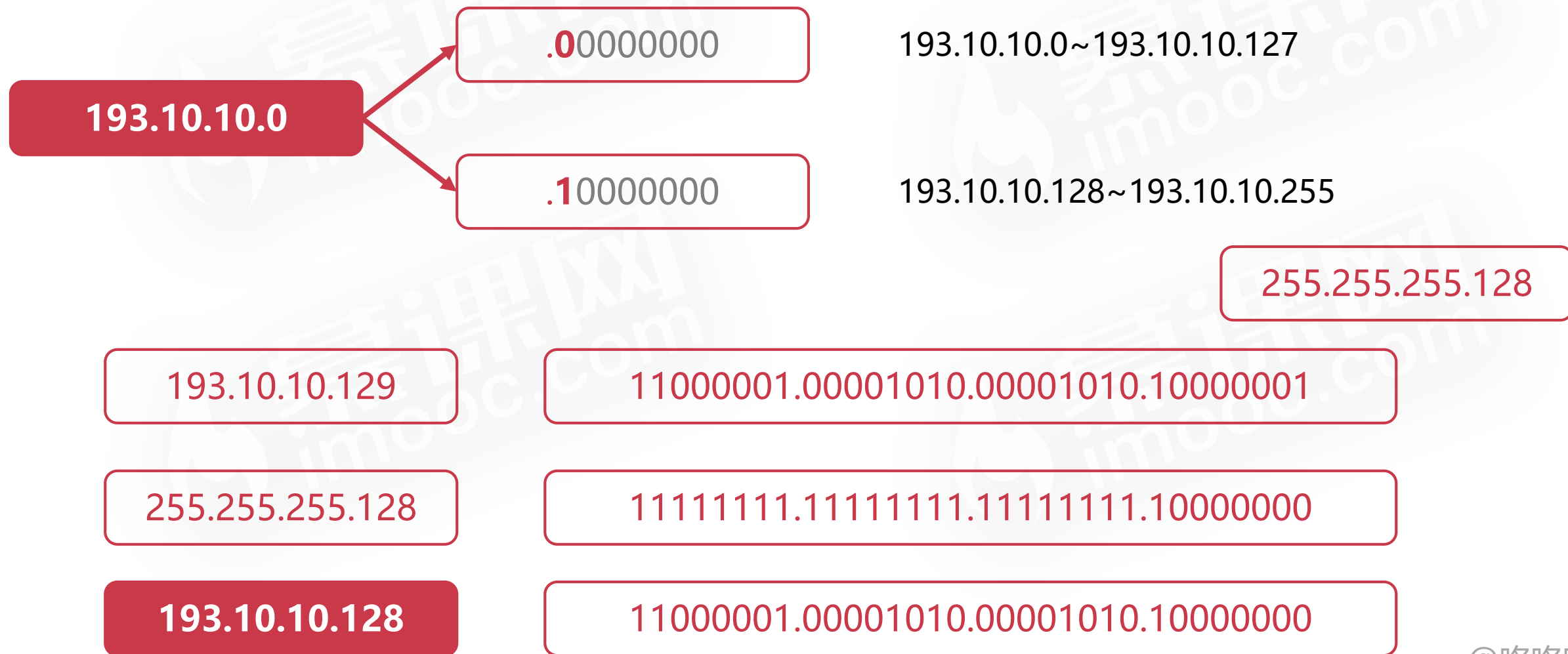


255.255.255.128

划分子网



划分子网



IP地址的子网划分

- ◆ 分类的IP地址
- ◆ 划分子网
- ◆ 无分类编址CIDR

无分类编址CIDR



无分类编址CIDR

- ◆ CIDR中没有A、B、C类网络号、和子网划分的概念
- ◆ CIDR将网络前缀相同的IP地址称为一个 “CIDR地址块”

网络前缀

主机号

网络前缀是任意位数的

无分类编址CIDR

斜线记法

◆ 193.10.10.129/25

11000001.00001010.00001010.10000001

无分类编址CIDR

CIDR前缀长度	掩码点分十进制	地址数
/13	255.248.0.0	512K
/14	255.252.0.0	256K
/15	255.254.0.0	128K
/16	255.255.0.0	64K
/17	255.255.128.0	32K
/18	255.255.192.0	16K
/19	255.255.224.0	8K

