



第五单元 网络层 -OSPF协议

2016.5.3



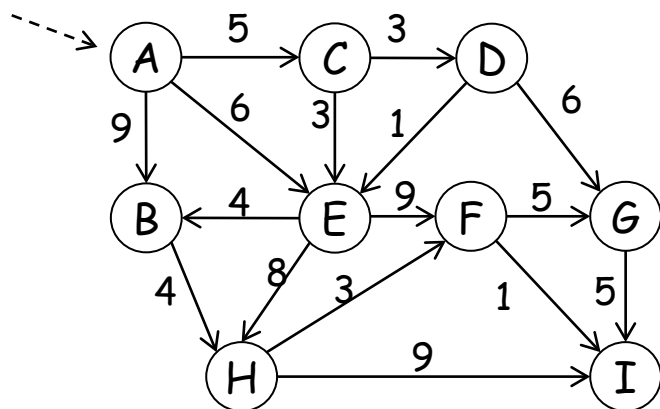
本节内容

- ❑ 概述
- ❑ 把网络转变为图
- ❑ OSPF协议的详细描述
- ❑ OSPF开销
- ❑ 数据库同步
- ❑ 路由器ID
- ❑ 指定路由器
- ❑ 分区OSPF
- ❑ OSPF分组格式
- ❑ LS Update分组
- ❑ Dijkstra最短路径算法

概述

□ 链路状态(link state)路由算法:

- (1) 利用最短路径算法(例如: **Dijkstra**最短路径算法)求出一个节点(源节点)到所有其它节点的最短路经。
- (2) 利用这些最短路径上的下一个节点作为下一跳得到源节点的转发表(路由表)。



节点A的转发表

目的	下一跳	距离
B	B	9
C	C	5
D	C	8
E	E	6
F	B	14
G	C	14
H	B	13
I	E	15

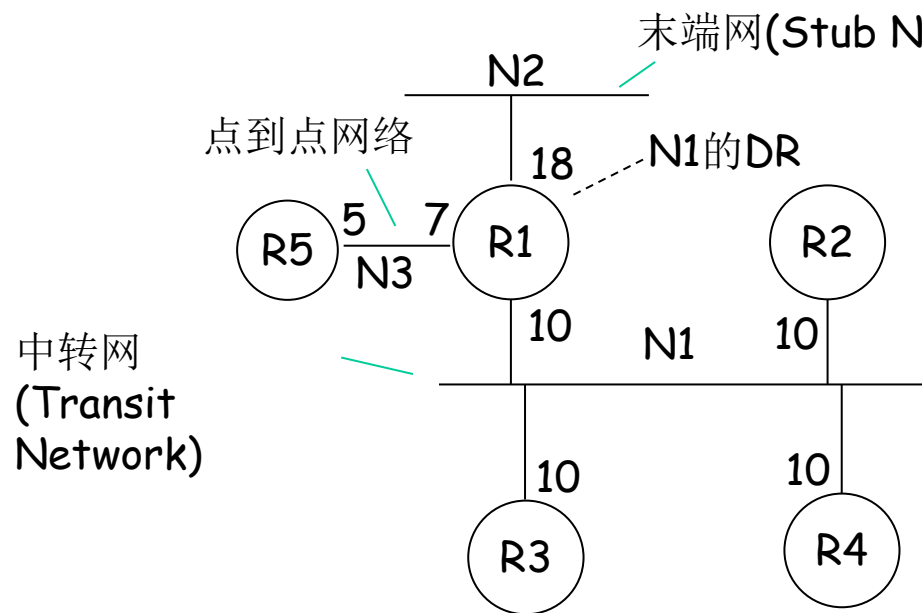
节点A的链路状态: $\langle AB, 9 \rangle$ $\langle AC, 5 \rangle$ $\langle AE, 6 \rangle$

<http://tools.ietf.org/html/rfc2328>

- ❑ **OSPF** 协议(**Open Shortest Path First**)采用链路状态路由算法, 它可能是在大型企业中使用最广泛的内部网关协议 (**IGP**)
- ❑ **OSPF**协议的简单描述:
 - (1) 周期性地收集链路状态, 并扩散给**AS**中的所有路由器;
 - (2) 用收到的链路状态建立整个**AS**的拓扑结构图;
 - (3) 利用**Dijkstra**算法计算到**AS**中所有网络的最短路径;
 - (4) 利用这些路径上的下一跳建立路由表。

如何利用把整个网络(**AS**)转化为**AS**的拓扑结构图?

