现代计算机网络

李伟明 lwm@hust.edu.cn

1.4 TCP/IP网络技术

报文交换网络无法组成一个大范围网络

1. 报文交换网络默认在一个广播域（VLAN是手工分割广播域）
2. Forwarding table无法扩展
3. 以太网地址是厂商+生产序号，无法更改，也无法聚合

需要更好的方案组成大范围的网络

1.4 TCP/IP网络技术

1.4.1 命名与定位（Naming & Locating）

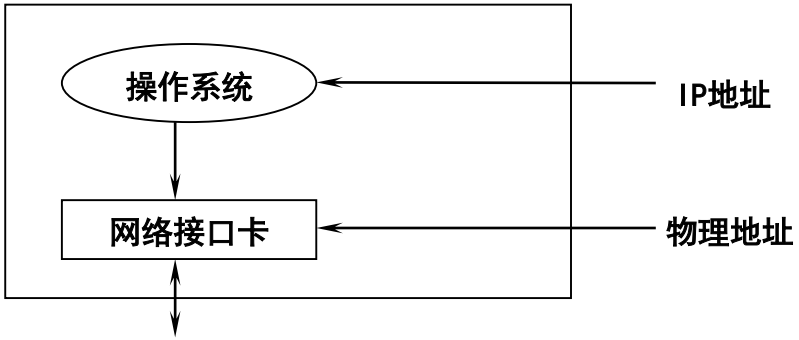
1.4.2 IP互连与分组交换

1.4.3 路由与寻址（Routing & Addressing）

1.4.4 传输层技术

1.4.1 因特网的三地址

- 用户识别地址、网络地址、物理地址
  - 用户识别地址：公司/机关/团体/个人注册的因特网可访问的ID：域名
  - 网络地址：同一体系结构中的可访问的计算机ID: IP地址
  - 物理地址：物理媒体可访问的计算机某端口的唯一ID: MAC地址



IP地址与物理地址的关系

IP地址

- IP 地址是不是只有点分十进制这种写法呢？
- 如果在URL中包含IP地址，有非常多的表示方法
- CuteIP（https://github.com/D4Vinci/Cuteit）

```
ip > 192.168.1.1

[0] http://0xc0.0xa8.0x1.0x1
[1] http://0xc0.0xa8.0x1.1
[2] http://0xc0.0xa8.1.1
[3] http://0xc0.168.1.1
[4] http://0x00000000c0.0x000000a8.0x00001.0x001
[5] http://0xc0a80101
[6] http://howsecureismypassword.net@192.168.1.1
[7] http://google.com@3232235777
[8] http://facebook.com@0xc0a80101
[9] http://%31%39%32%2E%31%36%38%2E%31%2E%31
[10] https://www.google.com@search@%31%39%32%2E%31%36%38%2E%31%2E%31
[11] http://anywebsite@0300.0250.0001.0001
[12] http://0300.0250.0001.0001
[13] http://3232235777
```

IP地址

- 因特网是全球单一可寻址的抽象网络，必须连接全球所有计算机或其它设备，必须给每个计算机或设备（路由器等）一个全球唯一ID
- IPv4 = 2^{32}个地址，A/D/C/E类
- IP 地址结构 = Net号 + Host号
- 位置与身份合一

1.4.1 （网络实体）命名与定位

- 直接定位与解析定位

- IP地址引发问题（简单,但重载）
  - 不便记忆：产生域名→DNS协议
  - 局域网早先出现：导致 MAC到IP变换→ARP协议
  - IP地址分配不均、不够：NAT，三个段公用私网地址：17621775个IP
    - 10/8:1个A
    - 172.16.—172.31/16:15个B
    - 192.168.—192.168/16:1个B
  - IPv4地址已经耗尽：2011年ICANN Assigns Its Last IPv4 Addresses。后来又回收了一些地址再次分配。

域名解析服务-DNS

所有域可以划分为域（domain），域构成树状层级结构

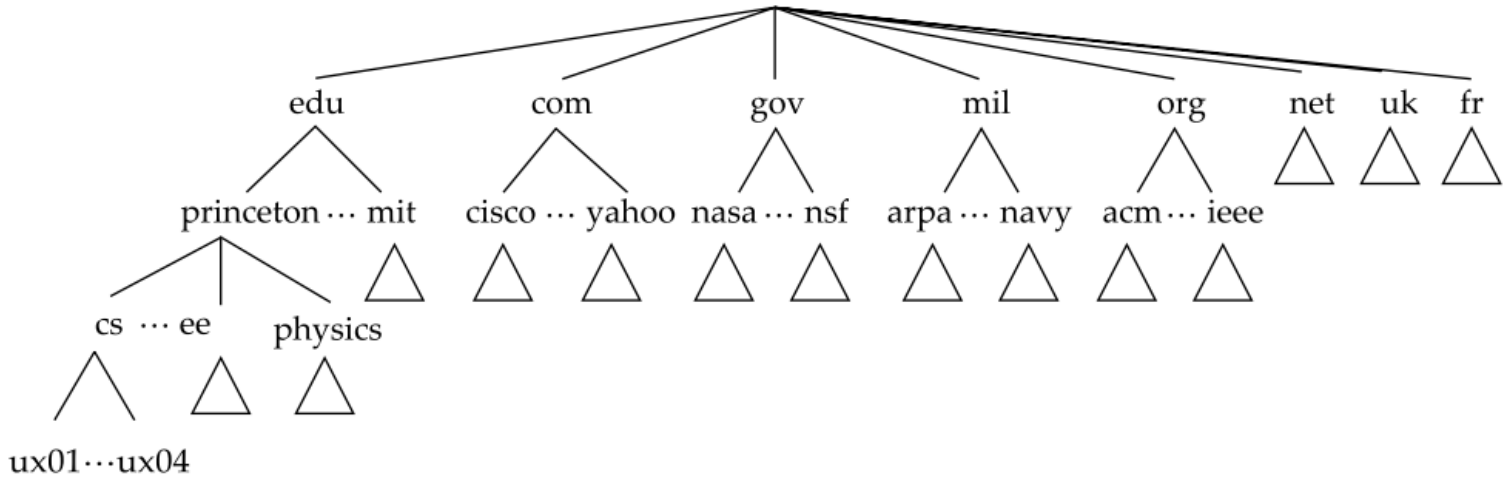


Figure 9.5 Example of a domain hierarchy.

域名解析服务-DNS

但是这种层级结构是抽象的，管理上是分成Zone进行管理：

- Zone管理某个域和特定子域（a domain or sub-domain）的DNS记录
- 例如顶级域名在一个Zone中,归ICANN管理
- Princeton和下面的physics在一个Zone
- CS在一个Zone

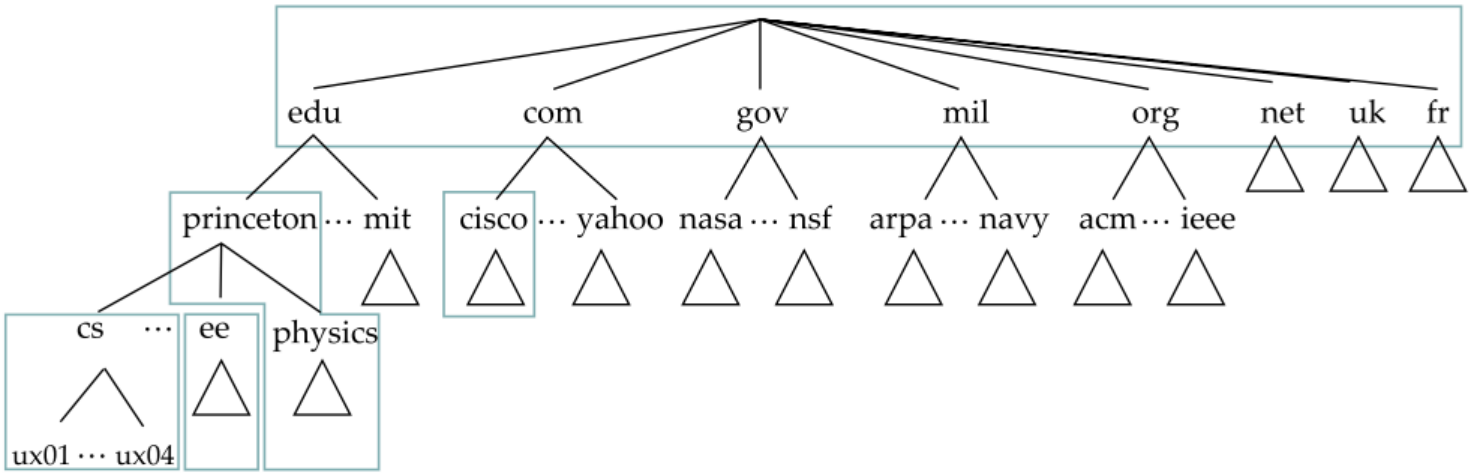


Figure 9.6 Domain hierarchy partitioned into zones.