相比原来子网划分更加灵活

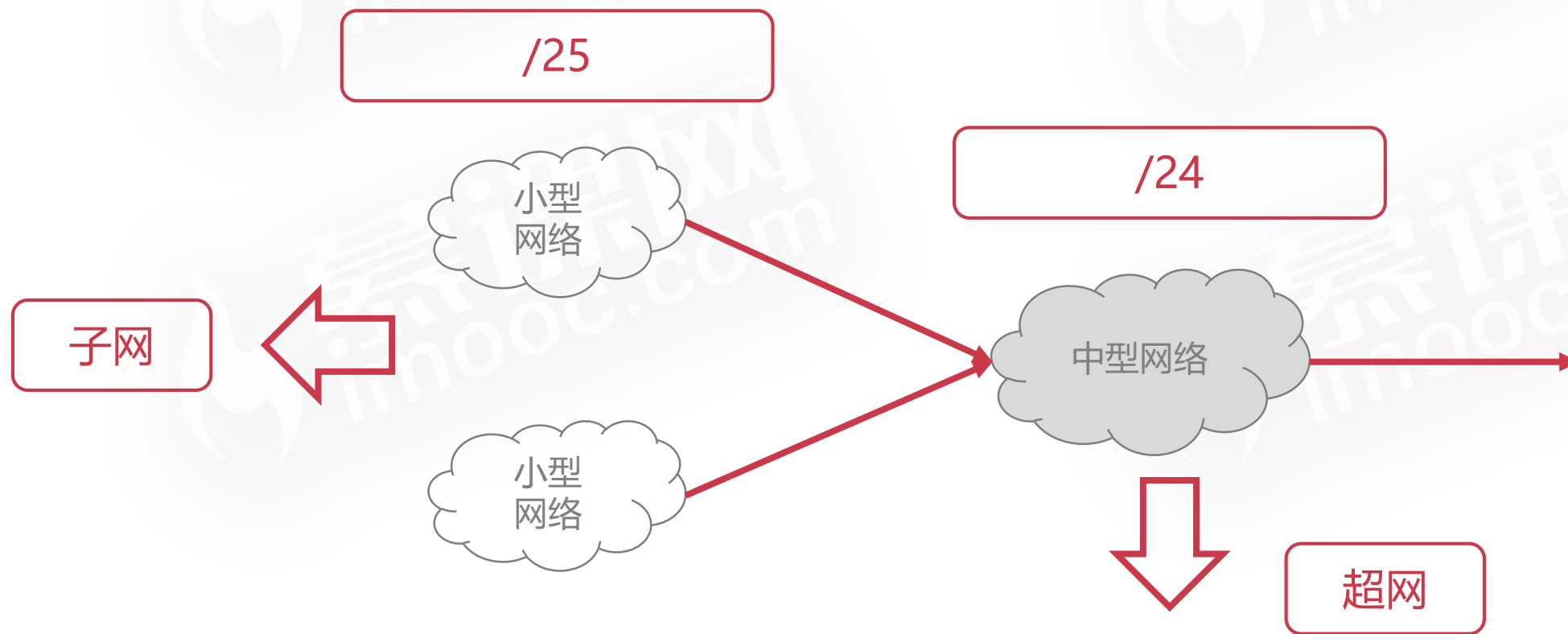
无分类编址CIDR

某公司拥有100名员工，每人配备一个计算机，请问该公司应该申请哪种网络段？

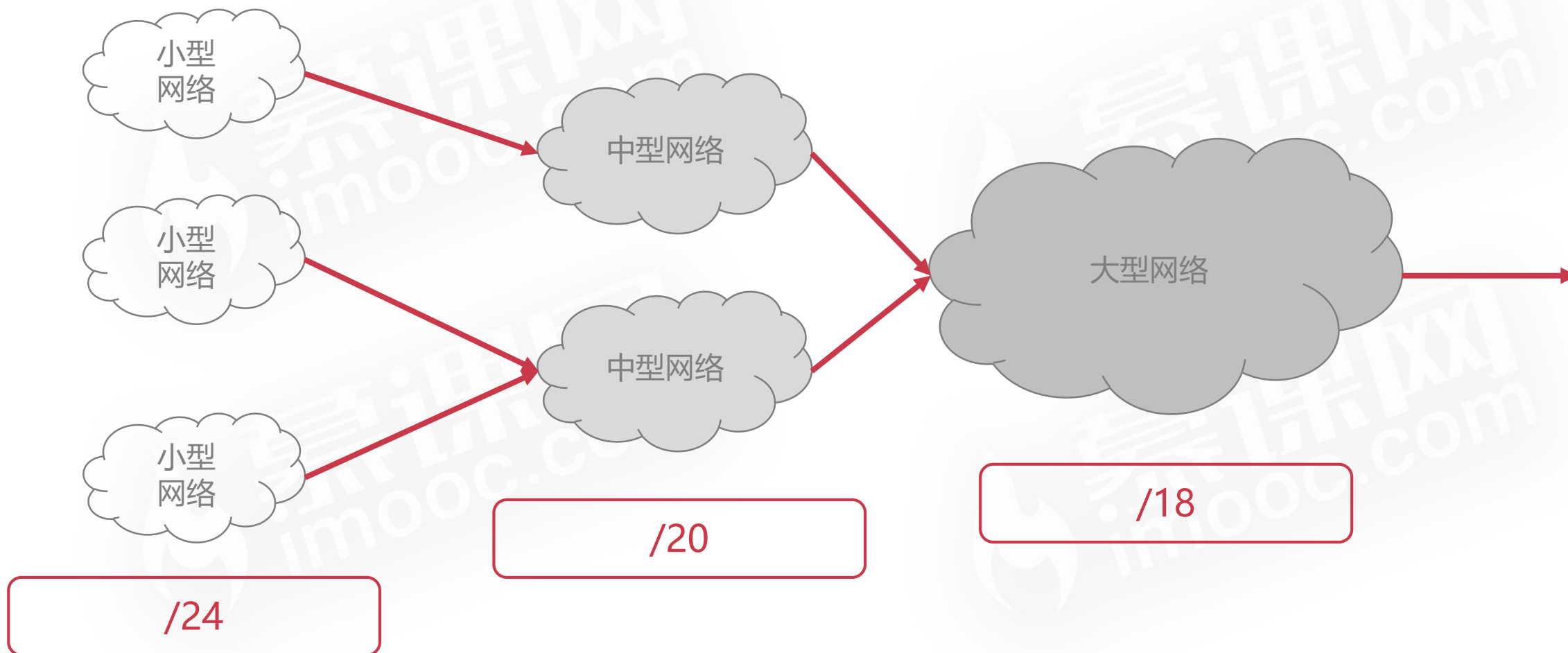
/25

无分类编址CIDR

某公司增加了100名员工，并且拆分成两个部门。



无分类编址CIDR



某个城市的CIDR划分

无分类编址CIDR

- ◆ 使用/18的子网掩码获得超网的网络号
- ◆ 使用/20的子网掩码获得二级超网的网络号
- ◆ 使用/24的子网掩码获得子网的网络号

IP地址的子网划分

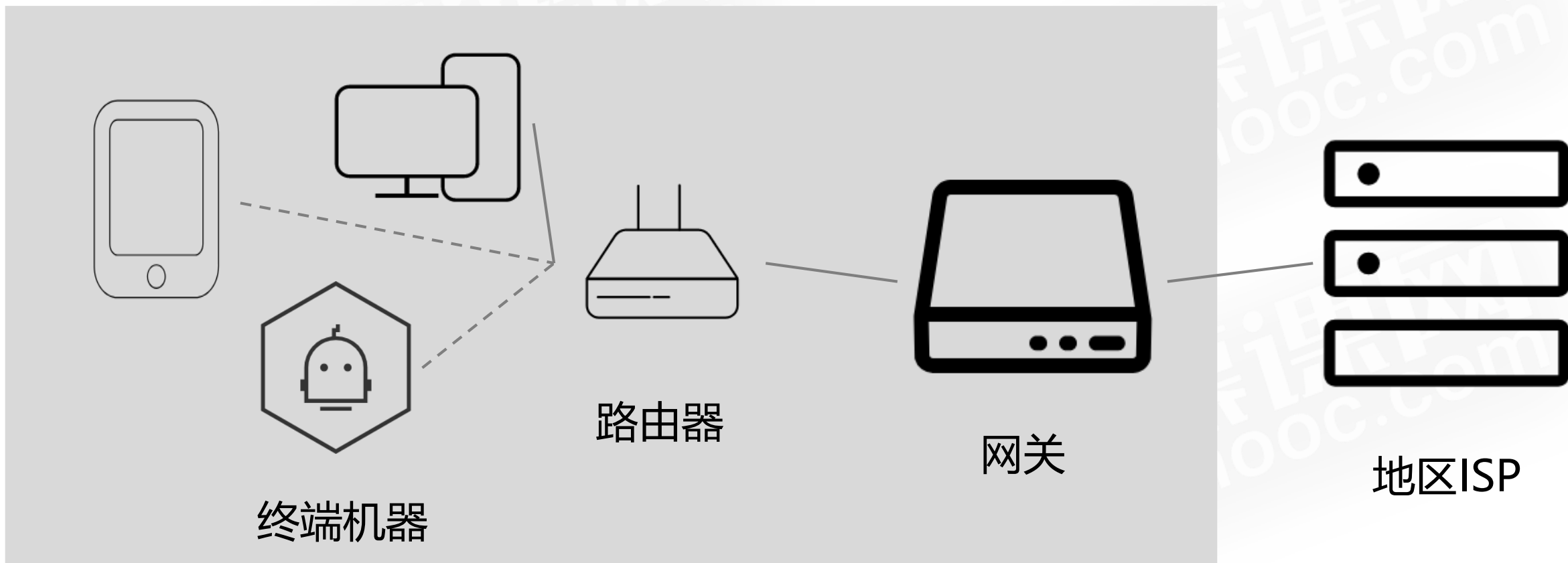
- ◆ 分类的IP地址
- ◆ 划分子网
- ◆ 无分类编址CIDR



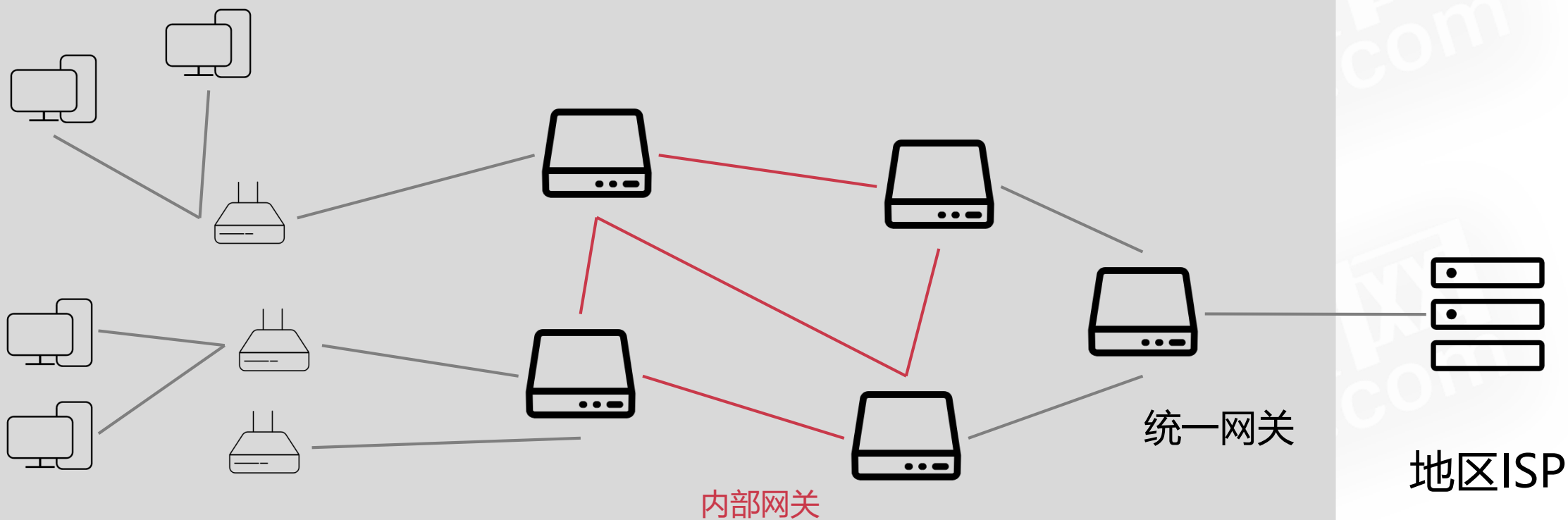
网络地址转换NAT技术

- ◆ IPv4最多只有40+亿个IP地址
- ◆ 早期IP地址的不合理规划导致IP号浪费

网络地址转换NAT技术



网络地址转换NAT技术



边缘部分：企业

网络地址转换NAT技术

内网地址

- ◆ 内部机构使用
- ◆ 避免与外网地址重复

外网地址

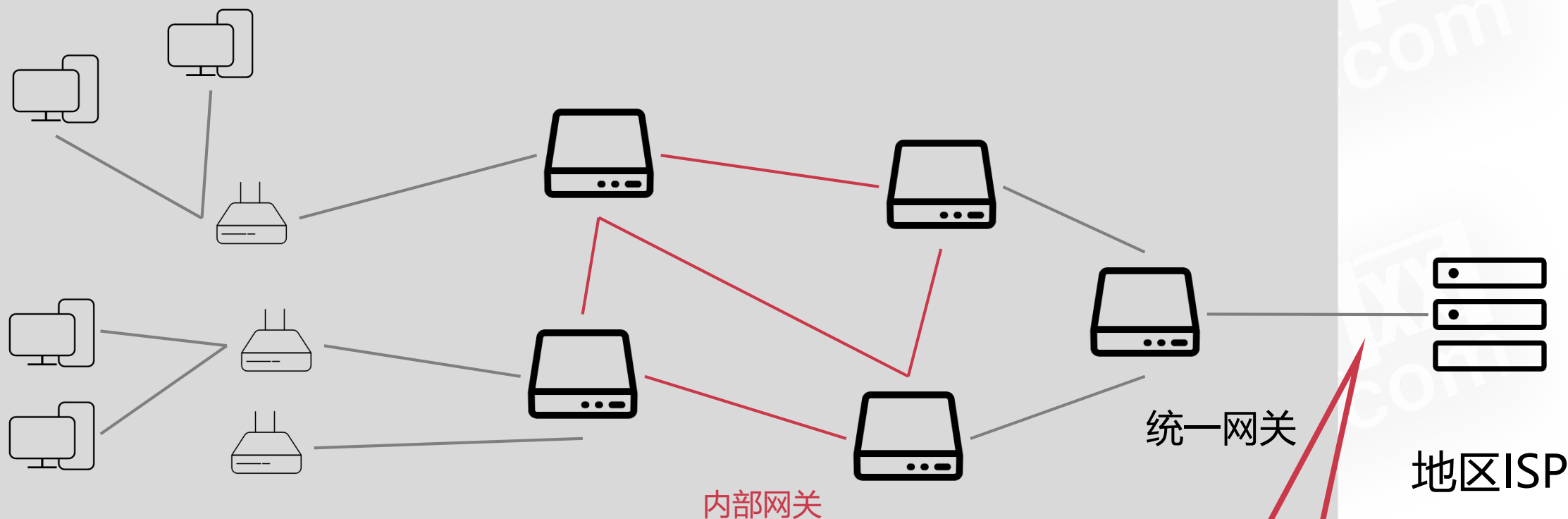
- ◆ 全球范围使用
- ◆ 全球公网唯一

网络地址转换NAT技术

- ◆ 10.0.0.0~10.255.255.255（支持千万数量级设备）
- ◆ 172.16.0.0~172.31.255.255（支持百万数量级设备）
- ◆ 192.168.0.0~192.168.255.255（支持万数量级设备）

三类内网地址

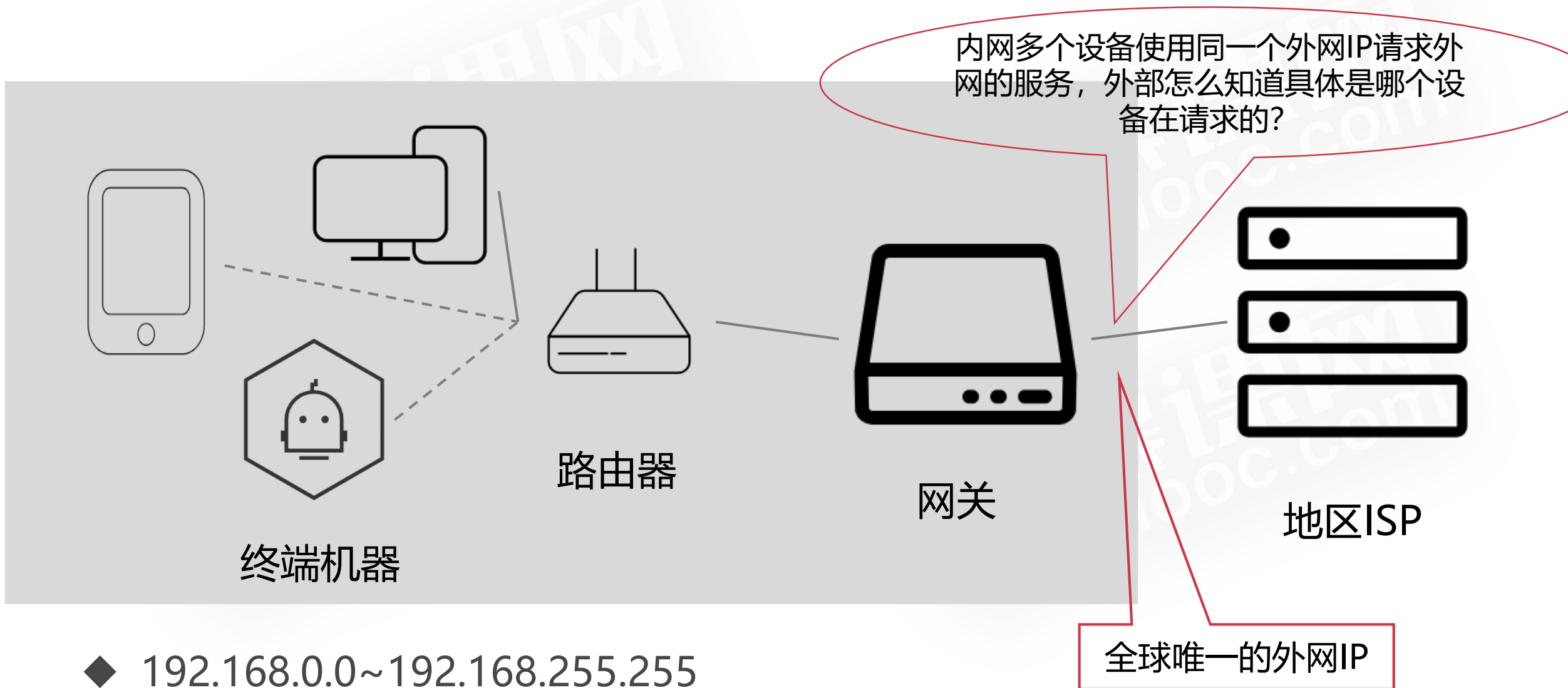
网络地址转换NAT技术



◆ 172.16.0.0~172.31.255.255

全球唯一的外网IP

网络地址转换NAT技术

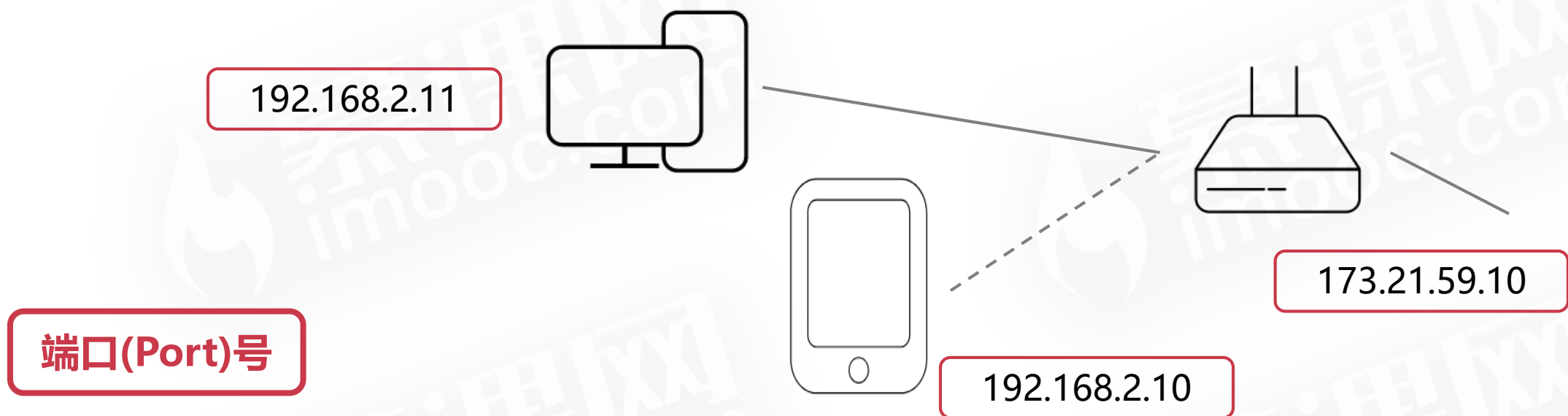


◆ 192.168.0.0~192.168.255.255

网络地址转换NAT技术

- ◆ 网络地址转换NAT(Network Address Translation)
- ◆ NAT技术用于多个主机通过一个公有IP访问互联网的私有网络中

网络地址转换NAT技术



方向	旧的地址和端口号	新的地址与端口号
出	192.168.2.11:6666	173.21.59.10:16666
出	192.168.2.10:7777	173.21.59.10:17777
入	173.21.59.10:16666	192.168.2.11:6666
入	173.21.59.10:17777	192.168.2.10:7777

NA(P)T表

网络地址转换NAT技术

- ◆ 网络地址转换NAT(Network Address Translation)
- ◆ NAT技术用于多个主机通过一个公有IP访问互联网的私有网络中
- ◆ NAT减缓了IP地址的消耗，但是增加了网络通信的复杂度



ICMP协议详解

- ◆ 网际控制报文协议 (Internet Control Message Protocol)
- ◆ ICMP协议可以报告错误信息或者异常情况

ICMP协议详解



ICMP协议详解

8位类型	8位代码	16位校验和
ICMP报文数据		

ICMP协议详解

协议：占8位，表明IP数据所携带的具体数据是什么协议的
(如：TCP、UDP等)

4位版本	4位首部长度	8位服务类型(TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

协议名	ICMP	IGMP	IP	TCP	UDP	OSPF	...
字段值	1	2	4	6	17	89	...

ICMP协议详解

差错报告报文

询问报文

ICMP协议详解

8位类型	8位代码	16位校验和
ICMP报文数据		

ICMP报文种类	类型的值	报文类型	具体代码
差错报告报文	3 (终点不可达)	网络不可达	0
		主机不可达	1
	5 (重定向)	对网络重定向	0
		对主机重定向	1
	11	传输超时	-
	12	坏的IP头	0
		缺少其他必要参数	1