把网络转变为图



链路状态通告 (Link State Advertisement)

- N1和N2是多路访问网络，例如：以太网
- N3为点到点网络，例如：ppp

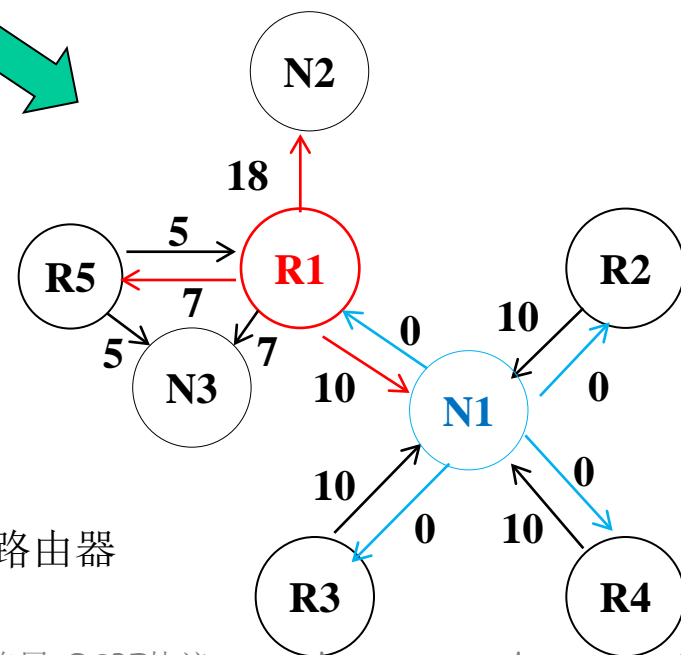


R1's Router LSA:

	R1 (From)
N1	10
N2	18
R5	7
N3	7

N1's Network LSA:

	N1(From)
R1	0
R2	0
R3	0
R4	0



- 对于每个中转账，要选举一个直连路由器作为其指定路由器 (designated router, DR)
- 如果图中点到点网络没有配置IP地址，则不要节点N3

N1's Network LSA:

	N1(From)
R1	0
R2	0
R3	0
R4	0

链路状态数据库

	R1	R2	R3	R4	R5	R6	N1	N2	(From)
R1							0		
R2							0		
R3							0		
R4							0		
R5	7								
R6									
N1	10								
N2	18								
N3	7								

R1's Router LSA:

	R1 (From)
N1	10
N2	18
R5	7
N3	7

- R2~R5 的Router LSAs也将被加入到链路状态数据库中。
- 链路状态数据库可以形成AS拓扑结构图的邻接矩阵。

OSPF协议的详细描述

❑ 发现邻居

OSPF路由器每10秒 (**Hello Interval**)向邻居发送**Hello**分组。如果40秒(**Dead Interval, 4 times hello interval**)都收不到邻居发来的**Hello**分组, 则把到邻居的链路标记为失效。多路访问网络采用多播 (**224.0.0.5, all OSPF routers**) 发送**Hello**分组。一个**Hello**分组包含优先权、已知的邻居 (收到过**Hello**)、**DR**和**BDR**。

❑ 完全相邻

在发现邻居之后, **OSPF**路由器将与邻居交换链路状态数据库中的**LSA**, 使之全部为最新**LSA**。在与邻居的链路状态数据库变得完全一样时, 它们就处于完全相邻状态 (**fully adjacency**)。

❑ 生成LSA

每30分钟或链路变化时, 每个**OSPF**路由器会生成 **router LSA**, 中转网的**DR**会生成 **network LSA**。

❑ 扩散LSA

产生的 **LSA**立即封装为**Update**分组, 被可靠地扩散出去 (需要确认)。每次产生的**LSA**的序号会加**1**。序列号越大表示越新。若通过收到多个 **LSA**, 由发出此**LSA**的路由器**ID**(发通告路由器), 链路状态和序列号唯一确定。通过序号, 也可以防止扩散形成回路, 第二次收到来自相同的发通告路由器、相同**LSA**类型和相同序号的**LSA**将丢弃它。

❑ 收集LSA

路由器收集到LSA之后，用新LSA替换链路状态数据库中旧LSA。如果一个LSA在60分钟(max age)没有被更新，它将从链路状态数据库移除。

❑ 计算最短路径

当链路状态数据库被改变时，OSPF路由器将利用Dijkstra算法计算到所有网络的最短路径。

❑ 建立路由表

利用得到的最短路径产生路由表。

OSPF开销

接口类型	带宽 (bps)	OSPF开销
Ethernet	10M	10
Fast Ethernet	100M	1
Gigabit Ethernet	1G	1
T1	1.544M	64
E1	2.048M	48
56000 point-to-point network	56K	1785
19.2K point-to-point network	19.6K	5208

- ❑ 开销的实际计算方法: **reference-bandwidth/bandwidth (Mbps)**
开销必须为大于0的整数, reference-bandwidth的默认值为100。
- ❑ 修改reference-bandwidth的方法:
`#ospf auto-cost reference-bandwidth 1000`

数据库同步

接口处于shutdown
状态



向对方不停地
发送Hello

Attempt

Hello(DR=0, Seen=0)

----->

Hello(DR=0, Seen=0)

<-----

收到对方的Hello

Init

Hello(DR=0, Seen=0)

<-----

对方hello指出已收
到自己的Hello

Two-way

Hello(DR=0, Seen=R1,...)