域名解析服务-DNS

一个Zone物理需要一个DNS服务器为其提供服务（为了备份和负载均衡，也可能多台但是内容相同），提供一个Zone内所有域名的解析。实际上一个DNS服务器也可以管理多个域

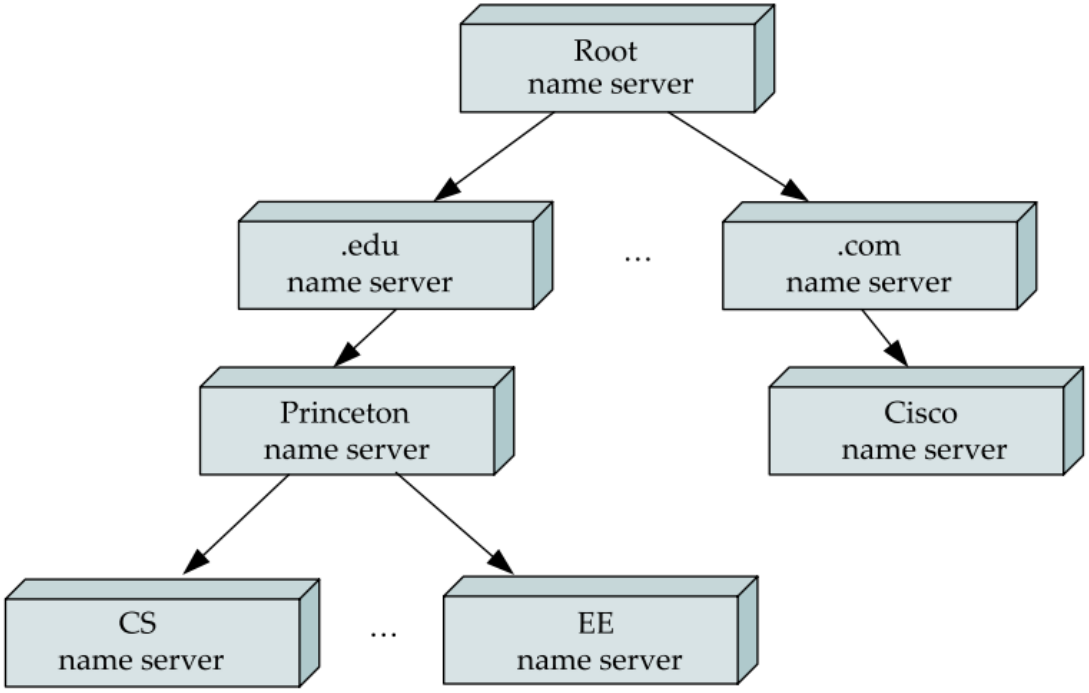


Figure 9.7 Hierarchy of name servers.

域名解析服务-DNS

每个Zone中的域名相关信息可以看作资源记录的集合，资源记录：（Name, Value, Type, Class, TTL）

Type域典型值：

- A: 域名和IPv4对应记录
- AAAA：域名和IPv6地址对应记录
- NS: 一台服务器的域名，这台服务器作为某个域的DNS服务器
- CNAME: 一个域名的别名（canonical name）
- MX: 域内邮件服务器域名

域名解析服务-DNS

下面的Resource Record例子：$ORIGIN defines a base name；IN表示Internet；TTL为2天；10表示优先度

```
; zone fragment for example.com
$TTL 2d ; zone default = 2 days or 172800 seconds
$ORIGIN example.com.
....
            IN      MX   10   mail.example.com.
mail        IN      CNAME    server1
server1     IN      A        192.168.0.3
```

域名解析服务-DNS

这样可以读懂下面这个DNS查询结果：

- www.baidu.com 是 www.a.shifen.com 的别名
- www.a.shifen.com 有两个A记录分别对应两个IP地址
- Non-authoritative answer表示不是来自保存资源记录的服务器直接结果、还可能是缓存的

```
@DESKTOP-HE011GO:~$ nslookup www.baidu.com
Server:         202.114.0.131
Address:        202.114.0.131#53

Non-authoritative answer:
www.baidu.com   canonical name = www.a.shifen.com.
Name:   www.a.shifen.com
Address: 182.61.200.6
Name:   www.a.shifen.com
Address: 182.61.200.7
```

域名解析服务-DNS

一定要知道这个解析是由哪个DNS服务器完成的怎么办？
Start of Authority Resource Record得到一个Zone的SOA记录

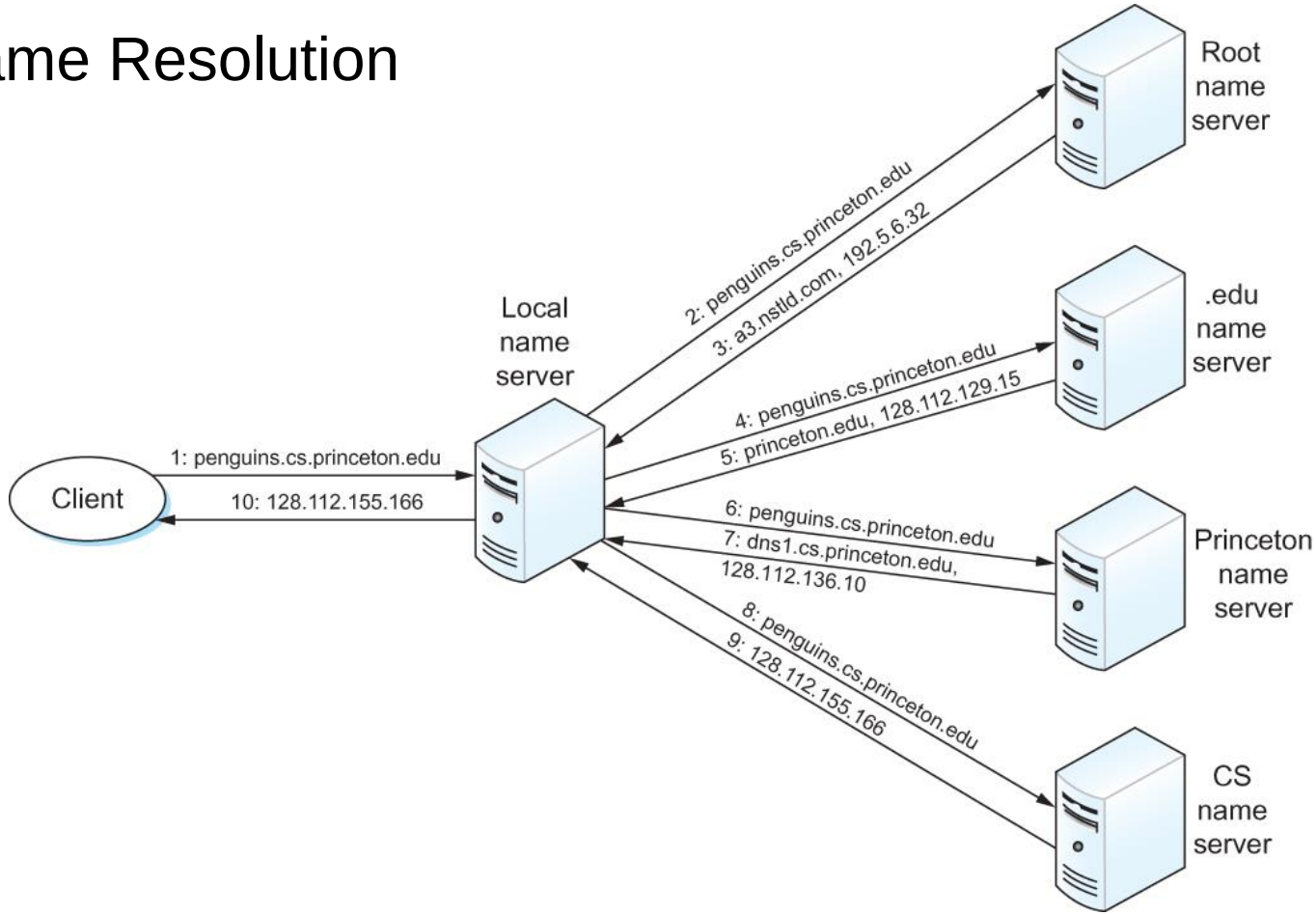
```
[redacted]@DESKTOP-HE011GO:~$ nslookup -type=soa www.baidu.com
Server:         202.114.0.131
Address:        202.114.0.131#53

Non-authoritative answer:
www.baidu.com   canonical name = www.a.shifen.com.

Authoritative answers can be found from:
a.shifen.com
        origin = ns1.a.shifen.com
        mail addr = baidu_dns_master.baidu.com
        serial = 1911140006
        refresh = 5
        retry = 5
        expire = 2592000
        minimum = 3600
```

域名解析服务-DNS

- Name Resolution



Name resolution in practice, where the numbers 1 – 10 show the sequence of steps in the process.