ICMP协议详解

8位类型	8位代码	16位校验和
ICMP报文数据		

ICMP报文种类	类型的值	报文类型	具体代码
询问报文	0或8	回送(Echo)请求或 应答	-
	13或14	时间戳 (Timestamp)请求 或应答	-

ICMP协议详解

差错报告报文

询问报文

ICMP协议详解





ICMP协议的应用

◆ Ping应用

◆ Traceroute应用

Ping应用

ICMP报文种类	类型的值	报文类型	具体代码
询问报文	0或8	回送(Echo)请求或 应答	-
	13或14	时间戳 (Timestamp)请求 或应答	-

Ping应用



Ping应用

- ◆ Ping回环地址127.0.0.1
- ◆ Ping网关地址
- ◆ Ping远端地址

ICMP协议的应用

◆ Ping应用

◆ Traceroute应用

Traceroute应用

Traceroute可以探测IP数据报在网络中走过的路径

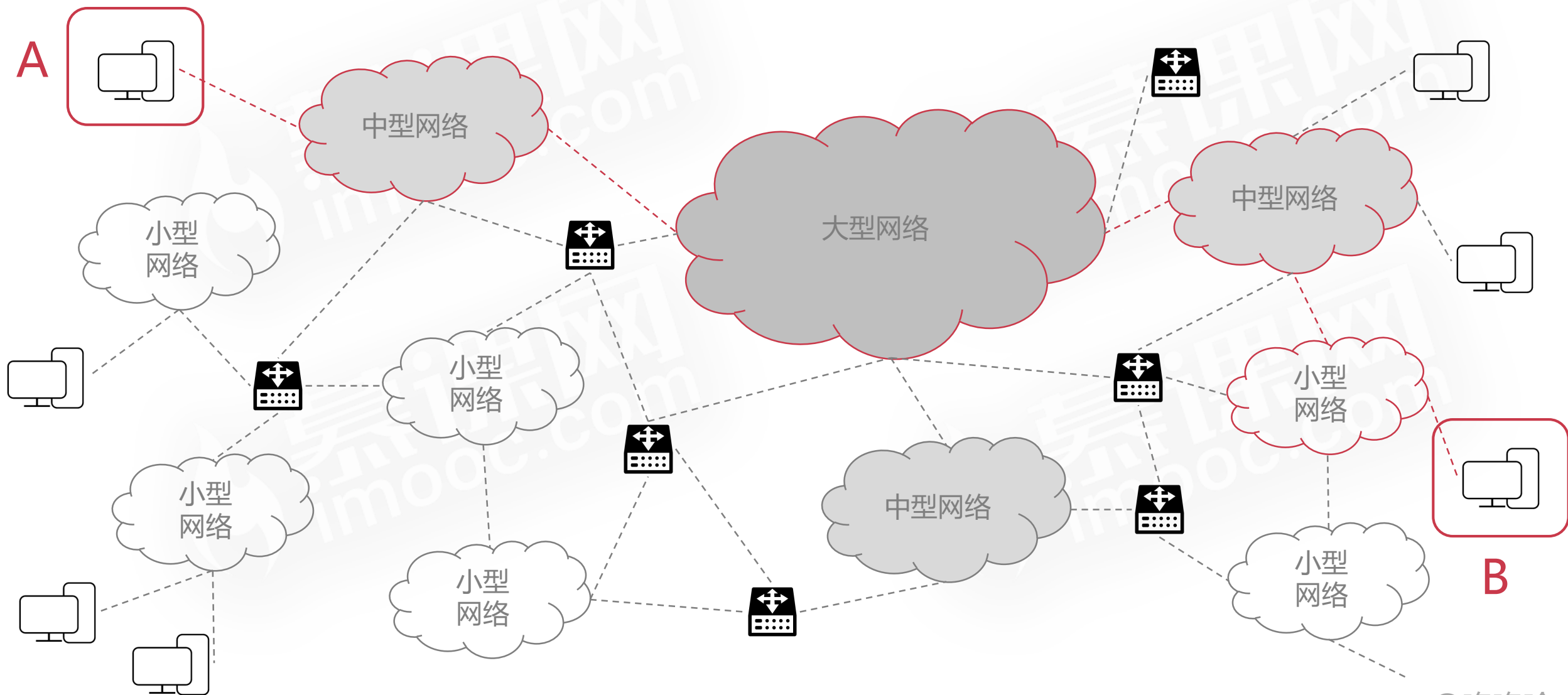
Traceroute应用

TTL: 占8位, 表明IP数据报文在网络中的寿命, 每经过一个设备, TTL减1, 当TTL=0时, 网络设备必须丢弃该报文

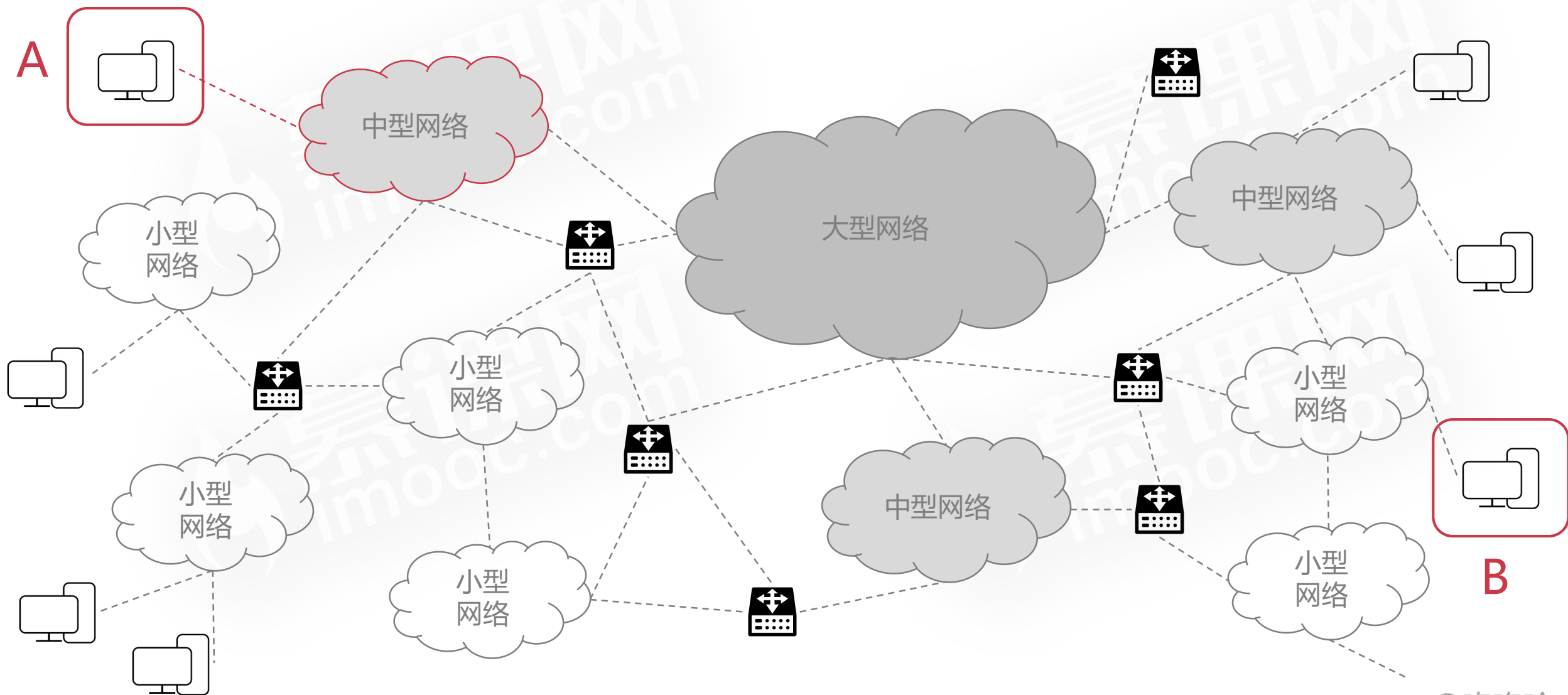
ICMP终点不可达差错报文

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

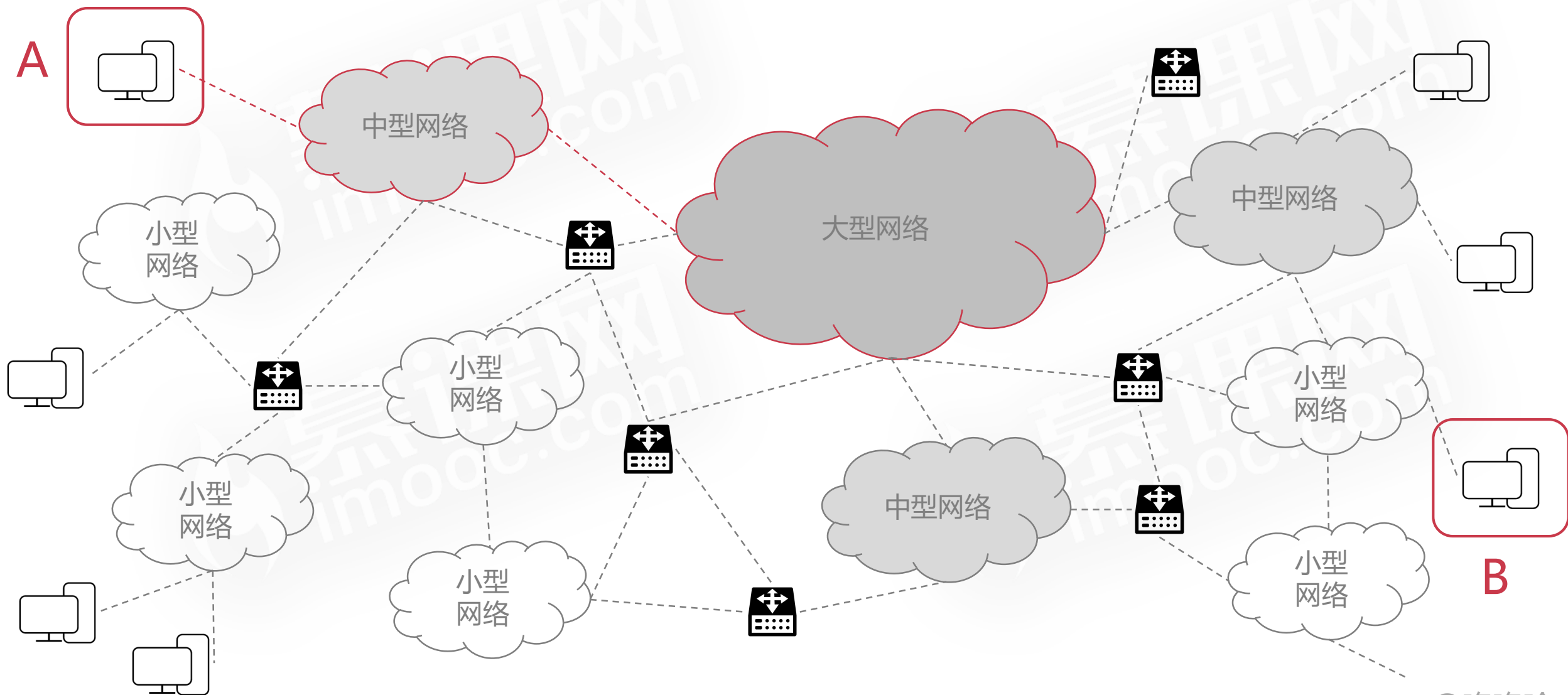
Traceroute应用



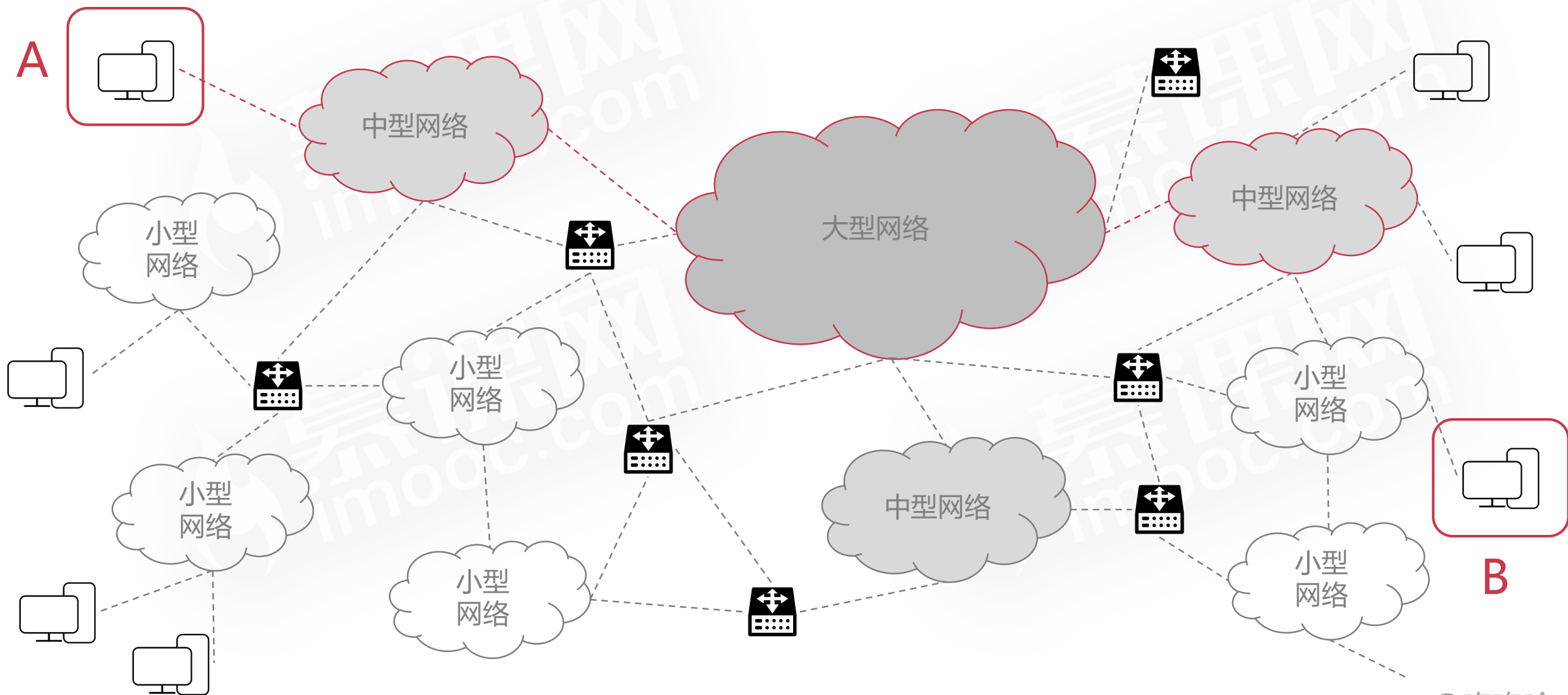
Traceroute应用



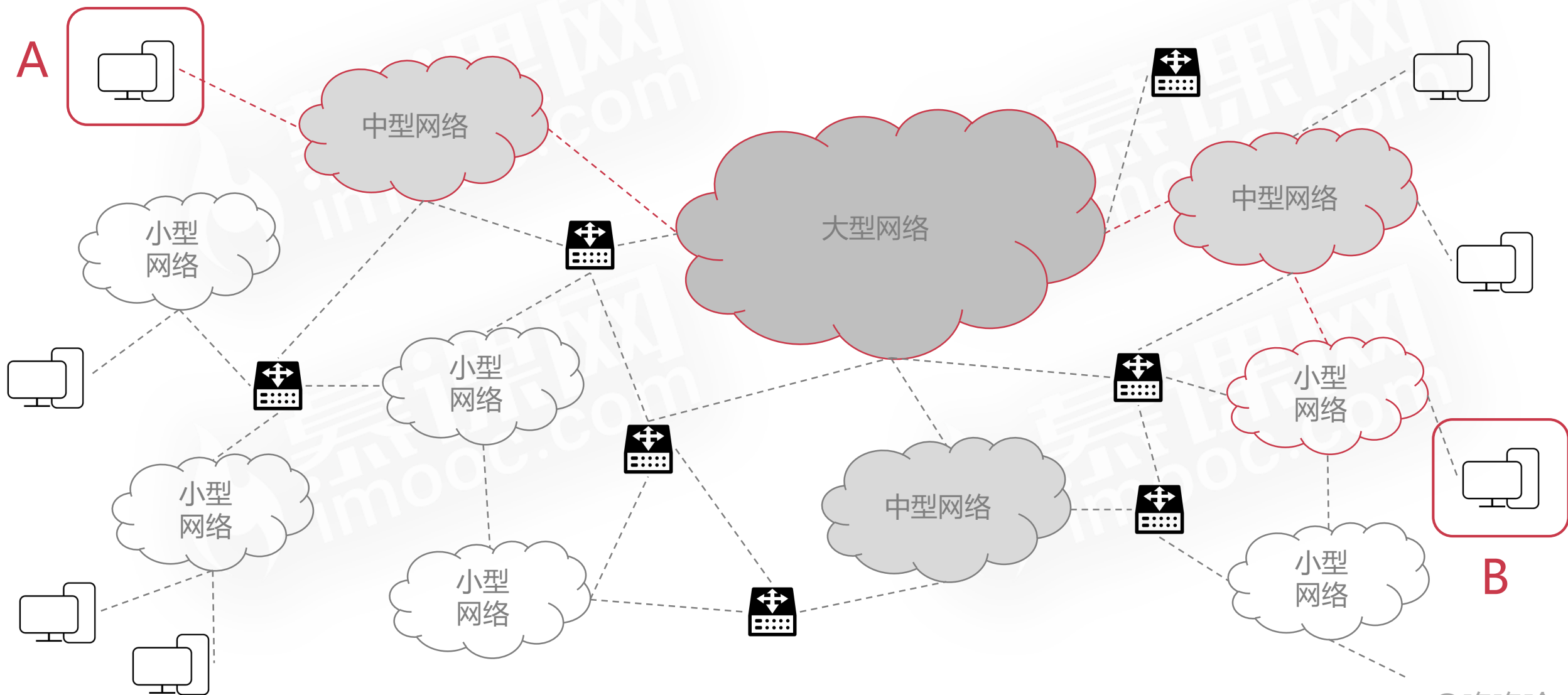
Traceroute应用



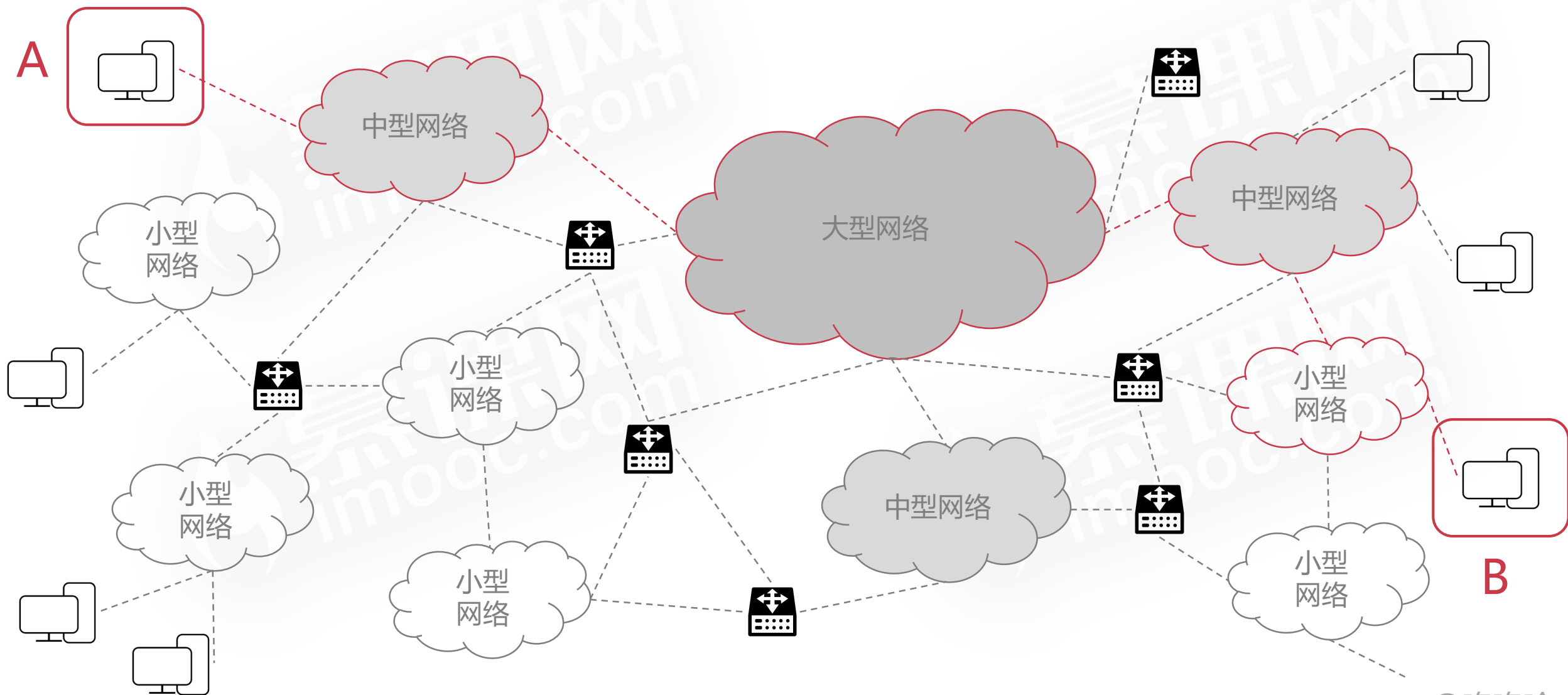
Traceroute应用



Traceroute应用



Traceroute应用



Traceroute应用

TTL: 占8位, 表明IP数据报文在网络中的寿命, 每经过一个设备, TTL减1, 当TTL=0时, 网络设备必须丢弃该报文

ICMP终点不可达差错报文

4位版本	4位首部长度	8位服务类型 (TOS)	16位总长度(字节)	
16位标识			3位标志	13位片偏移
8位生存时间(TTL)		8位协议	16位首部校验和	
32位源IP地址				
32位目的IP地址				
选项options (若有)				
IP数据				

ICMP协议的应用

◆ Ping应用

◆ Traceroute应用

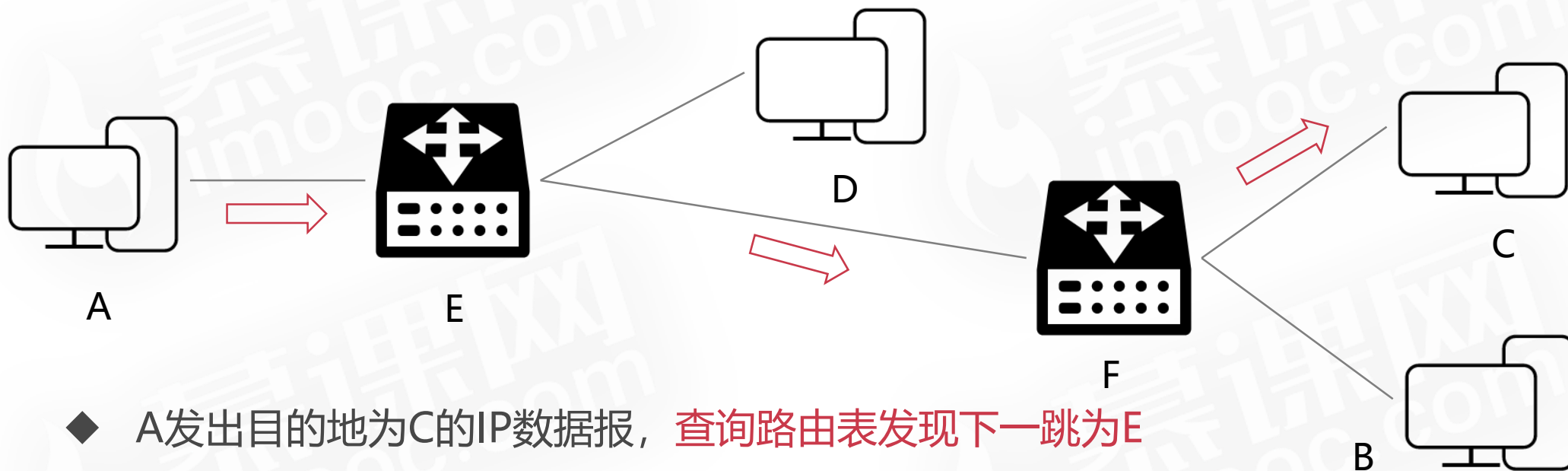
慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

网络层的路由概述



- ◆ A发出目的地为C的IP数据报，**查询路由表发现下一跳为E**
- ◆ A将数据报发送给E
- ◆ E**查询路由表发现下一跳为F**，将数据报发送给F
- ◆ F**查询路由表发现目的地C直接连接**，将数据报发送给C

网络层的路由概述

目的IP地址	下一跳IP地址
IP1	IP4
IP2	IP5
IP3	IP6
...	...

路由表是怎么来的呢？



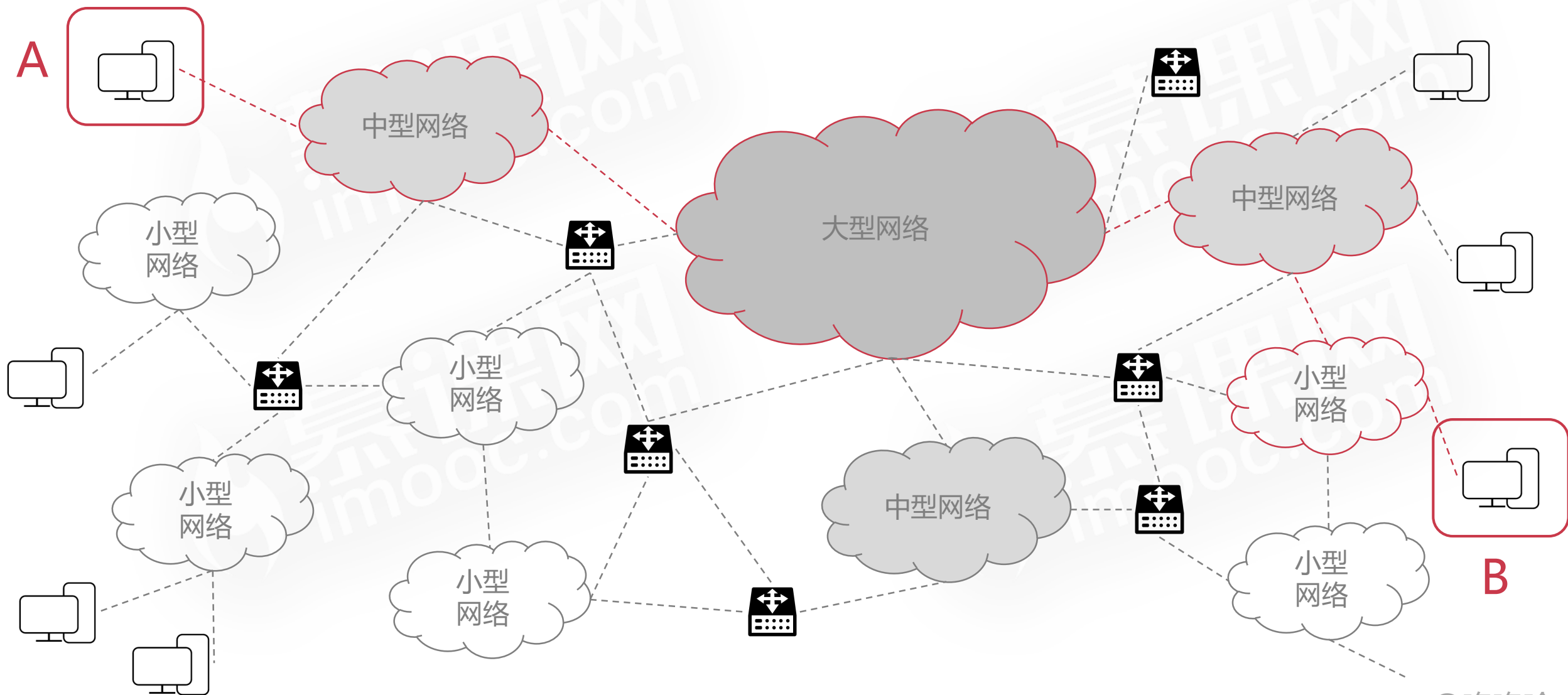
网络层的路由概述

目的IP地址	下一跳IP地址
IP1	IP4
IP2	IP5
IP3	IP6
...	...

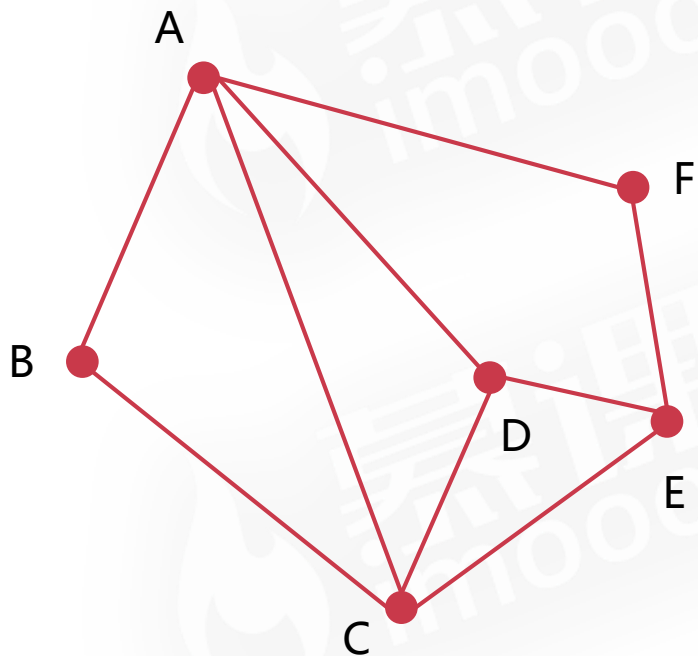
- ◆ 下一跳地址是怎么来的?
- ◆ 下一跳地址是唯一的吗?
- ◆ 下一跳地址是最佳的吗?
- ◆ 路由器怎么多, 他们是怎么协同工作的?

需要一个好的算法去解决这些事情

网络层的路由概述



网络层的路由概述



- ◆ 每一顶点表示一个网络、路由器或计算机
- ◆ 每一条边表示一条网络路径

路由算法实际上是图论的算法

- ◆ 网络环境复杂

路由算法要比图论的算法要复杂

网络层的路由概述

- ◆ 算法是**正确的**、**完整的**
- ◆ 算法在计算上应该尽可能的**简单**
- ◆ 算法可以**适应网络中的变化**
- ◆ 算法是**稳定的**和**公平的**

网络层的路由概述

- ◆ 互联网的规模是非常大的
- ◆ 互联网环境是非常复杂的

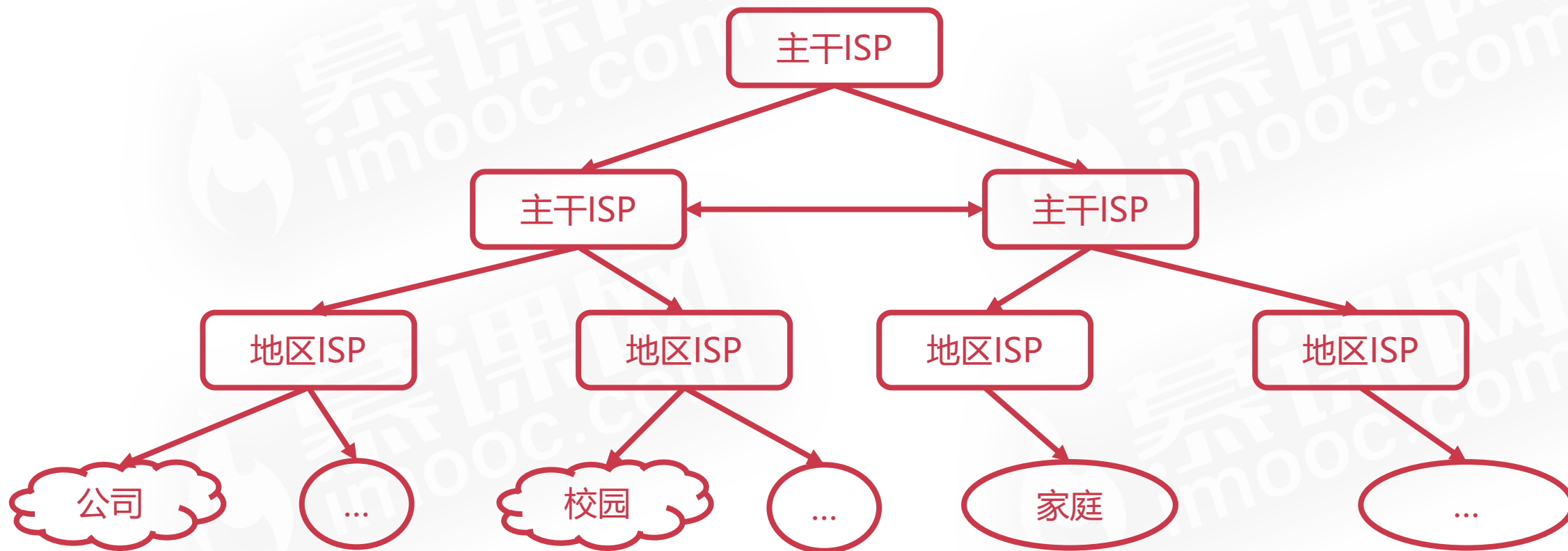
对互联网进行划分

网络层的路由概述

- ◆ 一个自治系统(AS)是处于一个管理机构下的网络设备群
- ◆ AS内部网络自行管理，AS对外提供一个或者多个出（入）口

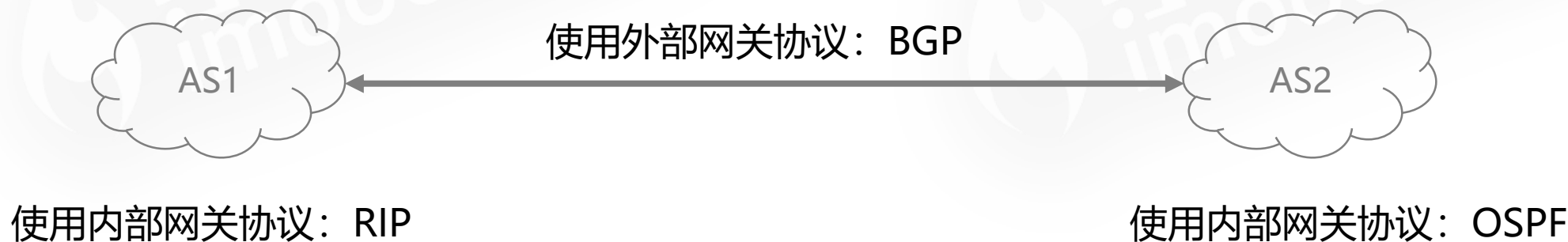
自治系统(Autonomous System)

网络层的路由概述



网络层的路由概述

- ◆ 自治系统内部路由的协议称为：内部网关协议(RIP、OSPF)
- ◆ 自治系统外部路由的协议称为：外部网关协议(BGP)



网络层的路由概述

- ◆ 路由算法的本质
- ◆ 自治系统



内部网关路由协议之RIP协议

- ◆ 距离矢量（DV）算法
- ◆ RIP协议的过程

距离矢量 (DV) 算法

- ◆ 每一个节点使用两个向量 D_i 和 S_i
- ◆ D_i 描述的是当前节点到别的节点的距离
- ◆ S_i 描述的是当前节点到别的节点的下一节点

距离矢量 (DV) 算法

- ◆ 每一个节点与相邻的节点交换向量 D_i 和 S_i 的信息
- ◆ 每一个节点根据交换的信息更新自己的节点信息

距离矢量 (DV) 算法

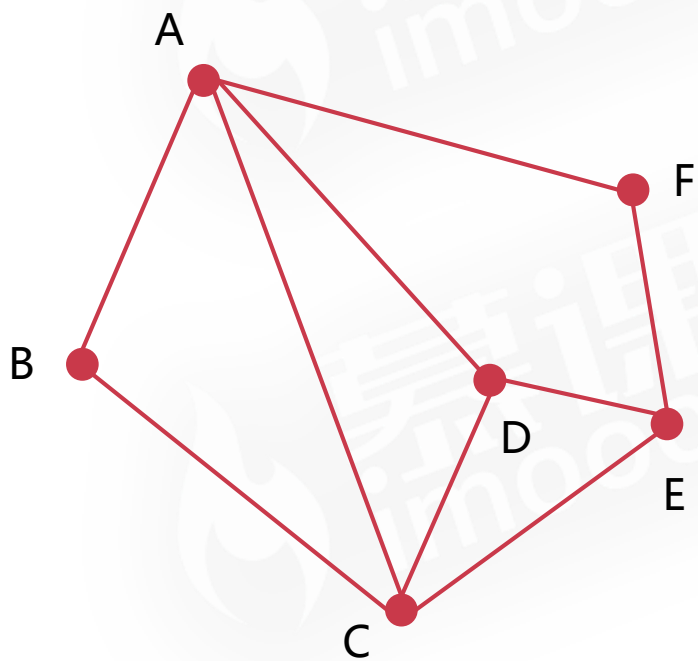
- ◆ d_{i1} 表示从节点 i 到节点 1 的距离
- ◆ s_{i1} 表示从节点 i 到节点 1 的下一个节点
- ◆ n 表示节点的数量

$$D_i = \begin{pmatrix} d_{i1} \\ d_{i2} \\ d_{i3} \\ d_{i4} \\ d_{\dots} \\ d_{in} \end{pmatrix}$$

$$S_i = \begin{pmatrix} s_{i1} \\ s_{i2} \\ s_{i3} \\ s_{i4} \\ s_{\dots} \\ s_{in} \end{pmatrix}$$

$$d_{ij} = \min(d_{ix} + d_{xj})$$

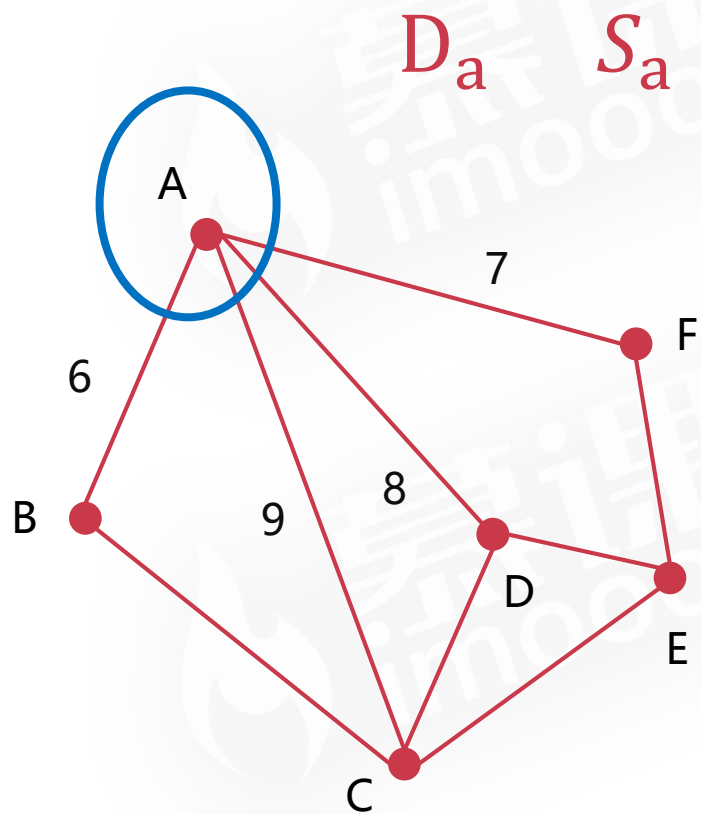
距离矢量 (DV) 算法



$$D_i = \begin{pmatrix} d_{ia} \\ d_{ib} \\ d_{ic} \\ d_{id} \\ d_{ie} \\ d_{if} \end{pmatrix}$$

$$S_i = \begin{pmatrix} S_{ia} \\ S_{ib} \\ S_{ic} \\ S_{id} \\ S_{ie} \\ S_{if} \end{pmatrix}$$