域名解析服务-DNS

- Root Server在资源记录中的例子
  （‘root’， a.root-servers.net, NS, IN ）
  （a.root-servers.net, 198.41.0.4, A, IN）

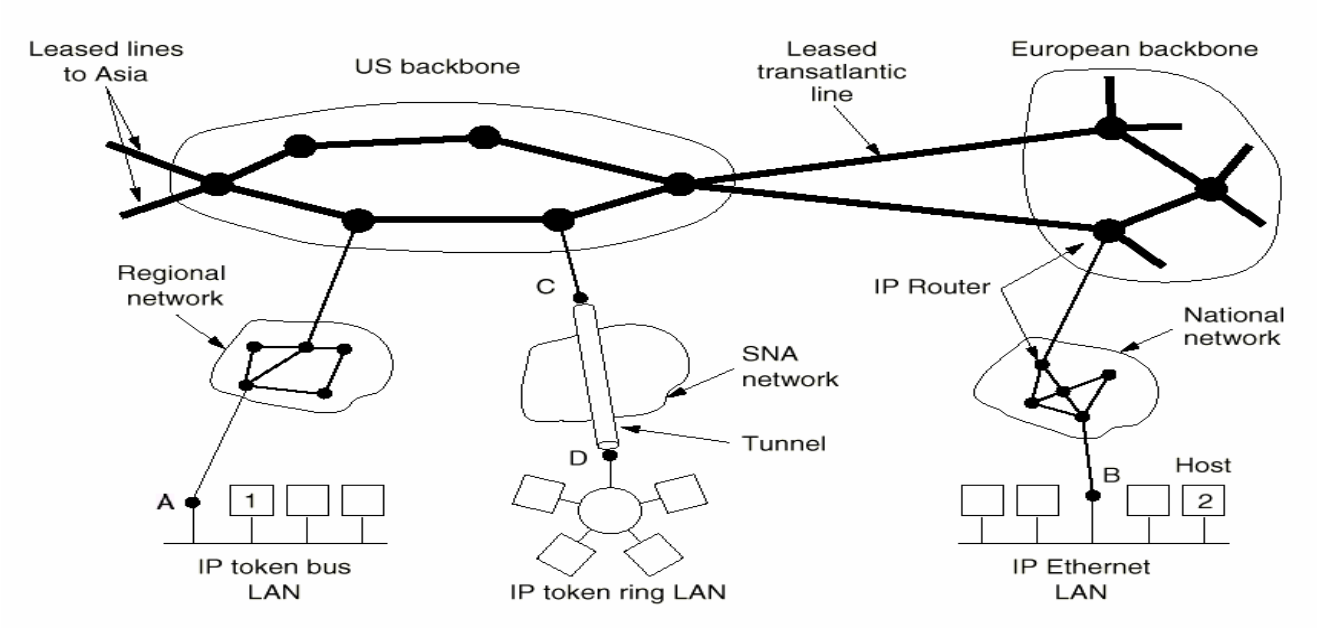
- 域传递Zone transfer，即Slave Server从Main Server拷贝数据,Slave不能修改资源记录，所以…….

1.4.2 IP互连与分组交换

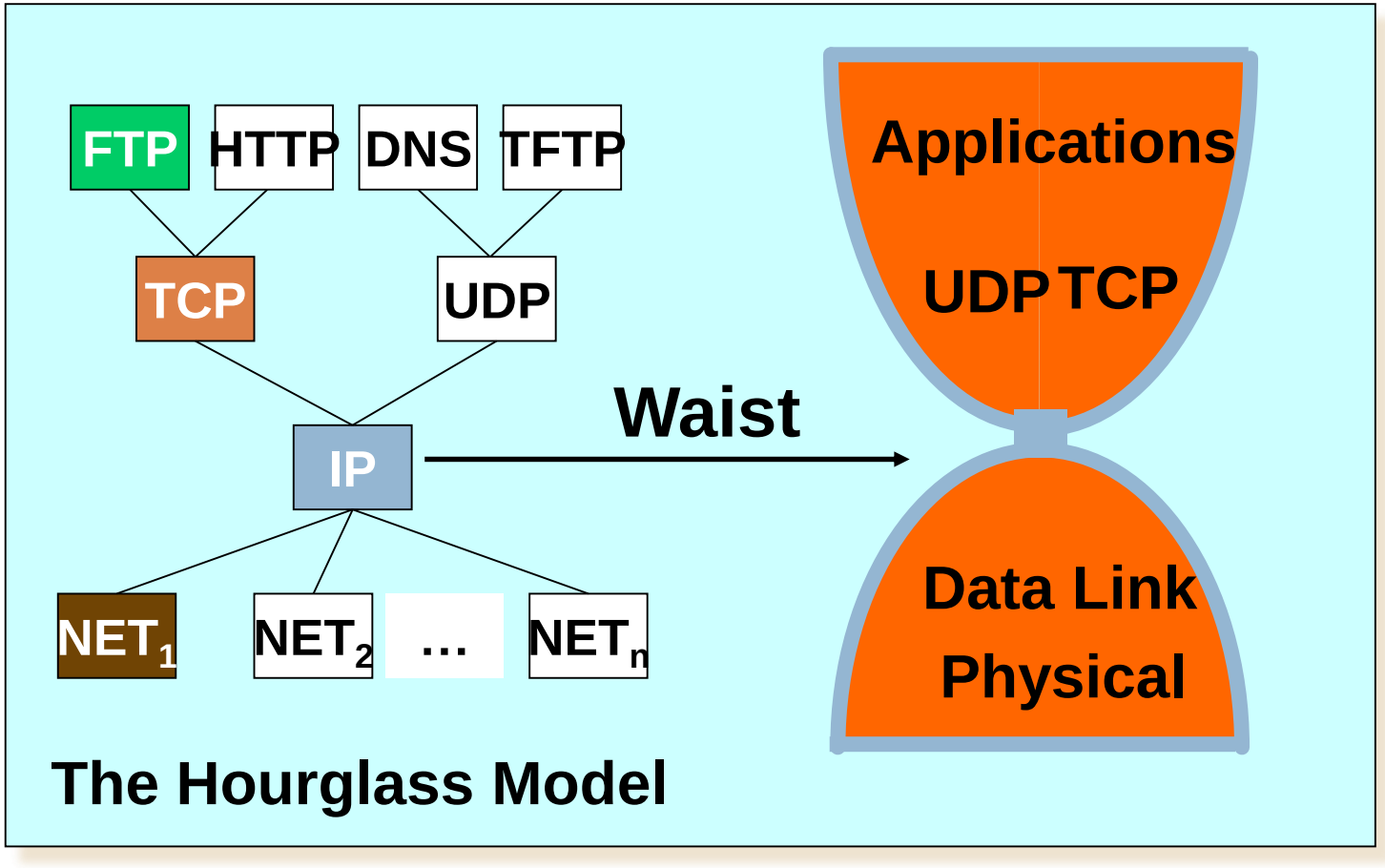
- 多个独立网络怎么互连？（历史：包容策略战胜统一策略）
- 多个网络互连的2个主要问题：
  - 异构：
    - 以太网、令牌环、点-点链路及各种交换网络，每个都有自己的地址模式、媒体访问协议及服务模式等
    - 传输介质不同/网络拓扑结构不同
    - 介质访问方式不同/网络编址方式不同
    - 分组长度/有连接/无连接服务的区别
    - 传输控制方式不同
    - 各层协议的功能定义、格式、接口与调用方式不同
  - 可扩展：
    - 网络不断扩大，路由规模指数增（100万—22.7亿个节点）

Internet网络层协议

- 在网络层，Internet可以看成是自治系统的集合，是由网络组成的网络。
- 网络之间互连的纽带是internet protocol协议。



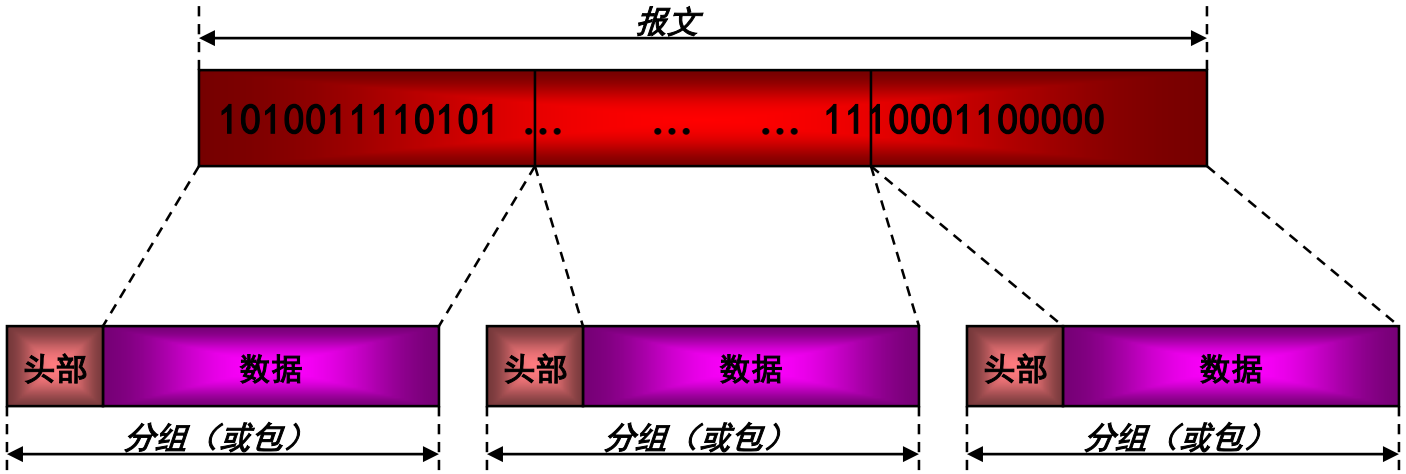
The “Narrow Waist” of IP



细腰促进了交互

IP采用分组交换

- 转发单元包含分组的目的地址
- 没有提前建立状态
- 作为基本构建单元，不提供可靠性
  - IP是best-effort，不可靠，上层协议需要提供可靠机制
- 几乎暗指统计多路复用