距离矢量 (DV) 算法



$AB=6$ $AC=9$ $AD=8$ $AF=7$

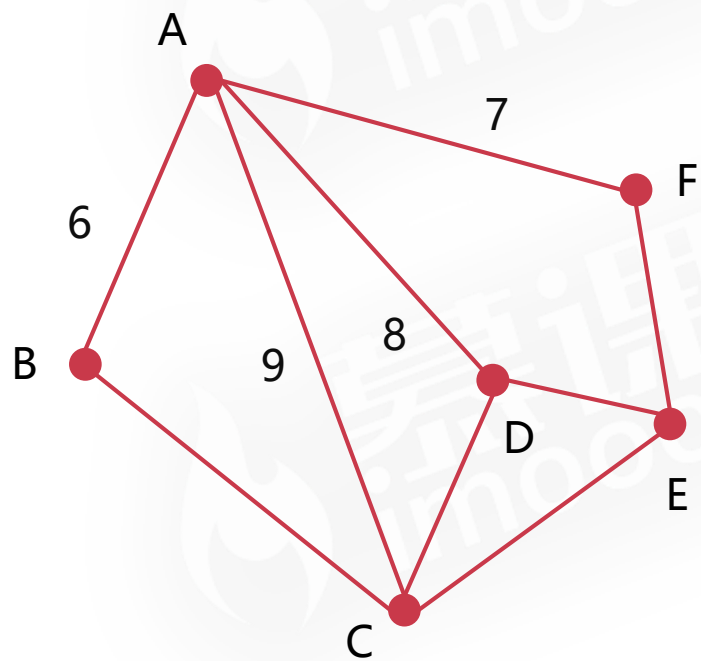
	A
A	0
B	11
C	12
D	10
E	21
F	17

A的距离矢量信息

	B	C	D	F
A	9	12	9	9
B	0	9	8	15
C	11	0	6	11
D	7	8	0	10
E	17	11	11	13
F	11	10	10	0

收到的距离矢量信息

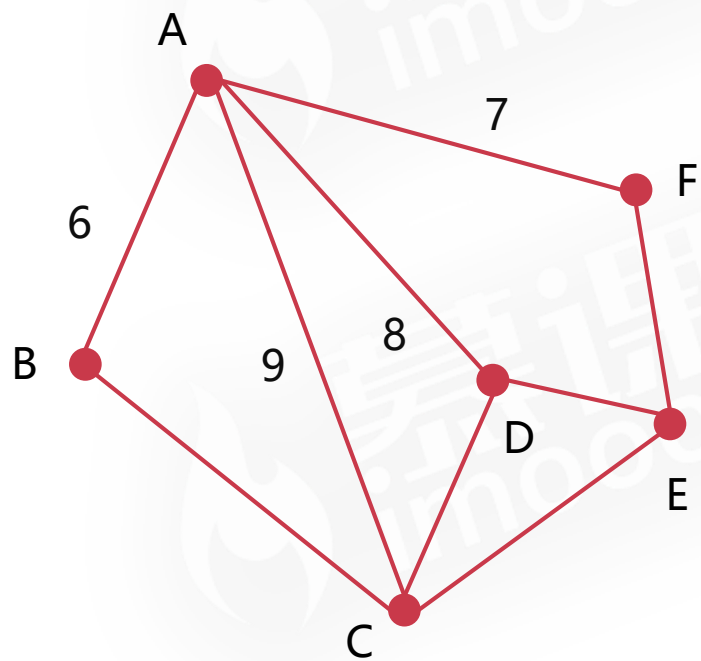
距离矢量 (DV) 算法



	A	B	C	D	F
A	0	9	12	9	9
B	11	0	9	8	15
C	12	11	0	6	11
D	10	7	8	0	10
E	21	17	11	11	13
F	17	11	10	10	0

AB=6 AC=9 AD=8 AF=7

距离矢量 (DV) 算法

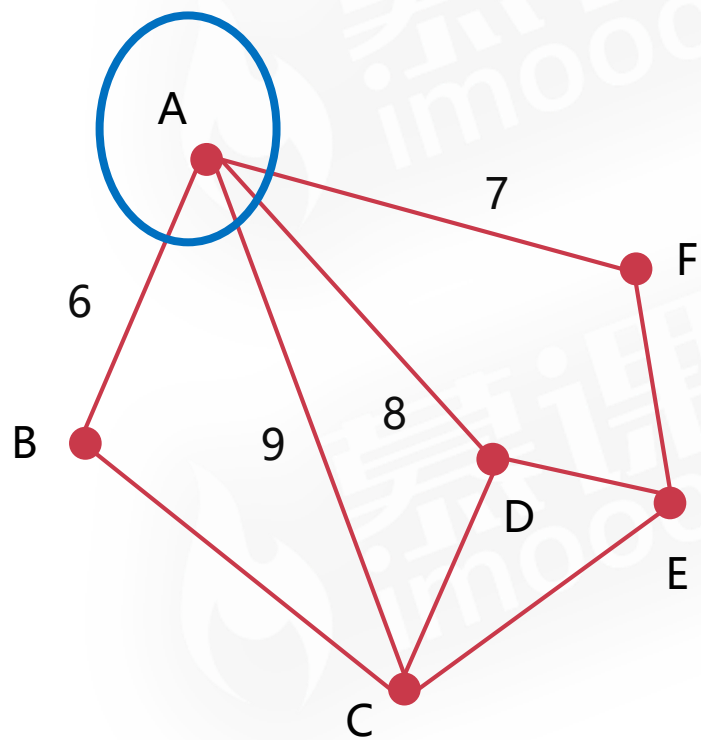


A->C->B

B->C->D->A

	A	B	C	D	F
A	0	9	12	9	9
B	11	0	9	8	15
C	12	11	0	6	11
D	10	7	8	0	10
E	21	17	11	11	13
F	17	11	10	10	0

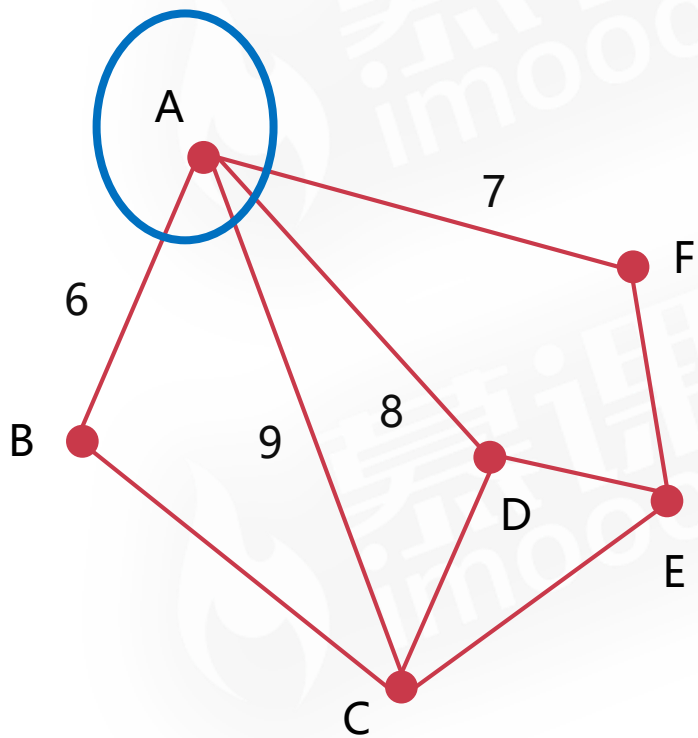
距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	11	0	9	8	15	-
C	12	11	0	6	11	-
D	10	7	8	0	10	-
E	21	17	11	11	13	-
F	17	11	10	10	0	-

AB=6 AC=9 AD=8 AF=7

距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	11	0	9	8	15	-
C	12	11	0	6	11	-
D	10	7	8	0	10	-
E	21	17	11	11	13	-
F	17	11	10	10	0	-

$A \rightarrow B = 6$

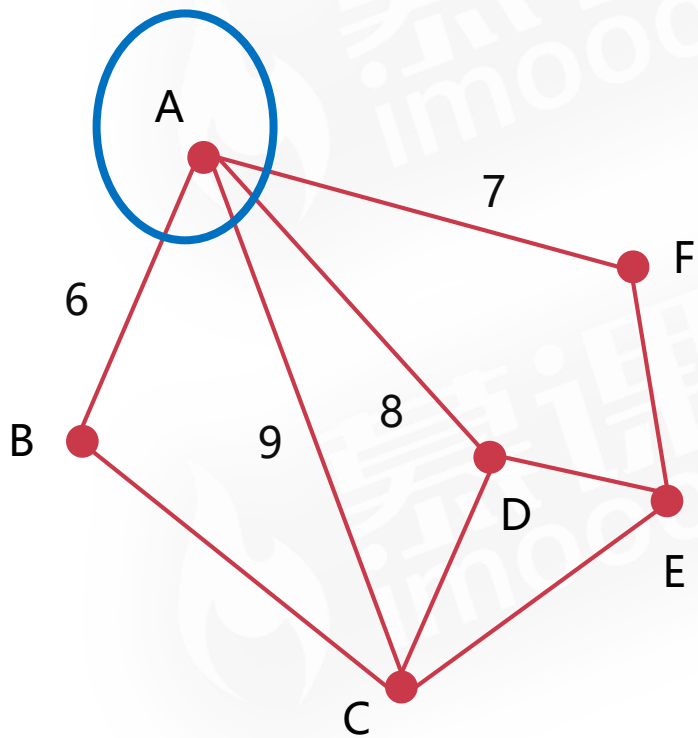
$A \rightarrow B \rightarrow C = 6 + 11 = 17$

$A \rightarrow B \rightarrow D = 6 + 7 = 13$

$A \rightarrow B \rightarrow E = 6 + 17 = 23$

$A \rightarrow B \rightarrow F = 6 + 11 = 17$

距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	6	0	9	8	15	B
C	12	11	0	6	11	-
D	10	7	8	0	10	-
E	21	17	11	11	13	-
F	17	11	10	10	0	B

$A \rightarrow B = 6$

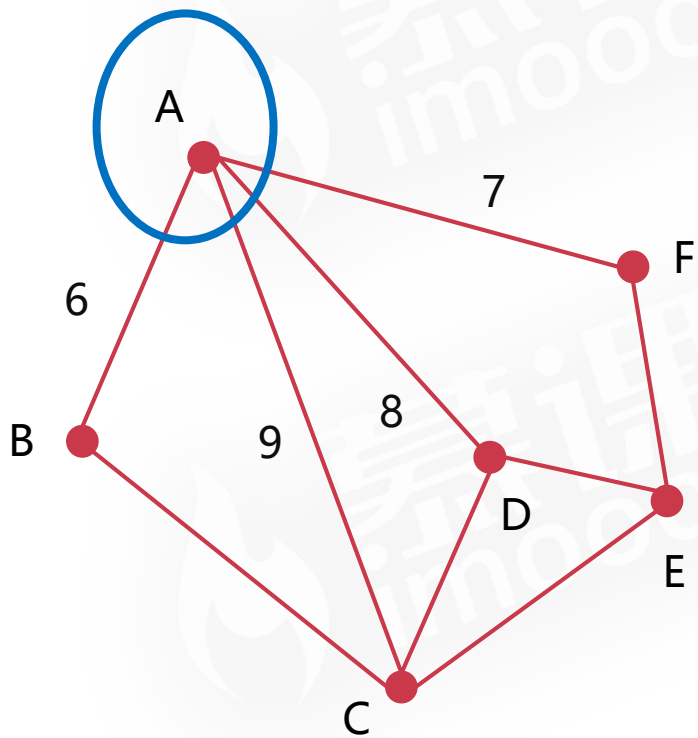
$A \rightarrow B \rightarrow C = 6 + 11 = 17$

$A \rightarrow B \rightarrow D = 6 + 7 = 13$

$A \rightarrow B \rightarrow E = 6 + 17 = 23$

$A \rightarrow B \rightarrow F = 6 + 11 = 17$

距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	6	0	9	8	15	B
C	12	11	0	6	11	-
D	10	7	8	0	10	-
E	21	17	11	11	13	-
F	17	11	10	10	0	B

$A \rightarrow C \rightarrow B = 9 + 9 = 18$

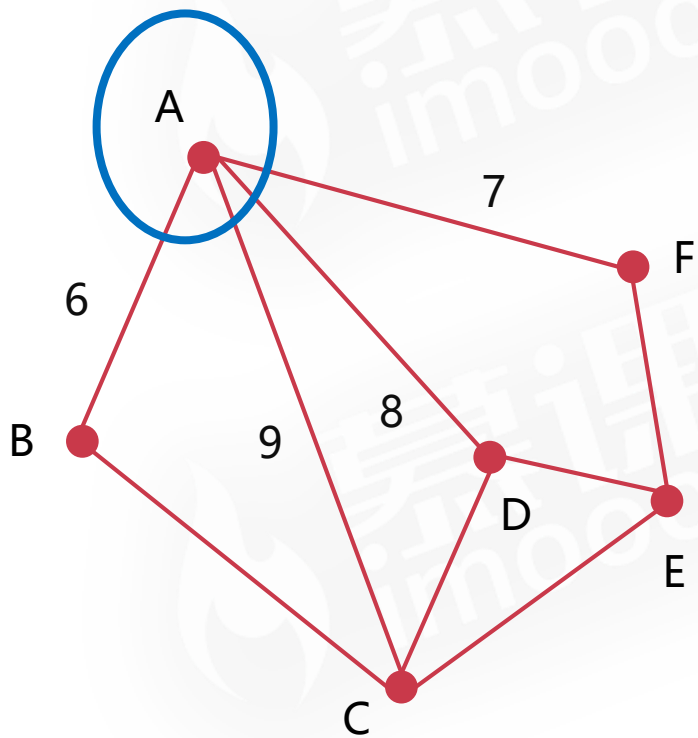
$A \rightarrow C = 9$

$A \rightarrow C \rightarrow D = 9 + 8 = 17$

$A \rightarrow C \rightarrow E = 9 + 11 = 20$

$A \rightarrow C \rightarrow F = 9 + 10 = 19$

距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	6	0	9	8	15	B
C	9	11	0	6	11	C
D	10	7	8	0	10	-
E	20	17	11	11	13	C
F	17	11	10	10	0	B

$A \rightarrow C \rightarrow B = 9 + 9 = 18$

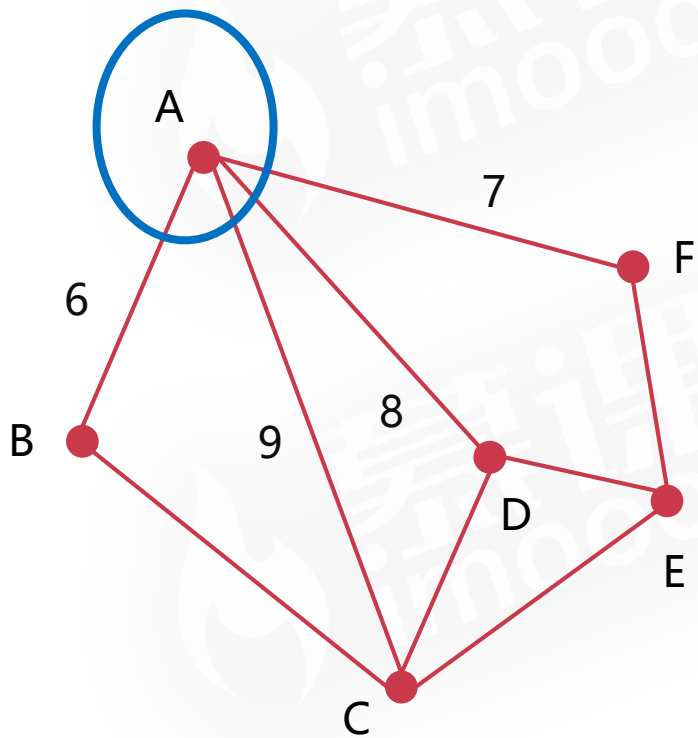
$A \rightarrow C = 9$

$A \rightarrow C \rightarrow D = 9 + 8 = 17$

$A \rightarrow C \rightarrow E = 9 + 11 = 20$

$A \rightarrow C \rightarrow F = 9 + 10 = 19$

距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	6	0	9	8	15	B
C	9	11	0	6	11	C
D	10	7	8	0	10	-
E	20	17	11	11	13	C
F	17	11	10	10	0	B

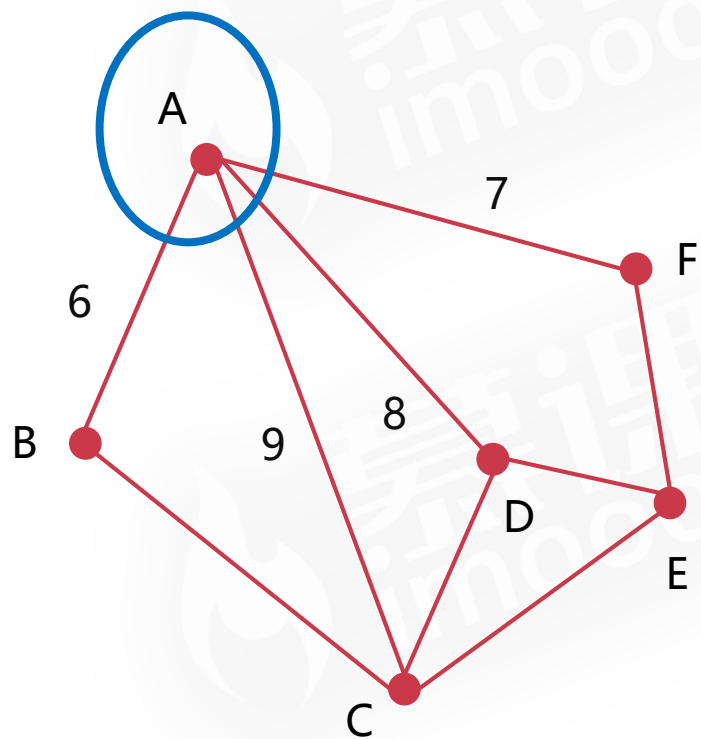
$A \rightarrow D \rightarrow B = 8 + 8 = 16$ $A \rightarrow D \rightarrow C = 8 + 6 = 14$

$A \rightarrow D = 8$

$A \rightarrow D \rightarrow E = 8 + 11 = 19$

$A \rightarrow D \rightarrow F = 8 + 10 = 18$

距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	6	0	9	8	15	B
C	9	11	0	6	11	C
D	8	7	8	0	10	D
E	19	17	11	11	13	D
F	17	11	10	10	0	B

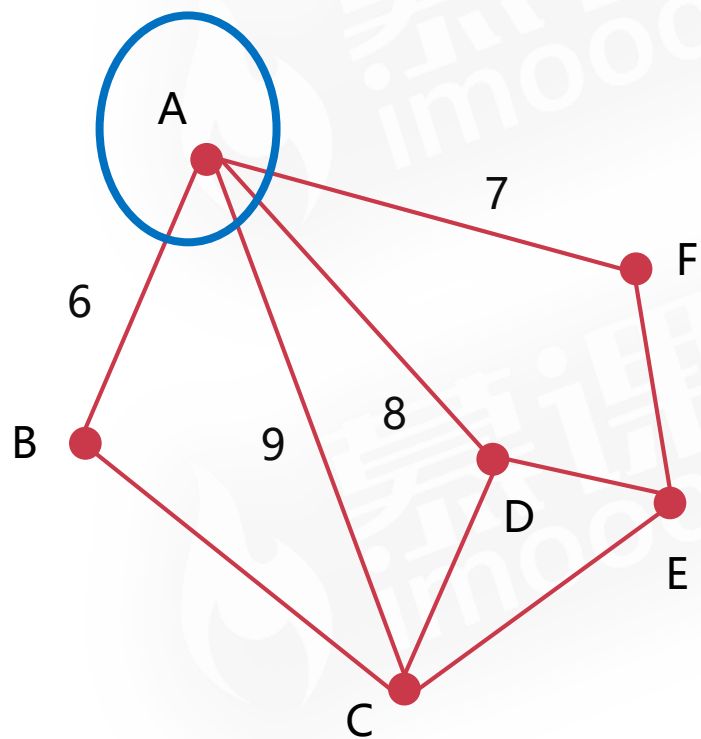
$$A \rightarrow D \rightarrow B = 8 + 8 = 16 \quad A \rightarrow D \rightarrow C = 8 + 6 = 14$$

$$A \rightarrow D = 8$$

$$A \rightarrow D \rightarrow E = 8 + 11 = 19$$

$$A \rightarrow D \rightarrow F = 8 + 10 = 18$$

距离矢量 (DV) 算法

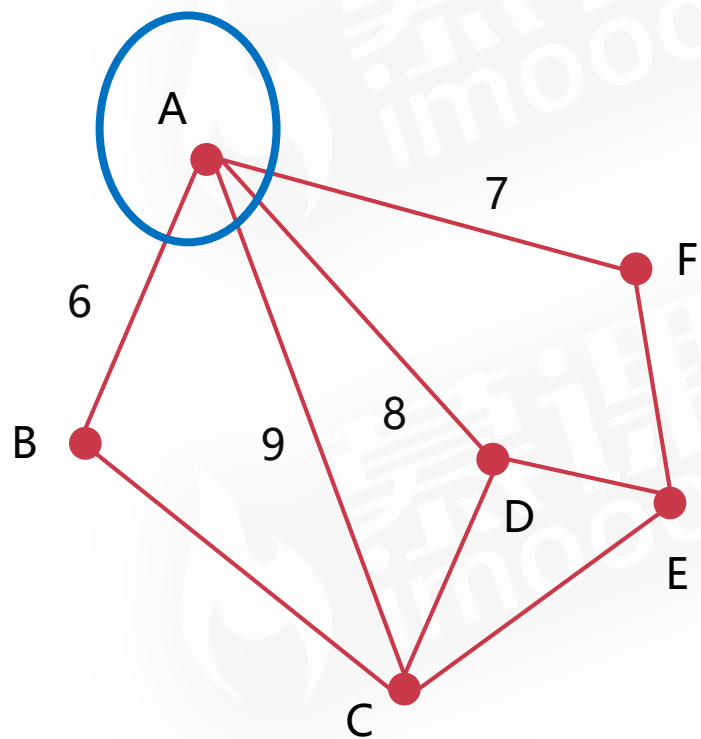


	A	B	C	D	F	S
A	0	9	12	9	9	-
B	6	0	9	8	15	B
C	9	11	0	6	11	C
D	8	7	8	0	10	D
E	19	17	11	11	13	D
F	17	11	10	10	0	B

$A \rightarrow F \rightarrow B = 7 + 15 = 22$ $A \rightarrow F \rightarrow C = 7 + 11 = 18$ $A \rightarrow F \rightarrow D = 7 + 10 = 17$ $A \rightarrow F \rightarrow E = 7 + 13 = 20$

$A \rightarrow F = 7$

距离矢量 (DV) 算法



	A	B	C	D	F	S
A	0	9	12	9	9	-
B	6	0	9	8	15	B
C	9	11	0	6	11	C
D	8	7	8	0	10	D
E	19	17	11	11	13	D
F	7	11	10	10	0	F

$A \rightarrow F \rightarrow B = 7 + 15 = 22$ $A \rightarrow F \rightarrow C = 7 + 11 = 18$ $A \rightarrow F \rightarrow D = 7 + 10 = 17$ $A \rightarrow F \rightarrow E = 7 + 13 = 20$

$A \rightarrow F = 7$

距离矢量 (DV) 算法

- ◆ 每一个节点使用两个向量 D_i 和 S_i
- ◆ D_i 描述的是当前节点到别的节点的距离
- ◆ S_i 描述的是当前节点到别的节点的下一节点

距离矢量 (DV) 算法

- ◆ 每一个节点与相邻的节点交换向量 D_i 和 S_i 的信息
- ◆ 每一个节点根据交换的信息更新自己的节点信息

内部网关路由协议之RIP协议

- ◆ 距离矢量（DV）算法
- ◆ RIP协议的过程

RIP协议的过程

- ◆ RIP(Routing Information Protocol)协议
- ◆ RIP协议是使用DV算法的一种路由协议

RIP协议的过程

- ◆ RIP协议把网络的跳数(hop)作为DV算法的距离
- ◆ RIP协议每隔30s交换一次路由信息
- ◆ RIP协议认为跳数>15的路由则为不可达路由

RIP协议的过程

1. 路由器初始化路由信息(两个向量 D_i 和 S_i)
2. 对相邻路由器X发过来的信息，对信息的内容进行修改（下一跳地址设置为X，所有距离加1）
 - i. 检索本地路由，将信息中新的路由插入到路由表里面
 - ii. 检索本地路由，对于下一跳为X的，更新为修改后的信息
 - iii. 检索本地路由，对比相同目的的距离，如果新信息的距离更小，则更新本地路由表
3. 如果3分钟没有收到相邻的路由信息，则把相邻路由设置为不可达(16跳)

RIP协议的过程

1. 路由器初始化路由信息(两个向量 D_i 和 S_i)
2. 对相邻路由器X发过来的信息，对信息的内容进行修改（下一跳地址设置为X，所有距离加1）

i. 检索本地路由，将信息中新的路由插入到路由表里面

	D_i	S_i
D	2	A



	D_i	S_i
A	4	C
B	2	C



	D_i	S_i
D	2	A
A	5	X
B	3	X

	D_i	S_i
A	5	X
B	3	X

RIP协议的过程

1. 路由器初始化路由信息(两个向量 D_i 和 S_i)
2. 对相邻路由器X发过来的信息，对信息的内容进行修改（下一跳地址设置为X，所有距离加1）

i. 检索本地路由，将信息中新的路由插入到路由表里面

ii. 检索本地路由，对于下一跳为X的，更新为修改后的信息

	D_i	S_i
A	5	X
B	3	X

	D_i	S_i
A	2	X
B	5	X



	D_i	S_i
A	4	C
B	2	C



	D_i	S_i
A	5	X
B	3	X

RIP协议的过程

1. 路由器初始化路由信息(两个向量 D_i 和 S_i)
2. 对相邻路由器X发过来的信息，对信息的内容进行修改（下一跳地址设置为X，所有距离加1）

i. 检索本地路由，将信息中新的路由插入到路由表里面

ii. 检索本地路由，对于下一跳为X的，更新为修改后的信息

iii. 检索本地路由，对比相同目的的距离，如果新信息的距离更小，则更新本地路由表

	D_i	S_i
A	5	X
B	3	X

	D_i	S_i
A	3	D
B	5	F



	D_i	S_i
A	4	C
B	2	C



	D_i	S_i
A	3	D
B	3	X

RIP协议的过程

1. 路由器初始化路由信息(两个向量 D_i 和 S_i)
2. 对相邻路由器X发过来的信息，对信息的内容进行修改（下一跳地址设置为X，所有距离加1）
 - i. 检索本地路由，将信息中新的路由插入到路由表里面
 - ii. 检索本地路由，对于下一跳为X的，更新为修改后的信息
 - iii. 检索本地路由，对比相同目的的距离，如果新信息的距离更小，则更新本地路由表
3. 如果3分钟没有收到相邻的路由信息，则把相邻路由设置为不可达(16跳)

RIP协议的过程

- ◆ 随便相信 “隔壁老王”
- ◆ “自己不思考” “视野不够”

故障信息传递慢



RIP协议的过程

- ◆ RIP协议：实现简单，开销很小
- ◆ RIP协议：限制了网络的规模
- ◆ RIP协议：“坏消息传的慢”，更新收敛时间过长

内部网关路由协议之RIP协议

- ◆ 距离矢量（DV）算法
- ◆ RIP协议的过程

慕课网
imooc.com

慕课网
imooc.com

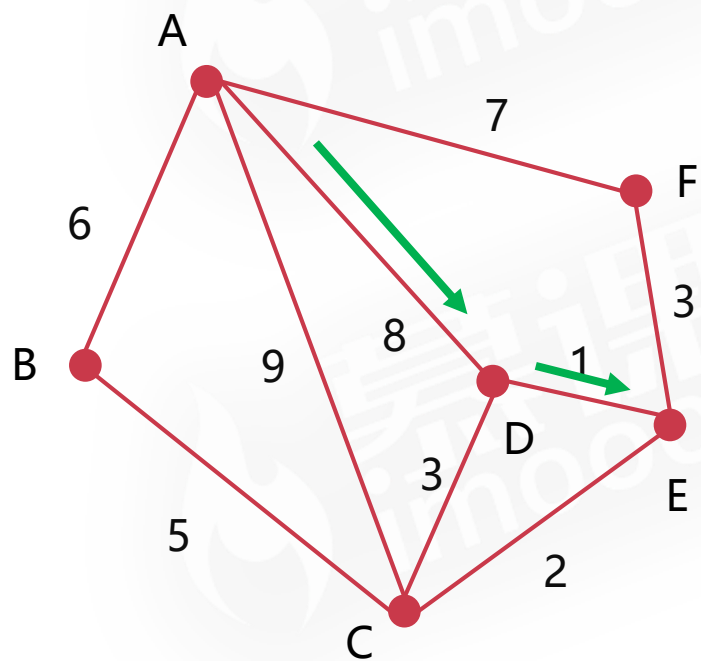
慕课网
imooc.com

慕课网
imooc.com

Dijkstra（迪杰斯特拉）算法

- ◆ Dijkstra算法是著名的图算法
- ◆ Dijkstra算法解决有权图从一个节点到其他节点的最短路径问题
- ◆ “以起始点为中心，向外层层扩展”

Dijkstra (迪杰斯特拉) 算法



A → B → C → E (13)

A → C → E (11)

A → D → C → E (13)

A → D → E (9)

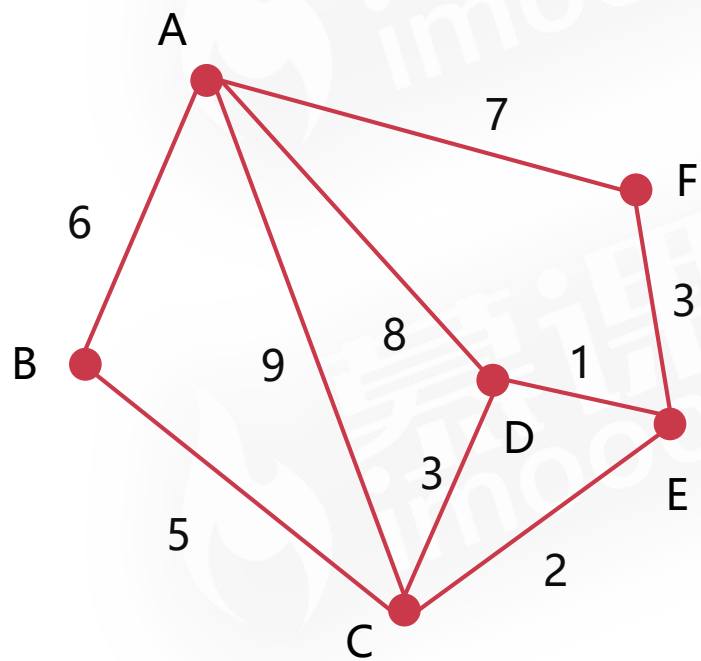
A → F → E (10)

最短路径问题

Dijkstra (迪杰斯特拉) 算法

1. 初始化两个集合(S , U) (S 为只有初始顶点点 A 的集合, U 为其他顶点集合)
2. 如果 U 不为空, 对 U 集合顶点进行距离的排序, 并取出距离 A 最近的一个顶点 D
 - i. 将顶点 D 的纳入 S 集合
 - ii. 更新通过顶点 D 到达 U 集合所有点的距离 (如果距离更小则更新, 否则不更新)
 - iii. 重复2步骤
3. 知道 U 集合为空, 算法完成

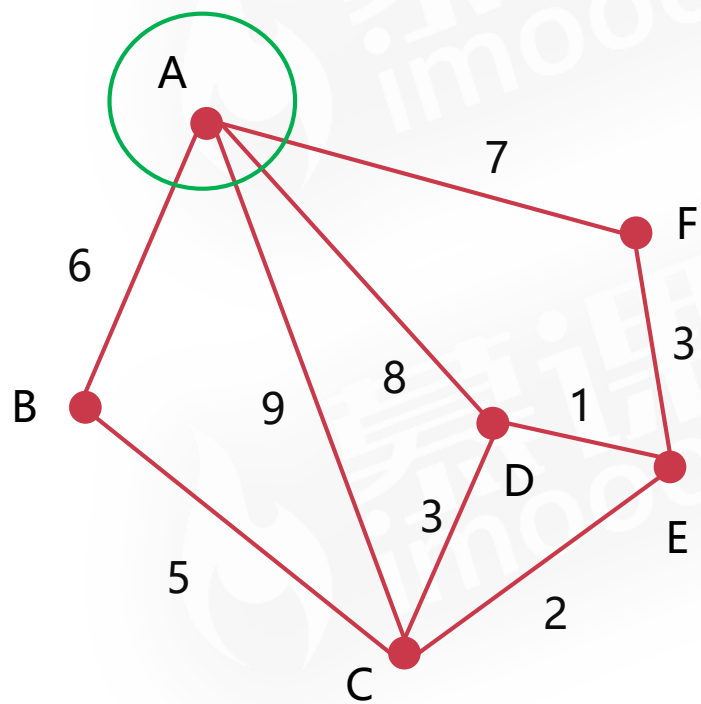
Dijkstra (迪杰斯特拉) 算法



A->(B、C、D、E、F)

	A
A	0
B	-
C	-
D	-
E	-
F	-

Dijkstra (迪杰斯特拉) 算法

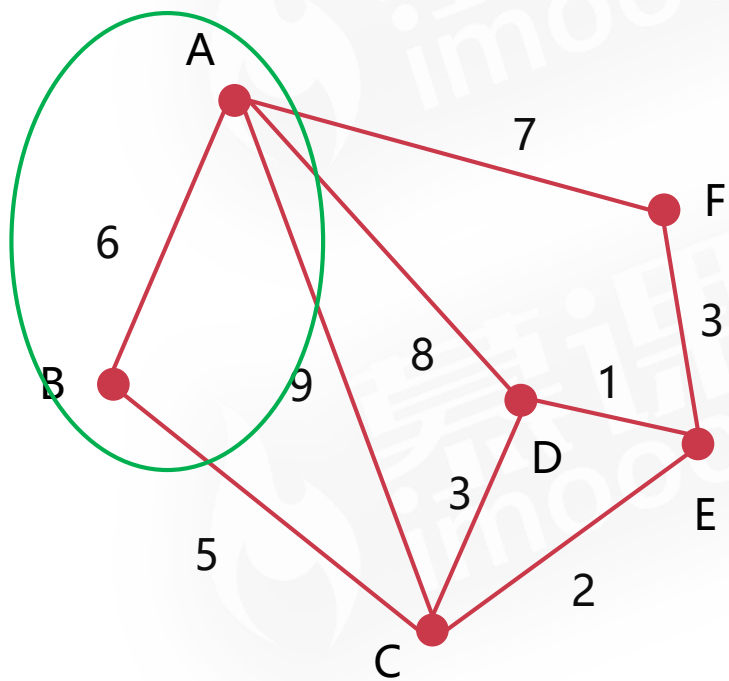


	A
A	0
B	6
C	9
D	8
E	-
F	7

$$S = (A \rightarrow A = 0)$$

$$U = \begin{pmatrix} A \rightarrow B = 6 \\ A \rightarrow C = 9 \\ A \rightarrow D = 8 \\ A \rightarrow E = - \\ A \rightarrow F = 7 \end{pmatrix}$$