IP报文

IP报文格式，非常简单（Flags三个bit用了两个：2(MF)、3(DF)）：

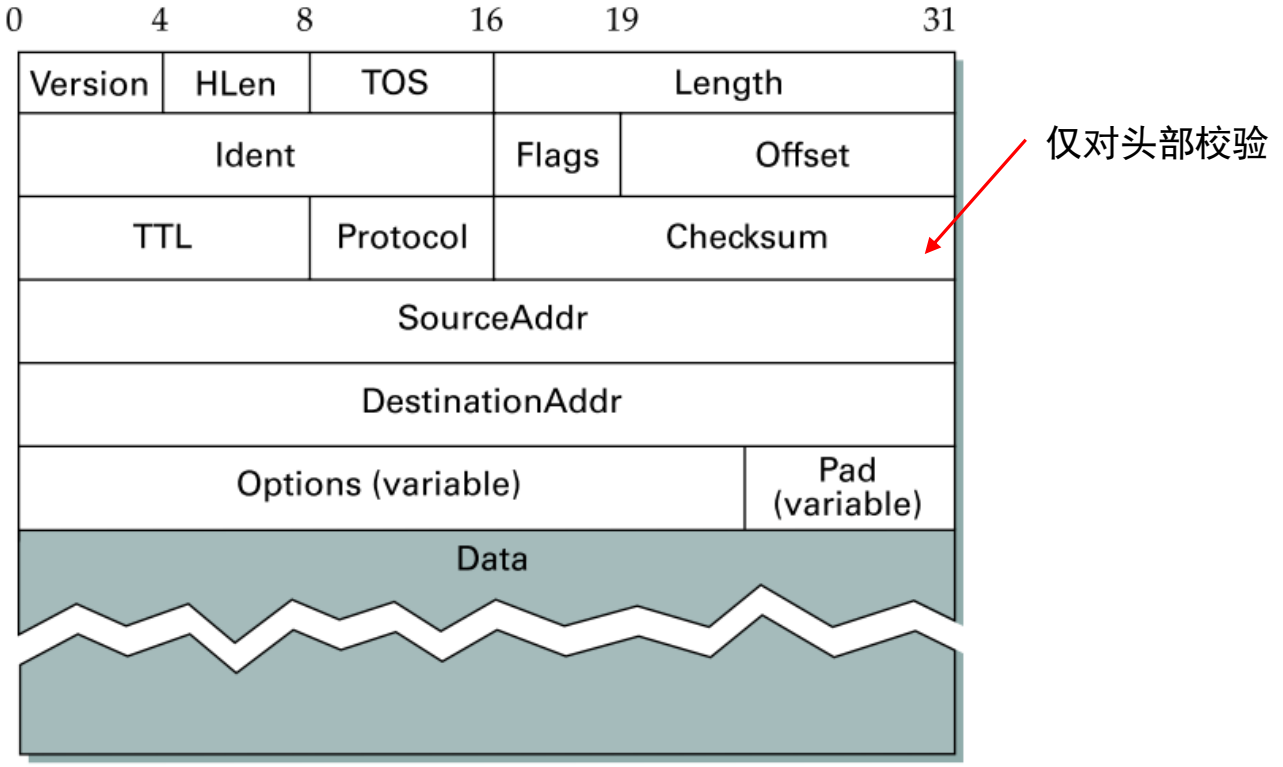
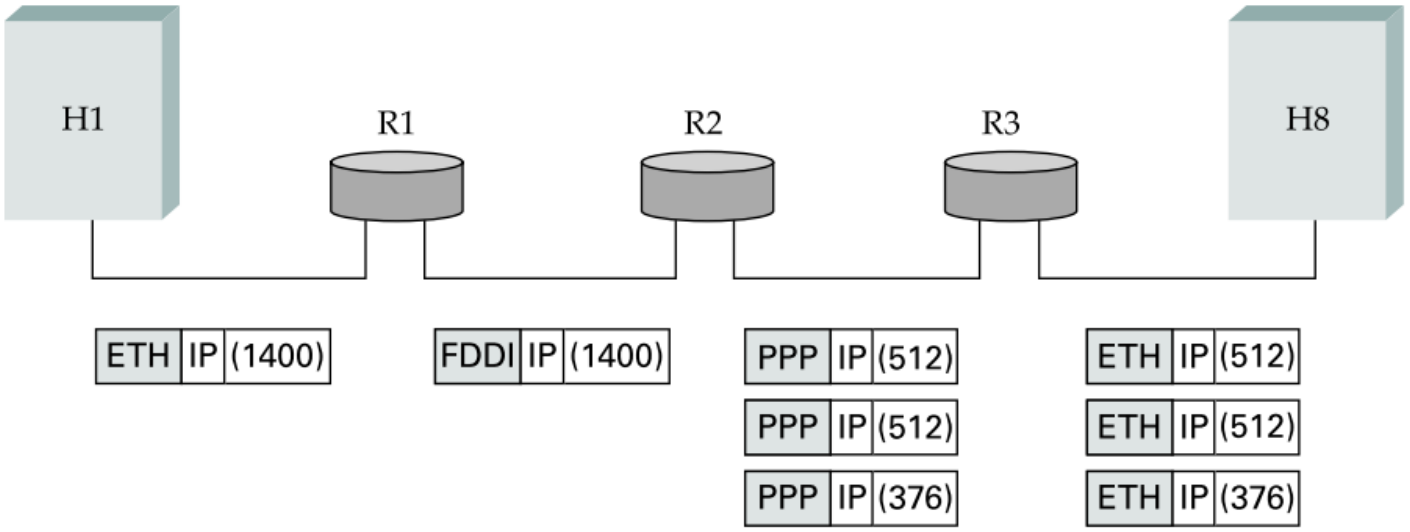


Figure 4.3 IPv4 packet header.

IP报文

比较复杂的是分片机制：

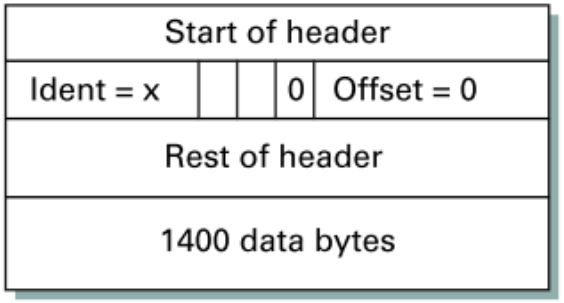
- 每个网络都有一个maximum transmission unit (MTU)
- FDDI（4500Byte）的报文到Ethernet（1500Byte）就要分片



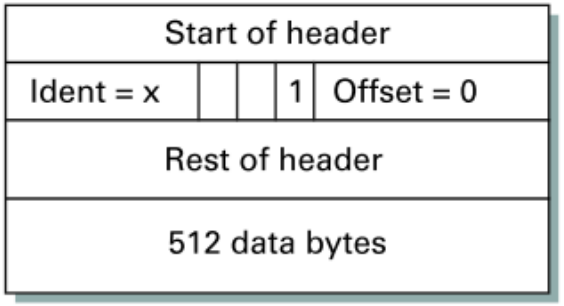
IP报文

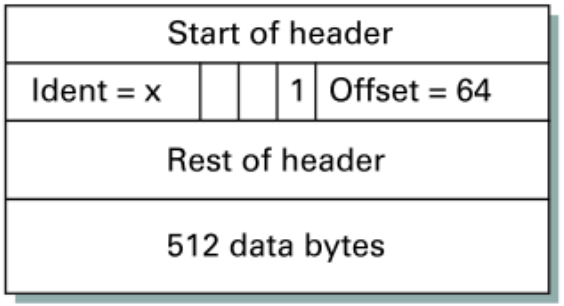
- 分片前后（a，b）的IP头部信息
- 三个域ID、MF、Offset
- 注意分片只有到达目的节点才重组（End to End）

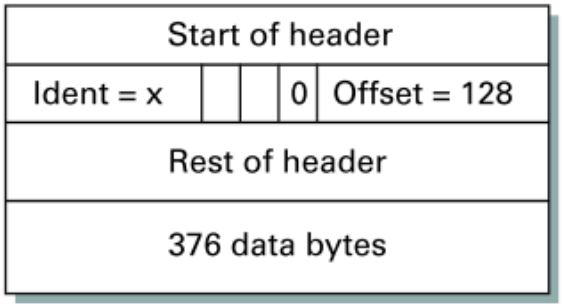
(a)



(b)







IP报文

但是教课书上对于分片机制的结论：

- IP fragmentation is generally considered to avoid.
- Hosts are now strongly encouraged to perform “path MTU discovery”

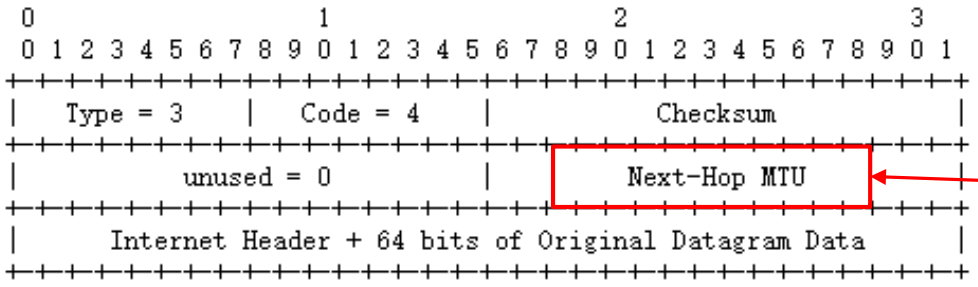
[MD90] J. Mogul and S. Deering. “Path MTU Discovery” Request for Comments 1191, November 1990.

IP报文

Path MTU discovery的原理是不断增大报文并设置DF位，直到中间路由器发出错误ICMP（类型3不可达）消息，告诉链路实际的MTU，就可以推断PMTU。但是PMTU不是精确的，愿意是路径变化。

4. Router specification

When a router is unable to forward a datagram because it exceeds the MTU of the next-hop network and its Don't Fragment bit is set, the router is required to return an ICMP Destination Unreachable message to the source of the datagram, with the Code indicating "fragmentation needed and DF set". To support the Path MTU Discovery technique specified in this memo, the router MUST include the MTU of that next-hop network in the low-order 16 bits of the ICMP header field that is labelled "unused" in the ICMP specification [7]. The high-order 16 bits remain unused, and MUST be set to zero. Thus, the message has the following format:



下一跳MTU小于报文无法发送

IP报文

对分片的攻击方法

- Ping of Death

IP报文的最大长度是2**16-1=65535个字节，那么去除IP首部的20个字节和ICMP首部的8个字节，实际数据部分长度最大为：65535-20-8=65507个字节。但是分片后offset最大可以到2**13*8=65536，重组后IP长度可以超过65535字节，构建超长ping报文。

（Bluetooth protocol就受到类似攻击影响）

- Teardrop

The teardrop attack sent packet fragments with overlapping offsets, which caused implementations that didn’t check for this irregular condition to inevitably crash.

IP报文

- jolt2

在一个死循环中不停的发送ICMP/UDP的IP碎片，可以使Windows系统的机器死锁。攻击Windows 2000，CPU利用率会立即上升到100%，鼠标无法移动。

```
01/07-15:33:26.974096 192.168.0.9 -> 192.168.0.1
ICMP TTL:255 TOS:0x0 ID:1109 IpLen:20 DgmLen:29
Frag Offset: 0x1FFE  Frag Size: 0x9

08 00 00 00 00 00 00 00 00
```

Vulnerable systems:

- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows NT 4.0 Workstation
- Microsoft Windows NT 4.0 Server
- Microsoft Windows NT 4.0 Server, Enterprise Edition
- Microsoft Windows NT 4.0 Server, Terminal Server Edition
- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Cisco 26xx
- Cisco 25xx
- Cisco 4500
- Cisco 36xx

IP报文

- Tiny fragment

攻击者通过操作，可将 TCP 报头 (通常为 20 字节) 分布在 2 个分片中，这样一来，目的端口号可以包含在第二个分片中。

对于包过滤设备或者入侵检测系统来说，首先通过判断目的端口号来采取允许 / 禁止措施。但是由于通过恶意分片使目的端口号位于第二个分片中，通过这种方法可以绕过一些入侵检测系统及一些安全过滤系统。对于路由器的flow统计也不利。

- Fragment Overlap

结合Tiny fragment和teardrop。攻击者为了发动攻击将攻击IP包分为两个分片。第一个分片中包含包过滤设备允许的 http(TCP 80) 等端口。在第二个分片中通过极小的偏移量造成第二个分片覆盖第一个分片的一部分内容。通常攻击者覆盖包含端口内容的部分。

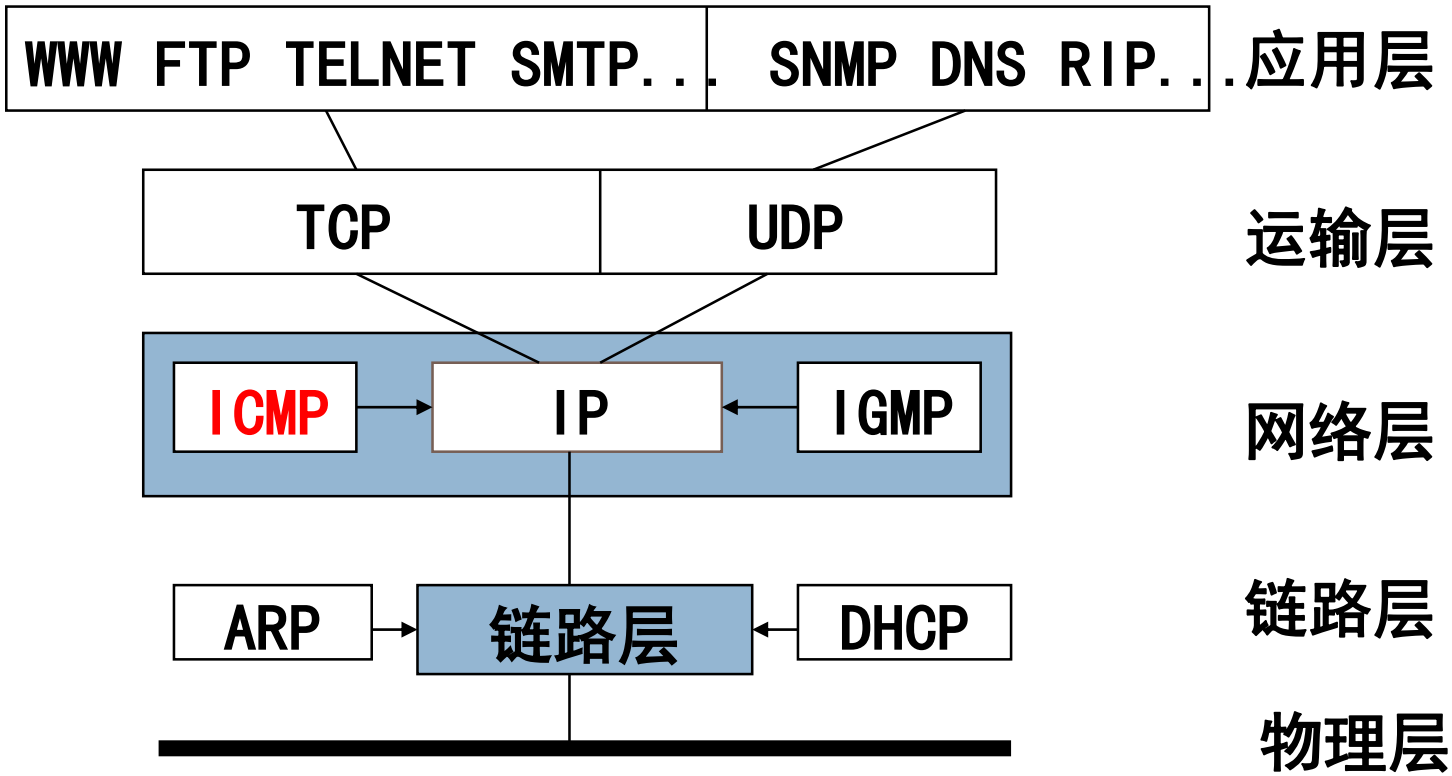
IP报文

结论

- 现在所有的操作系统默认发出的报文都设置了DF，不分片
- 没有分片后，IP头部的域更少了
- 跟其他协议不一样，IP在Internet大发展的过程中反而简化了

错误报告协议(ICMP)

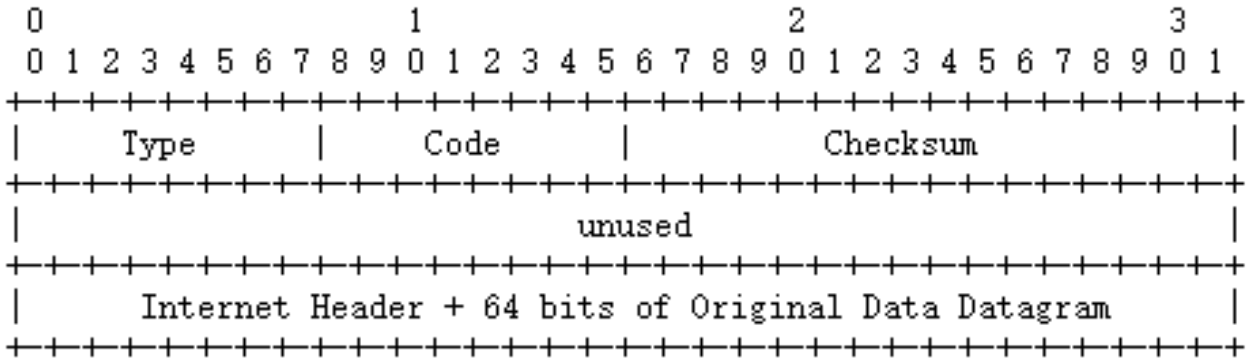
- 协议层次的回顾



ICMP协议

ICMP 报文分类

- 差错和控制报文协议：
  - 报告IP传输中发生的（差错报文+控制报文+测试报文）
  - ICMP报文=头部+数据部分
- ICMP报文封装在IP数据包中进行传输
  - IP头中的包类型=1;
  - ICMP并不是IP的上一层协议，仅用IP的转发功能