Dijkstra (迪杰斯特拉) 算法



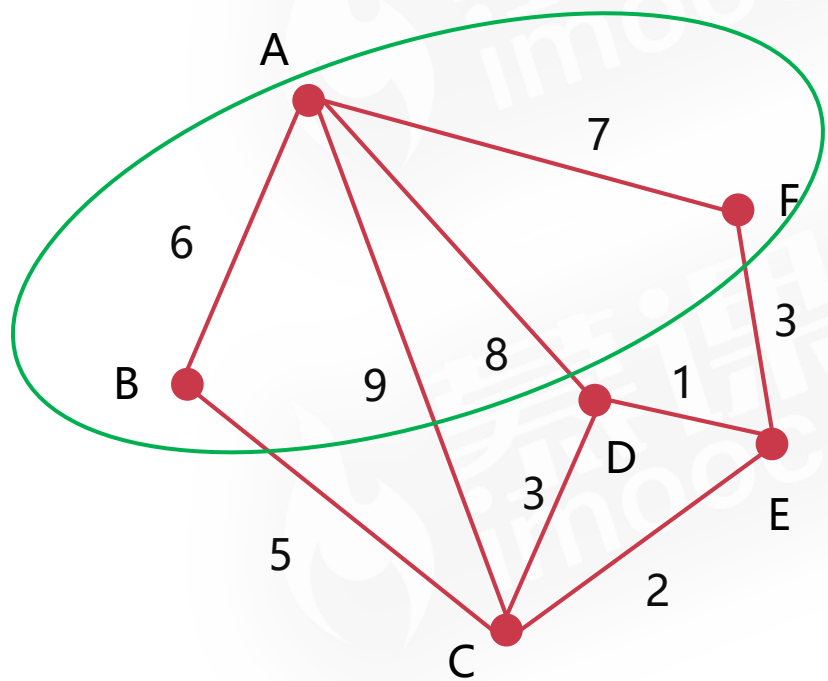
	A
A	0
B	6
C	9
D	8
E	-
F	7

$$S = \begin{pmatrix} A \rightarrow A = 0 \\ A \rightarrow B = 6 \end{pmatrix}$$

$$U = \begin{pmatrix} A \rightarrow C = 9 \\ A \rightarrow D = 8 \\ A \rightarrow E = - \\ A \rightarrow F = 7 \end{pmatrix}$$

A=>C: A->B->C(6+5=11)

Dijkstra (迪杰斯特拉) 算法



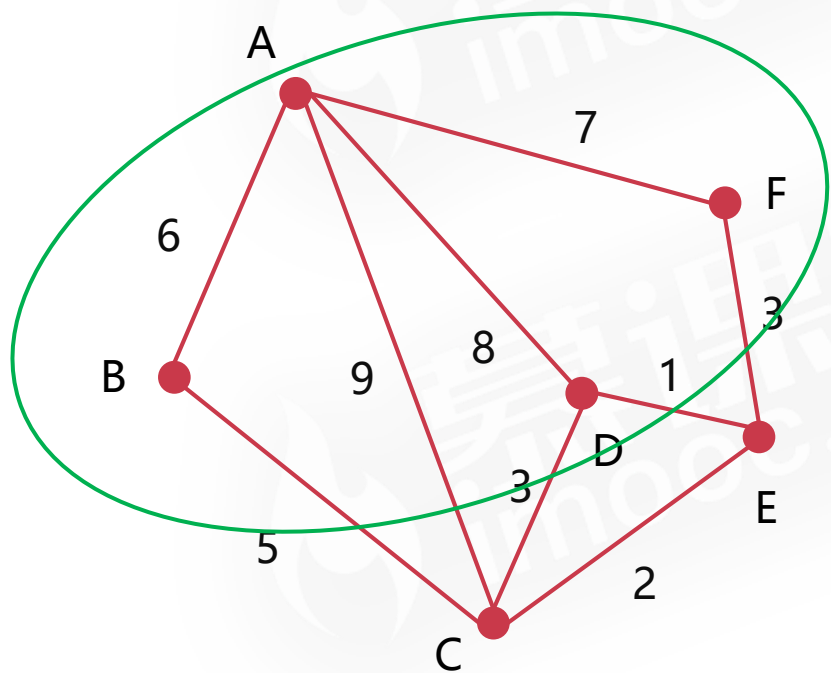
	A
A	0
B	6
C	9
D	8
E	10
F	7

$$S = \begin{pmatrix} A \rightarrow A = 0 \\ A \rightarrow B = 6 \\ A \rightarrow F = 7 \end{pmatrix}$$

$$U = \begin{pmatrix} A \rightarrow C = 9 \\ A \rightarrow D = 8 \\ A \rightarrow E = 10 \end{pmatrix}$$

$A \Rightarrow E: A \rightarrow F \rightarrow E (7 + 3 = 10)$

Dijkstra (迪杰斯特拉) 算法



	A
A	0
B	6
C	9
D	8
E	10
F	7

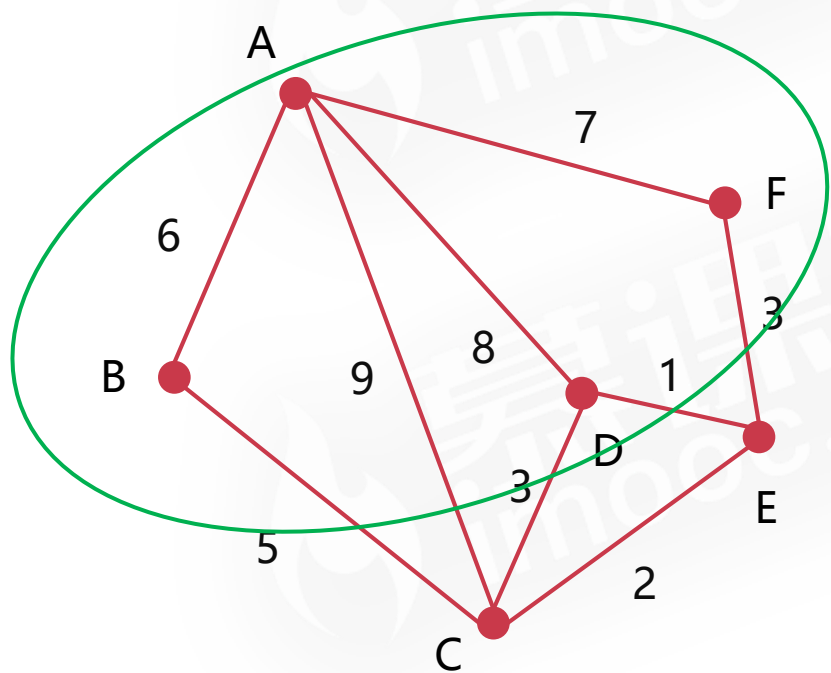
$$S = \begin{pmatrix} A \rightarrow A = 0 \\ A \rightarrow B = 6 \\ A \rightarrow F = 7 \\ A \rightarrow D = 8 \end{pmatrix}$$

$$U = \begin{pmatrix} A \rightarrow C = 9 \\ A \rightarrow E = 9 \end{pmatrix}$$

$A \rightarrow C: A \rightarrow D \rightarrow C (8 + 3 = 11)$

$A \rightarrow E: A \rightarrow D \rightarrow E (8 + 1 = 9)$

Dijkstra (迪杰斯特拉) 算法



	A
A	0
B	6
C	9
D	8
E	9
F	7

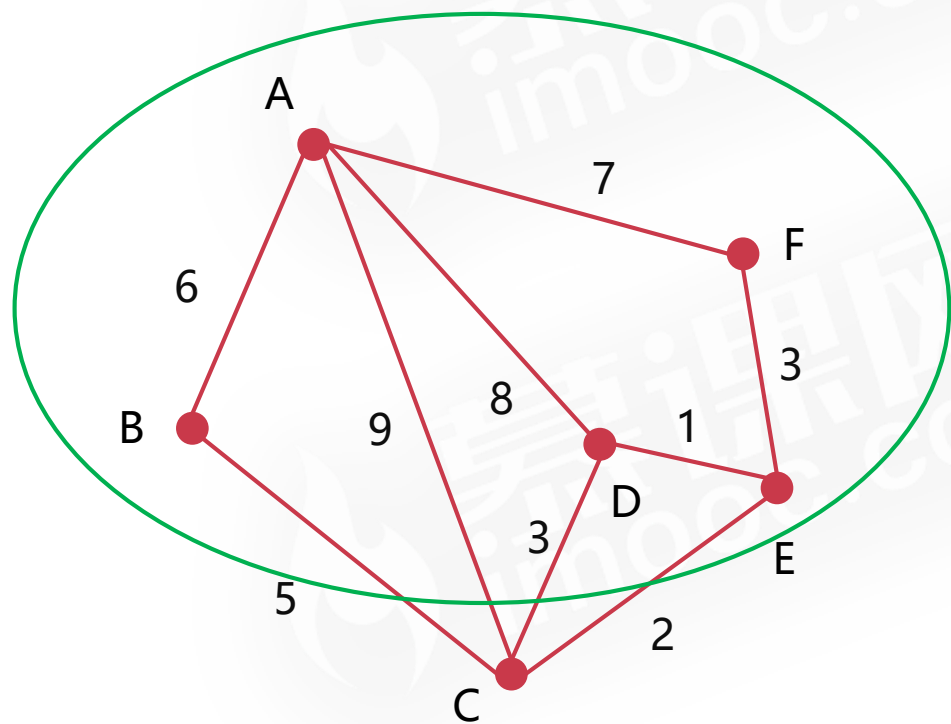
$$S = \begin{pmatrix} A \rightarrow A = 0 \\ A \rightarrow B = 6 \\ A \rightarrow F = 7 \\ A \rightarrow D = 8 \end{pmatrix}$$

$$U = \begin{pmatrix} A \rightarrow C = 9 \\ A \rightarrow E = 9 \end{pmatrix}$$

$A \rightarrow C: A \rightarrow D \rightarrow C (8 + 3 = 11)$

$A \rightarrow E: A \rightarrow D \rightarrow E (8 + 1 = 9)$

Dijkstra (迪杰斯特拉) 算法



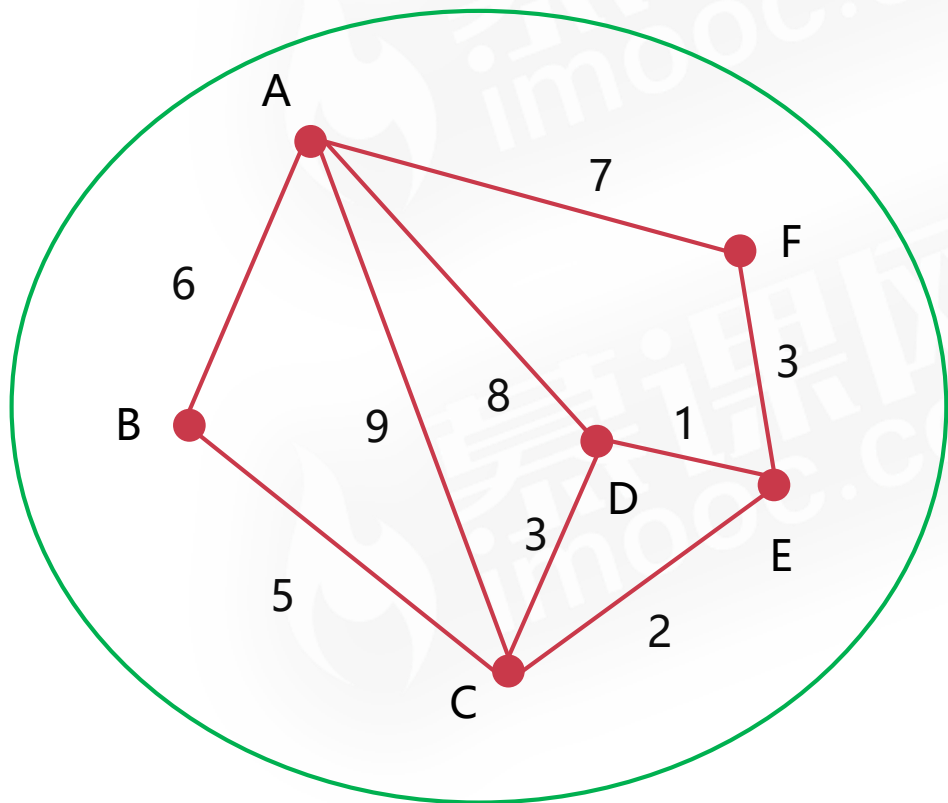
	A
A	0
B	6
C	9
D	8
E	9
F	7

$$S = \begin{pmatrix} A \rightarrow A = 0 \\ A \rightarrow B = 6 \\ A \rightarrow F = 7 \\ A \rightarrow D = 8 \\ A \rightarrow E = 9 \end{pmatrix}$$

$$U = (A \rightarrow C = 9)$$

$A \Rightarrow C: A \rightarrow E \rightarrow C (9 + 2 = 11)$

Dijkstra (迪杰斯特拉) 算法



	A
A	0
B	6
C	9
D	8
E	9
F	7

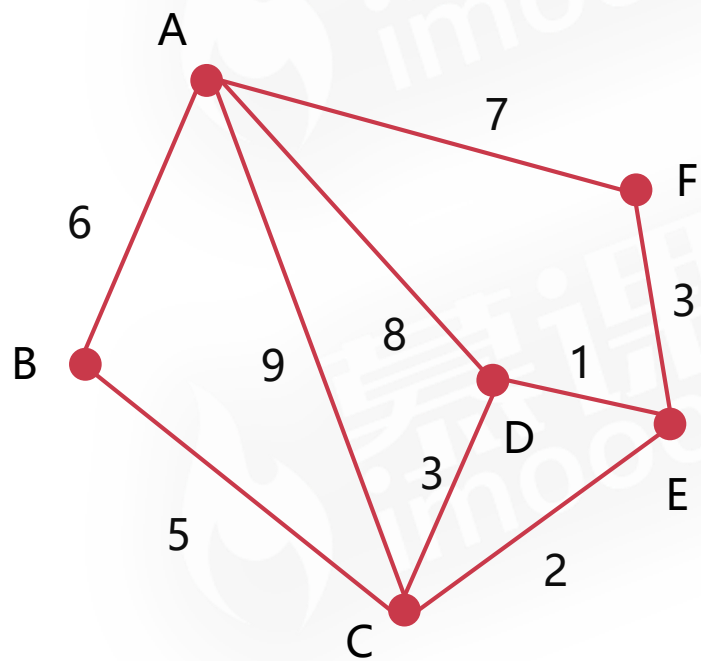
$$S = \begin{pmatrix} A \rightarrow A = 0 \\ A \rightarrow B = 6 \\ A \rightarrow F = 7 \\ A \rightarrow D = 8 \\ A \rightarrow E = 9 \\ A \rightarrow C = 9 \end{pmatrix}$$

$$U = ()$$

Dijkstra (迪杰斯特拉) 算法

1. 初始化两个集合(S , U) (S 为只有初始顶点点 A 的集合, U 为其他顶点集合)
2. 如果 U 不为空, 对 U 集合顶点进行距离的排序, 并取出距离 A 最近的一个顶点 D
 - i. 将顶点 D 的纳入 S 集合
 - ii. 更新通过顶点 D 到达 U 集合所有点的距离 (如果距离更小则更新, 否则不更新)
 - iii. 重复2步骤
3. 知道 U 集合为空, 算法完成

Dijkstra (迪杰斯特拉) 算法



A->(B、C、D、E、F)

	F
A	-
B	-
C	-
D	-
E	-
F	0

	E
A	-
B	-
C	-
D	-
E	0
F	-

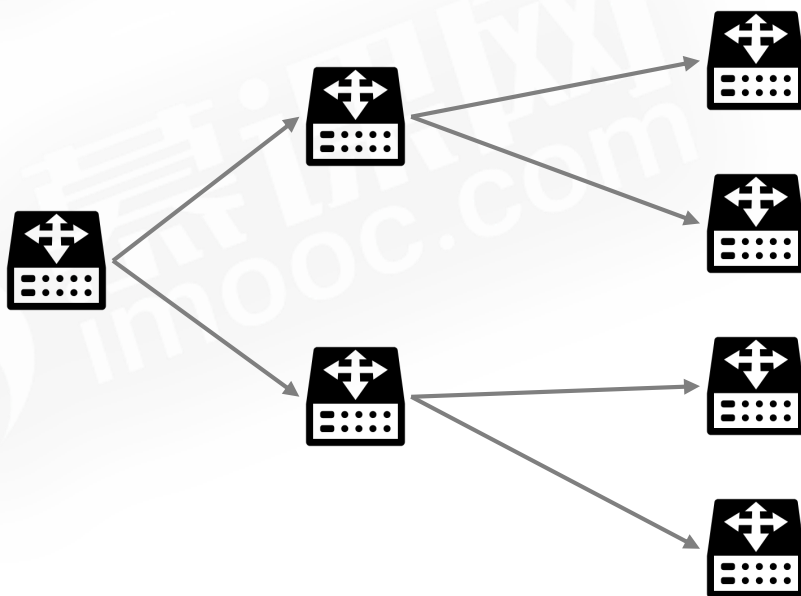


内部网关路由协议之OSPF协议

- ◆ 链路状态（LS）协议
- ◆ OSPF协议的过程

链路状态 (LS) 协议

- ◆ 向所有的路由器发送消息



一传十、十传百

只和相邻的路由器交换信息

链路状态 (LS) 协议

- ◆ 消息描述该路由器与相邻路由器的链路状态

距离

时延

带宽

...

网络管理人员决定的



“代价”

只使用跳数

链路状态 (LS) 协议

- ◆ 只有链路状态发生变化时，才发送更新信息

每隔30s交换路由信息

链路状态 (LS) 协议

- ◆ 向所有的路由器发送消息
- ◆ 消息描述该路由器与相邻路由器的链路状态
- ◆ 只有链路状态发生变化时，才发送更新信息

随便相信 “隔壁老王”

“视野不够”

内部网关路由协议之OSPF协议

- ◆ 链路状态（LS）协议
- ◆ OSPF协议的过程

OSPF协议的过程

- ◆ OSPF(Open Shortest Path First: 开放最短路径优先)
- ◆ OSPF协议的核心是Dijkstra算法

OSPF协议的过程

- ◆ 向所有的路由器发送消息
- ◆ 消息描述该路由器与相邻路由器的链路状态
- ◆ 只有链路状态发生变化时，才发送更新信息

OSPF协议的过程

◆ 向所有的路由器发送消息

获得网络中的所有信息



“网络的完整拓扑”

- ◆ 也称为“链路状态数据库”
- ◆ “链路状态数据库”是全网一致的

Dijkstra算法

OSPF协议的过程

- ◆ 消息描述该路由器与相邻路由器的链路状态

距离

时延

带宽

...

OSPF协议更加客观、更加先进

OSPF协议的过程

- ◆ 只有链路状态发生变化时，才发送更新信息

减少了数据的交换，更快收敛

OSPF协议的过程

五种消息类型

- ◆ 问候消息 (Hello)
- ◆ 链路状态数据库描述信息
- ◆ 链路状态请求信息

OSPF协议的过程

五种消息类型

- ◆ 链路状态更新信息
- ◆ 链路状态确认信息

OSPF协议的过程



OSPF协议的过程

RIP协议	OSPF协议
从邻居看网络	整个网络的拓扑
在路由器之间累加距离	Dijkstra算法计算最短路径
频繁、周期更新，收敛很慢	状态变化更新，收敛很快
路由间拷贝路由信息	路由间传递链路状态，自行计算路径

内部网关路由协议之OSPF协议

- ◆ 链路状态（LS）协议
- ◆ OSPF协议的过程

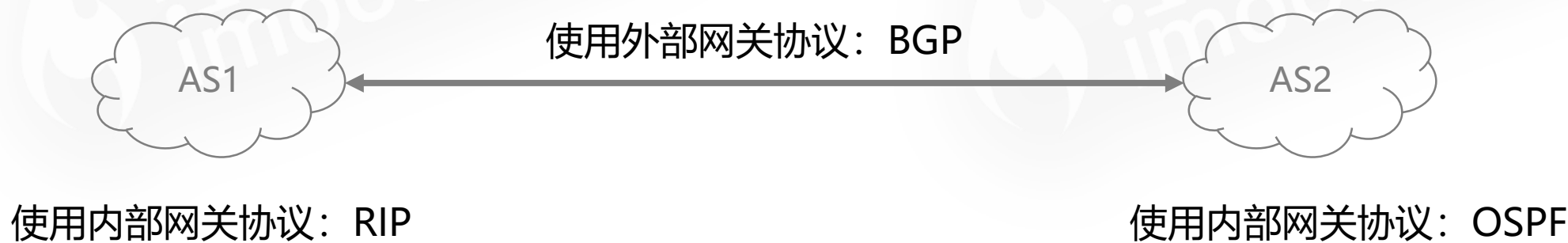


外部网关路由协议之BGP协议

- ◆ BGP(Border Gateway Protocol: 边际网关协议)
- ◆ BGP协议是运行在AS之间的一种协议

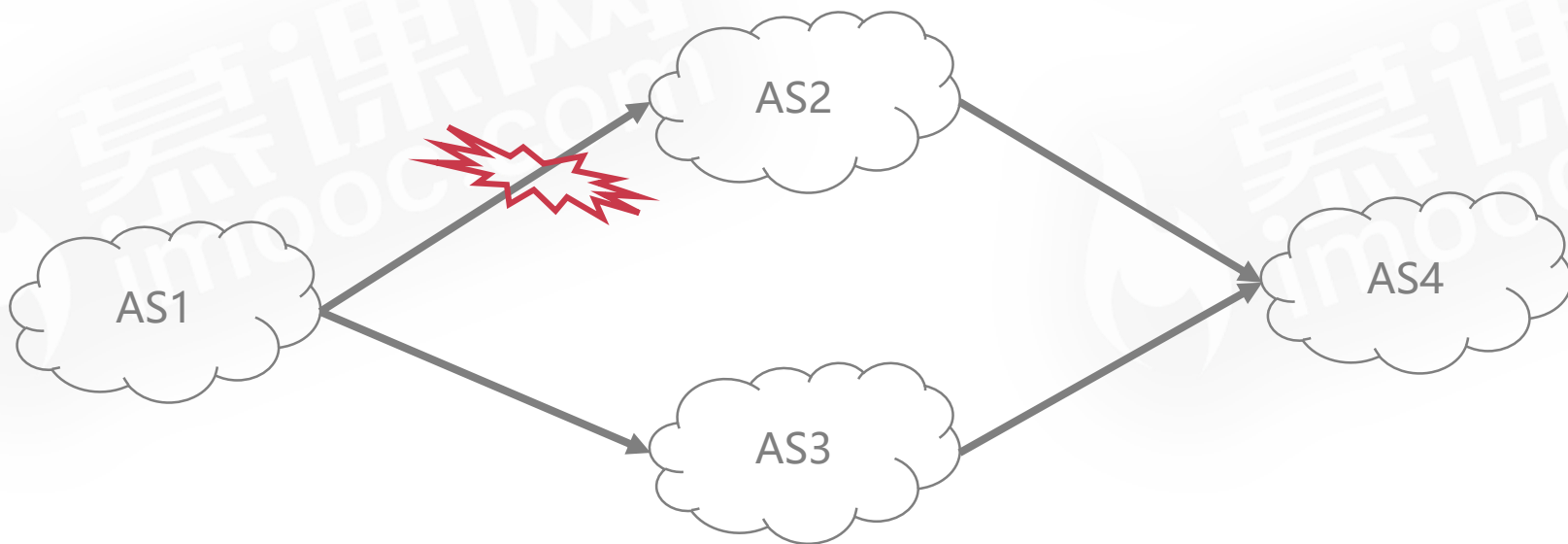
外部网关路由协议之BGP协议

- ◆ 互联网的规模很大
- ◆ AS内部使用不同的路由协议



外部网关路由协议之BGP协议

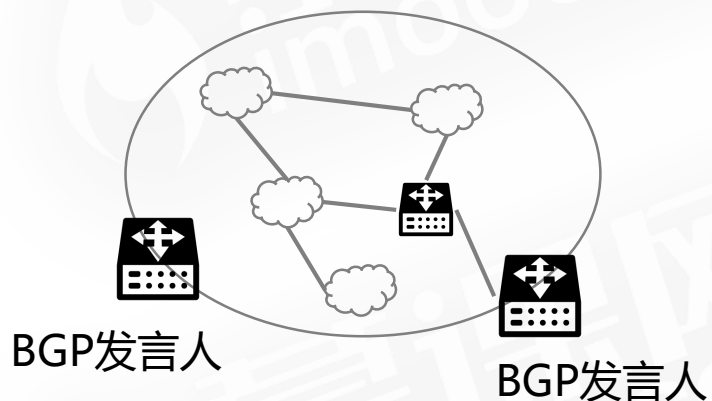
- ◆ AS之间需要考虑除网络特性以外的一些因素（政治、安全...）



外部网关路由协议之BGP协议

- ◆ BGP(Border Gateway Protocol, 边界网关协议)
- ◆ BGP协议能够找到一条到达目的比较好的路由

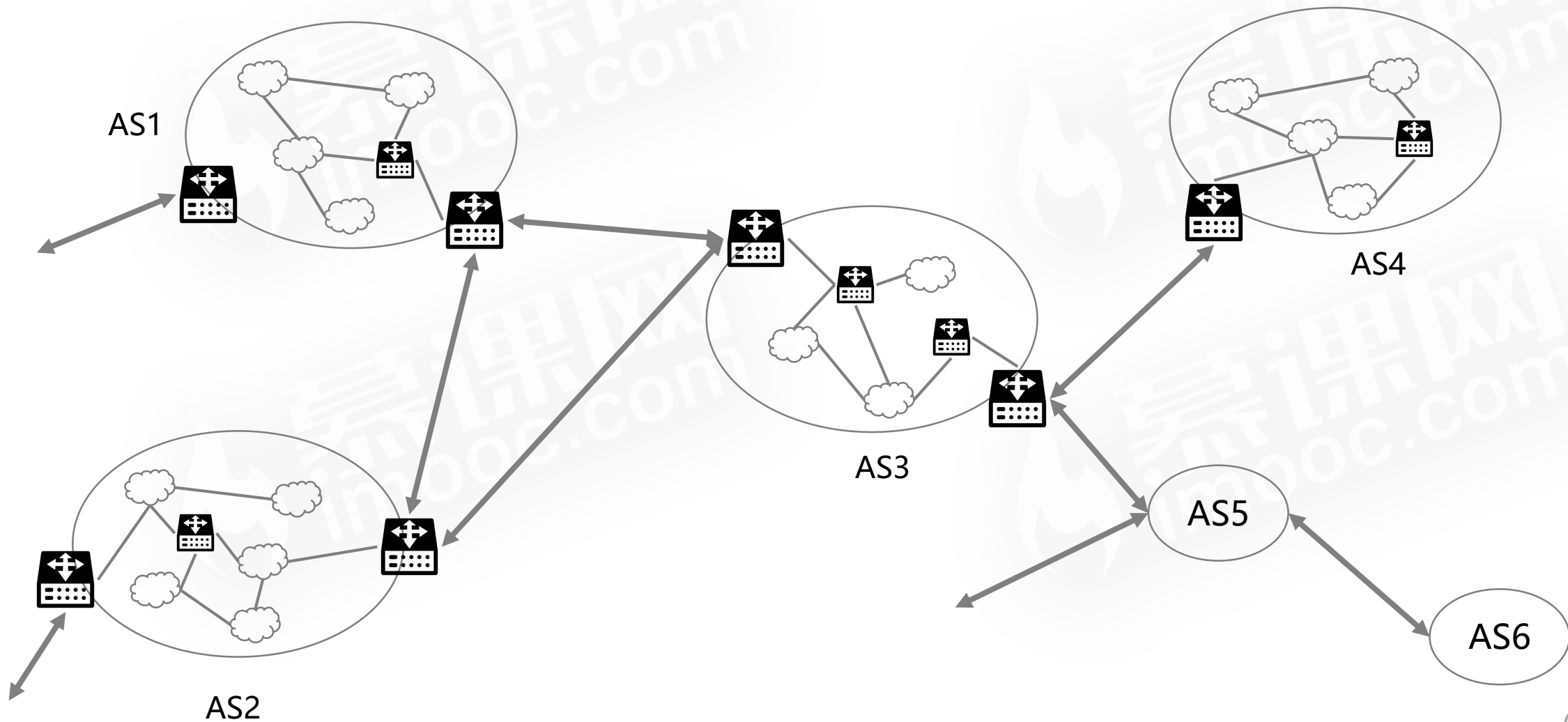
外部网关路由协议之BGP协议



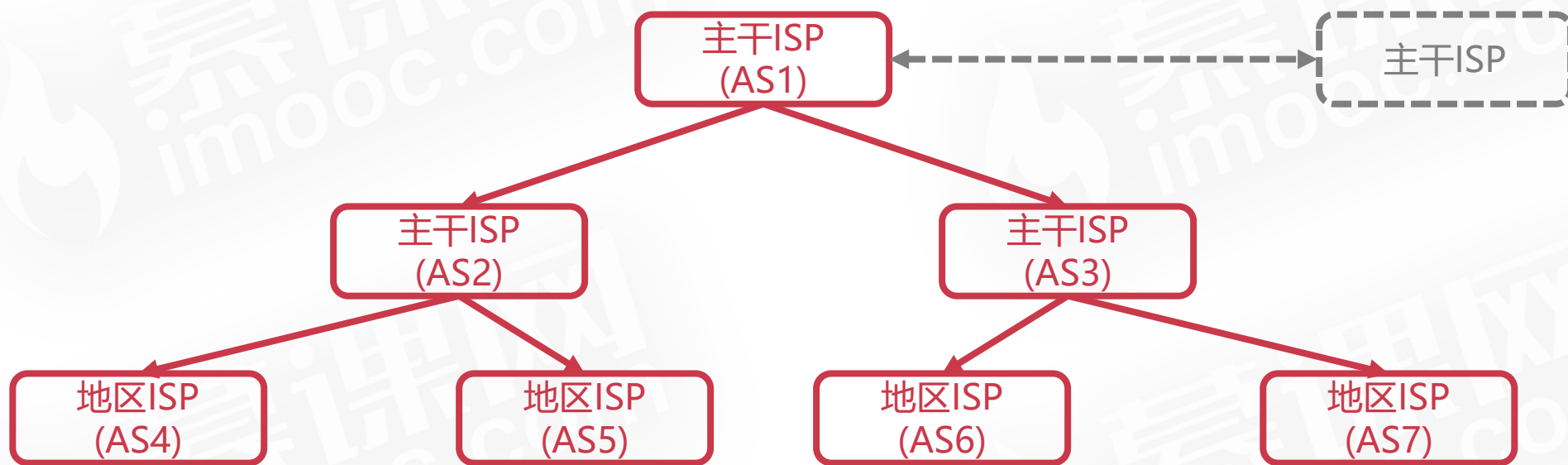
- ◆ BGP并不关心内部网络拓扑
- ◆ AS之间通过BGP发言人交流信息
- ◆ BGP Speaker可以人为配置策略

BGP发言人(speaker)

外部网关路由协议之BGP协议

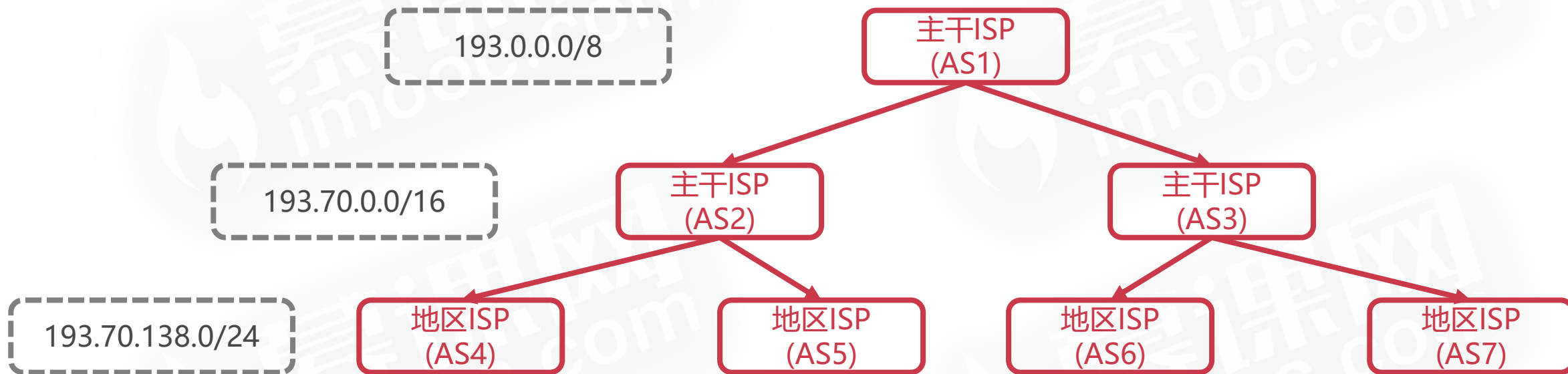


外部网关路由协议之BGP协议



- ◆ AS1: 通过我可以到达AS2和AS3, 但是你需要持有“通行证”
- ◆ AS2: 通过我可以到达AS4和AS5

外部网关路由协议之BGP协议



- ◆ AS1: 通过我可以到达AS2和AS3, 但是你需要持有“通信证”
- ◆ AS2: 通过我可以到达AS4和AS5

外部网关路由协议之BGP协议

- ◆ BGP协议能够找到一条到达目的比较好的路由
- ◆ AS之间通过BGP发言人来进行路由信息的交换

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com