数据部分，变化

ICMP协议

ICMP最典型的应用

ping：

- 通过发送回送请求报文（echo request）和回送回答报文（echo reply）来检测源主机到目的主机的链路是否有问题，
- 同时可以检测目的地是否可达，以及通信的延迟情况。

ICMP协议

ICMP最典型的应用

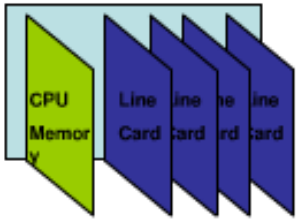
traceroute：通过发送探测报文来获取路由信息。

- 第一个探测报文TTL为1，到达第一个路由器时，TTL减1为0所以丢掉这个探测包，同时向源主机发回ICMP时间超过报文，这时源主机就获得了第一个路由器的IP地址；
- 接着源主机发送第二个探测报文，TTL增1为2，到达第一个路由器TTL减1为1并转发探测包到第二个路由器，这时TTL减1为0，丢掉这个探测包并向源主机发回ICMP时间超过报文，源主机就获得了第二个路由器的IP地址；
- 以此类推，直到探测报文到达traceroute的目的地，这时源主机就获得了到目的地的每一跳路由的IP地址。

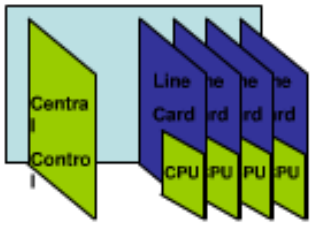
1.4.3 路由与寻址

- IP寻址
  - 每个报文经过每路由器都要决定从哪个接口到下一跳？
  - 路由器需要匹配包之目的地址与路由表，选择最佳路由
- 路由器：网络层互连设备，丰富灵活，可扩展
  - 静态路由：手工配置/简单拓扑/直观；不适合大规模/多变网络
  - 动态路由：实时自动动态更新，并算出当前最佳路由
- 问题：
  - 路由器怎样实时知道变化着的网络拓扑？
  - 路由器怎样选择每个报文的最优下一跳？
- 方法:
  - 通过路由协议获得节点和链路的信息
  - 通过路由算法选择走向目的地的最优下一跳

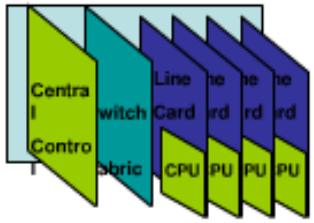
First Generation
Single CPU – multiple line cards
Single electrical backplane



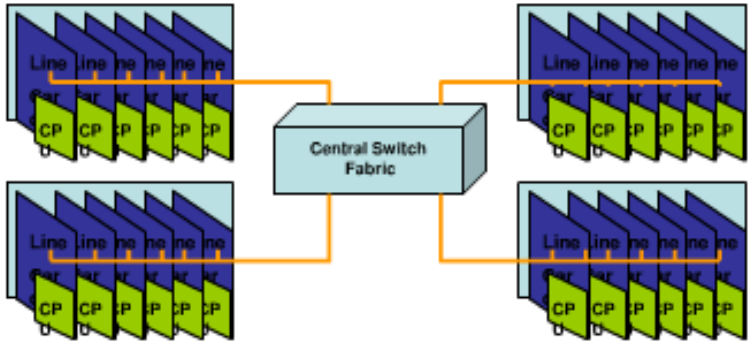
Second Generation
One CPU per Line Card
Central Controller for Routing Protocols

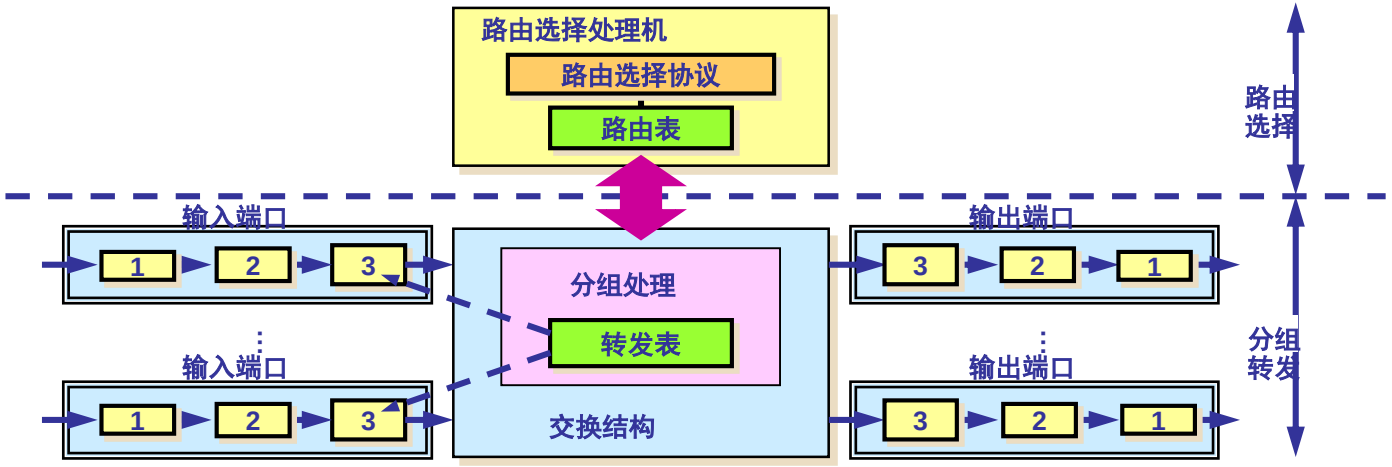


Third Generation
One CPU per Line Card
Central Controller for Routing Protocols
Switch Fabric for inter-connection



Fourth Generation
Multiple shelves of Line Cards
Centralized Switch Fabric
Optical links interconnecting Line Cards and Fabric

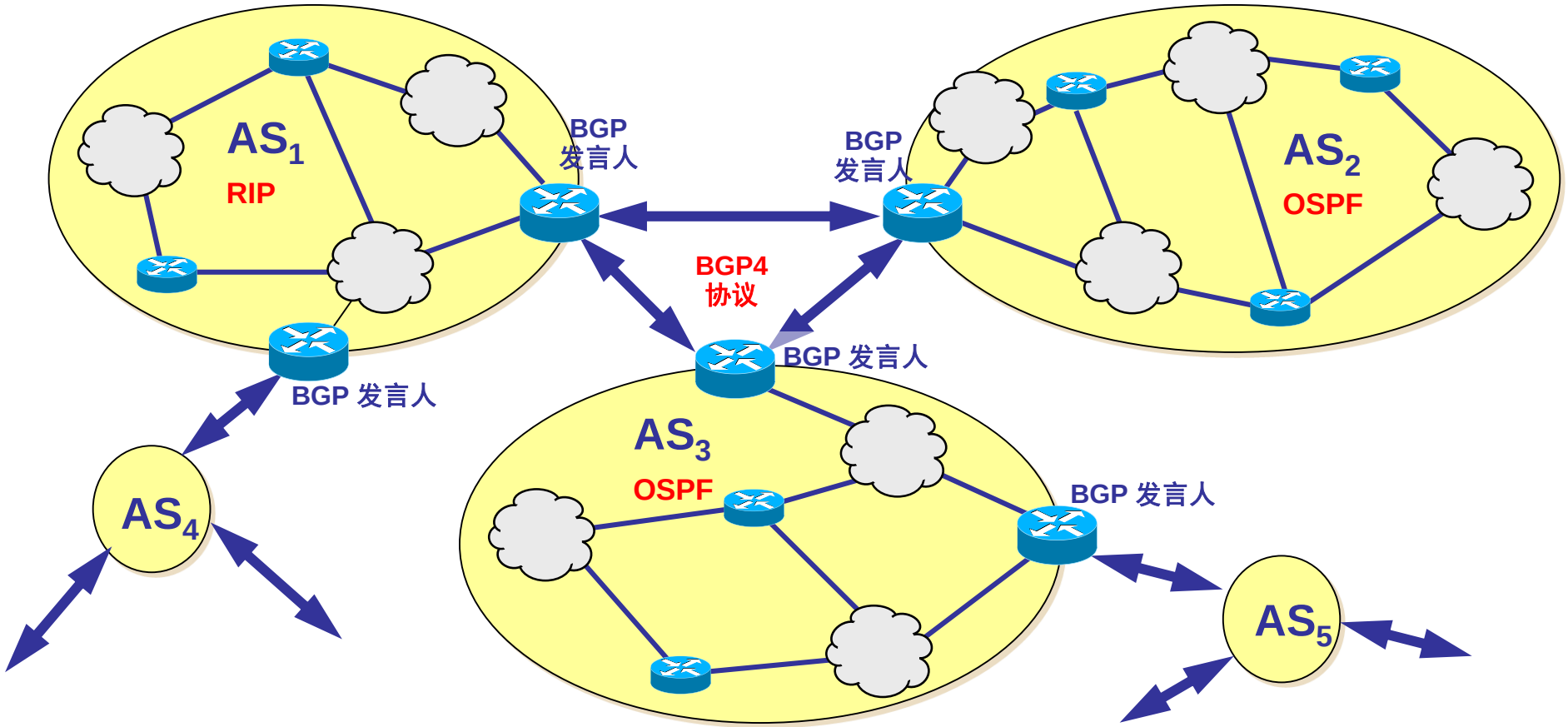




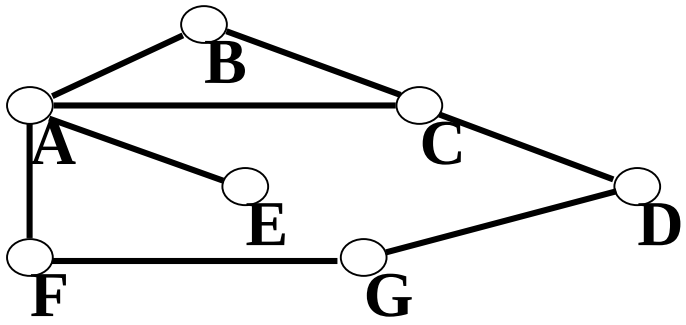
典型路由器的结构

路由协议

- 自治域AS
  - ICANN分配唯一编号。全球所有AS组成因特网。
- 内部网关协议IGP：在一个AS内运行
  - RIP(Routing Information Protocol)适应小规模网络：定时只向邻居说（如30s），说知道的所有（自路由表）；
  - OSPF（Open Shortest Path First Protocol）适应较大规模网络：（链路）变化时向所有人说（区内路由器组播），只说邻居的事（链路状态）
- 外部网关协议EGP：在AS间运行
  - 原因：因特网规模大，AS间直接OSPF收敛时间很长；AS间最佳路由不现实（cost意义不统一）；各AS管理策略不同。
  - BGP4：只寻求一条能到达目网的较好路由（不循环），并不最佳；
  - BGP采用路径向量（Path Vector）路由选择协议；
  - 变动触发BGP发言人（对接路由器）间交换彼此的路由信息


AS1
RIP
BGP
发言人
BGP
发言人
AS2
OSPF
BGP4
协议
BGP 发言人
BGP 发言人
AS3
OSPF
BGP 发言人
AS4
AS5

距离向量算法,如RIP采用（P272）



Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

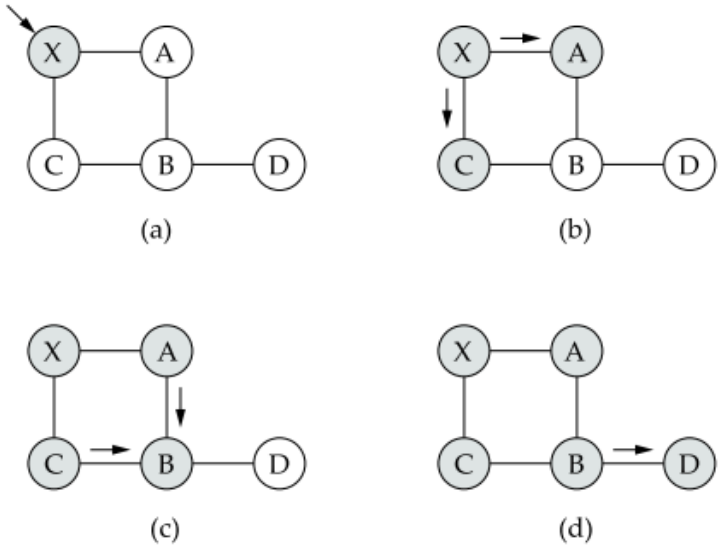
B和A、C交换自己的路由表全部信息

B has the route （E，2，A）

链路状态算法：如OSPF采用(P277)

LSP(link-state packet)

OSPF一个Domain中的路由器会交换链路状态，LSP采用洪泛的方法传给所有Domain中的路由器：



LSP包含的内容：

- The ID of the node that created the LSP;
- A list of directly connected neighbors of that node, with the cost of the link to each one;
- A sequence number;
- A time to live for this packet.

链路状态算法：如OSPF采用(P277)

LSP产生的原因主要有两个：

1）路由器定期产生，由于flooding会影响性能，所以这个周期很长

2）路由器定期向邻居发hello报文，探测邻居可达性，一旦发现拓扑变化，就发出LSP

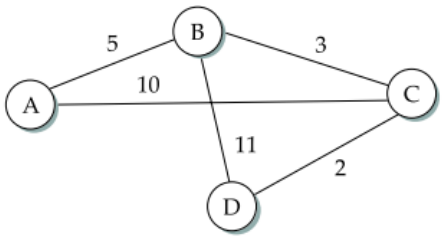
LSP推动路由器重新计算OSPF路由

LSP在Cisco路由器的OSPF实现中对应link-state advertisement (LSA)

链路状态算法：如OSPF采用(P277)

OSPF计算过程：

- 路由器D计算网络中的路由
- Tentative 和Confirmed是计算过程的两个路由集合
- D检查LSP的顺序是D、C、B、A



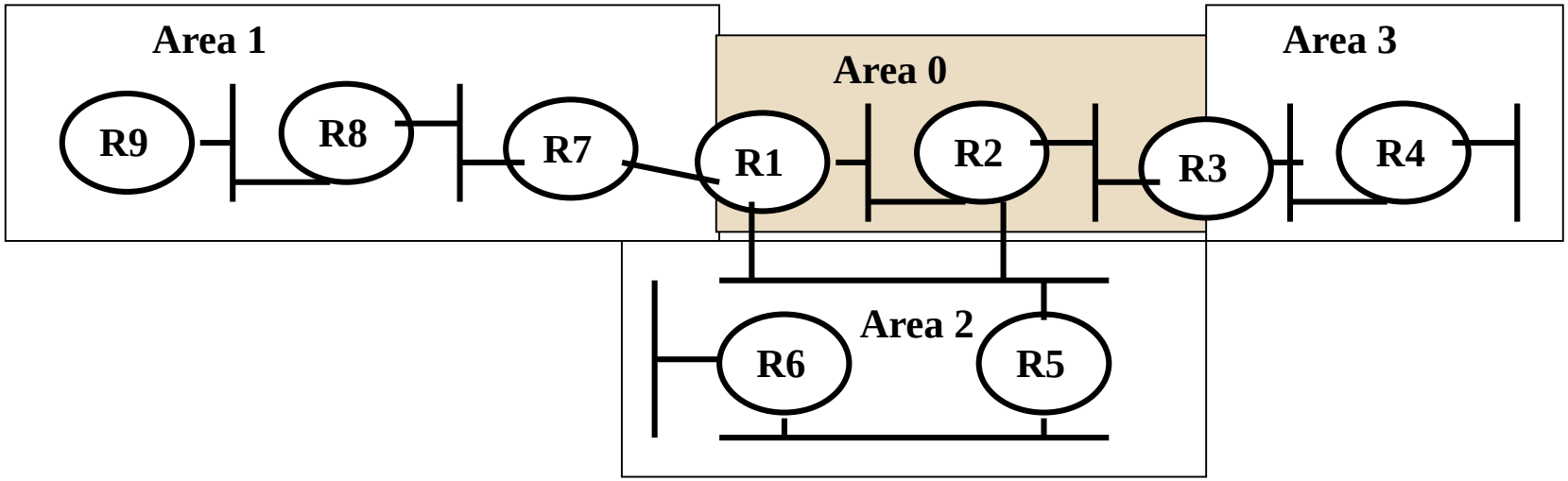
Step	Confirmed	Tentative	Comments
1	(D,0,-)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,-)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,-) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,-) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,-) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,-) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,-) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.

路由区

- 因为OSPF路由器之间会将所有的链路状态（LSA）相互交换（洪泛），当网络规模达到一定程度时，LSA需要较大带宽，也需要较大的存储空间，势必会给OSPF计算带来巨大的压力；
- 为了能够降低复杂程度，OSPF将Domain分为区，采用分区域计算
- 每个区域负责各自区域精确的LSA传递与路由计算，然后再将一个区域的LSA简化和汇总之后转发到另外一个区域
- 在区域内部，拥有网络精确的LSA，而在不同区域，则传递简化的LSA。

路由区

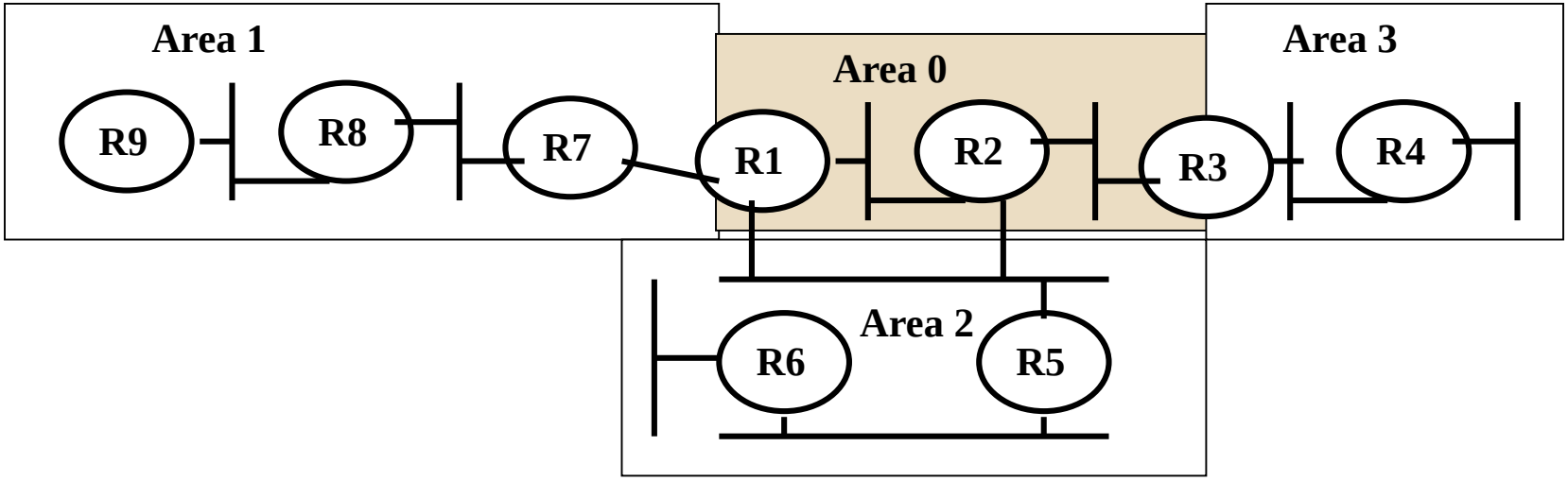
- OSPF为了防止环路产生，以采用了Hub-Spoke的拓朴架构来组织各个区，主干是一个特别区--Area 0骨干区域
- 其它区域称为Normal区域（常规区域），在理论上，所有的常规区域应该直接和骨干区域相连，常规区域只能和骨干区域交换LSA，常规区域与常规区域之间即使直连也无法互换LSA
- OSPF 区域是基于路由器的接口划分的，而不是基于整台路由器划分的，一台路由器可以属于单个区域，也可以属于多个区域



路由区

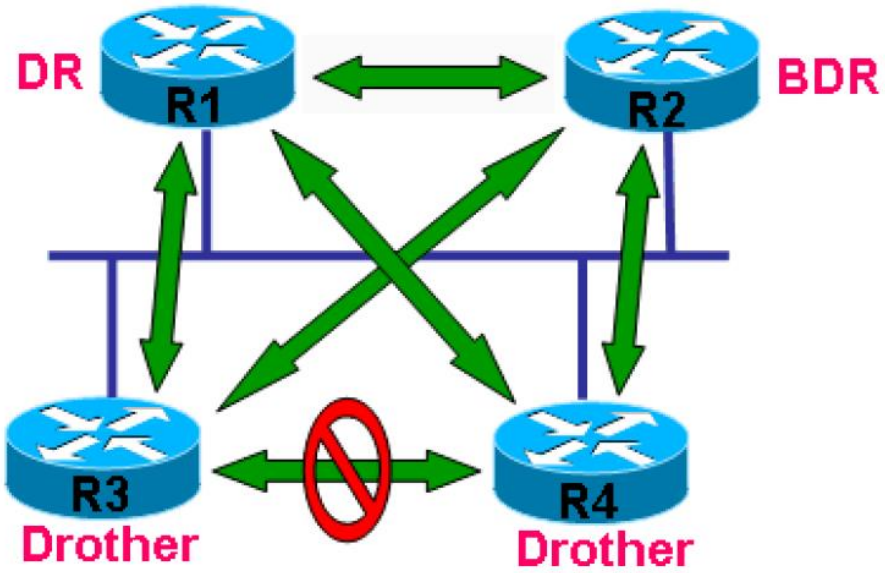
如果一台 OSPF 路由器所有接口属于单个区域，，那么这台路由器称为 Internal Router（IR），如图中的 R4，R7和 R5

如果一台 OSPF路由器有个多个接口属于多个区域，那么这台路由器称为Area Border Router（ABR）



路由区

- 当多台 OSPF 路由器连到同一个多路访问网段（以太网）时，如果每两台路由器之间都相互交换 LSA，那么该网段将充满着众多 LSA 条目
- 为了能够尽量减少 LSA 的传播数量，通过在多路访问网段中选择出一个核心路由器，称为 DR（Designated Router），网段中所有的 OSPF 路由器都和 DR 互换 LSA
- BDR（Backup Designated Router）是DR的备份，其他路由器是Drother



路由区

OSPF工作过程

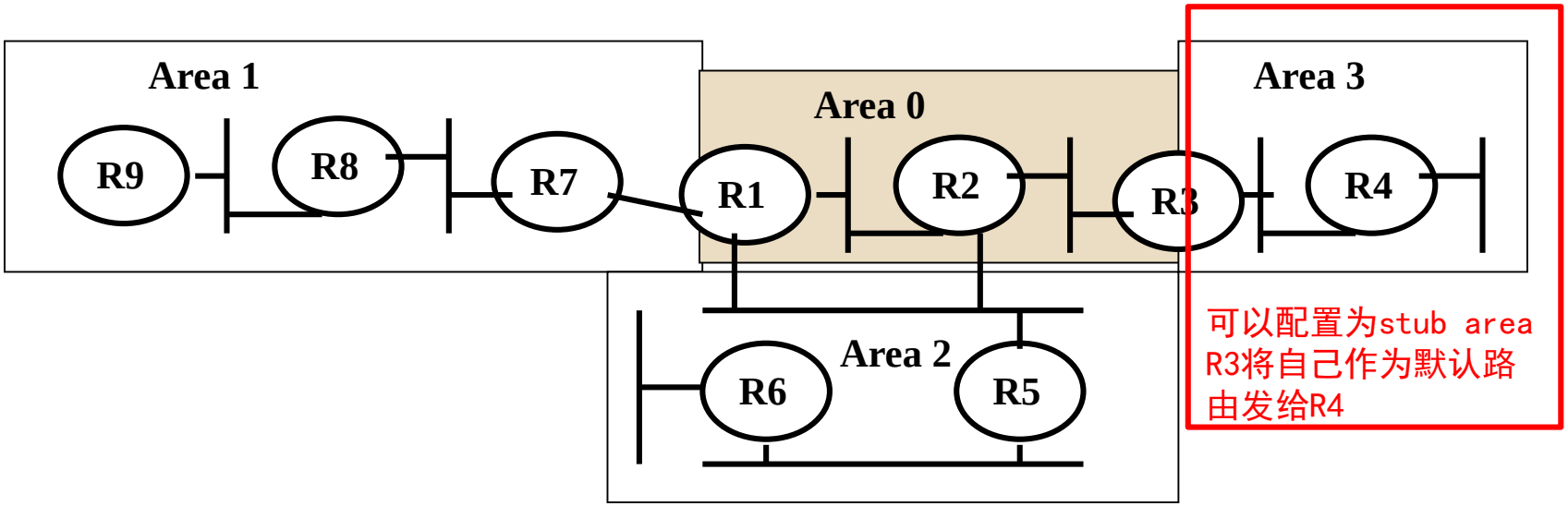
1. Hello包是用来建立和维护 OSPF 邻居的，要交换 LSA，必须先通过用组播地址（224.0.0.5）发送Hello包建立OSPF 邻居。
2. Database Description Packets（DBD） 邻居建立之后，并不会立刻就将自己链路状态数据库中所有的 LSA 全部发给邻居，而是将 LSA 的基本描述信息发给邻居，这就是 Database Description Packets （DBD），是 LSA 的目录信息
3. Link-state Request（LSR）邻居在看完发来的 LSA 描述信息（DBD）之后，就知道哪些 LSA 是需要邻居发送给自己的，自己就会向邻居发送 LSA 请求（LSR），告诉邻居自己需要哪些 LSA。
4. Link-state update（LSU）当邻居收到其它路由器发来的 LSA 请求（LSR）之后，将需要的LSA 内容全部发给邻居，以供计算路由表。
5. 如果已经收到了所有需要邻居发给自己的 LSA，这时的链路状态数据库（LSDB）已经达到收敛状态。

路由区

OSPF 末节区域

在一个末梢网络中，许多路由信息是多余的，并不需要通告进来，因为一个OSPF区域内的所有路由器都能够通过该区域的ABR去往其它OSPF区域或者OSPF 以外的外部网络

所以末梢网络的路由器只要知道去往ABR，就能去往区域外的网络这样的区域称为OSPF末节区域（Stub Area）；



路由区

Cisco路由器还把Stub Area细分为：

1. Stub Area（末节区域）
2. Totally Stub Area（完全末节区域）
3. Not-so-Stubby Area（NSSA）
4. Totally Not-so-Stubby Area（Totally NSSA）

OSPF 由于有着多种区域类型，多种网络类型，多种链路类型，多种路由器身份，所以 LSA分为11种