<-----

neighbor

广播型网络
要选举DR

DR Elected
if needed

Hello(DR=0, Seen=R1, DR=R1)

<----->

ExStart

发送不带LSA的DD确定
主从关系
RID大的成为Master

DD(Seq=x,I,M,Master)

----->

M-More

I- initialize

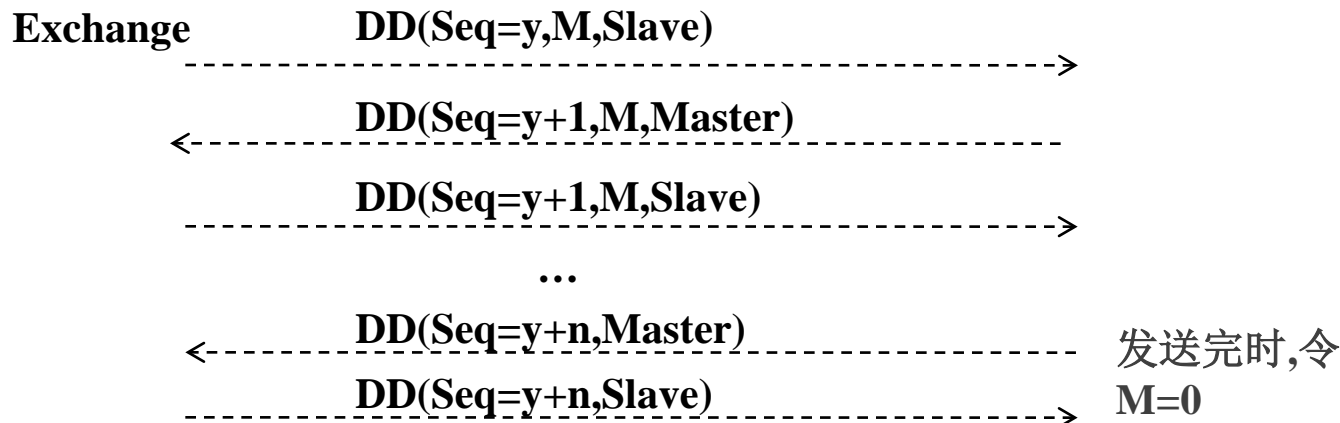
DD(Seq=y,I,M,Master)

<-----

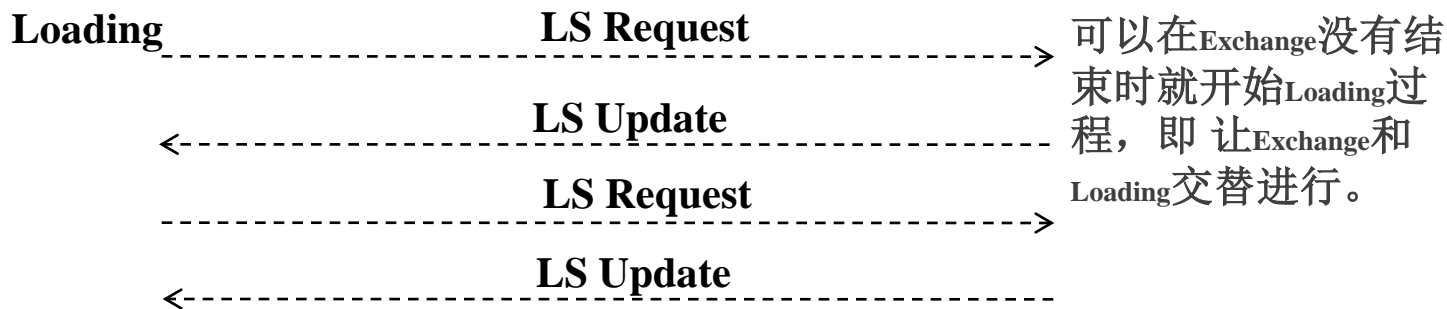
DD指 Database Description Packet



交换LSA头部，
以便知道自己的
LS数据库中过时
或缺少的LSA



双方交换完整
的LSA



收到了所有过时
或缺少的LSA

Full

Fully adjacent

DROther把所有DD多播(224.0.0.6)给DR和BDR, DR单播确认, DR把新的DD多播(224.0.0.5)给该网络的所有OSPF路由器。对于无DR网络, OSPF路由器直接多播(224.0.0.5)DD和LSU给所有OSPF路由器。

路由器ID

- ❑ OSPF协议采用路由器ID(Router ID, RID)标识每一个路由器。
- ❑ 路由器ID由以下方法得到：
 - ① 使用直接配置的RID (`#router-id id`) 。
 - ② 所有活动环回接口中最大的IP地址。
 - ③ 所有活动物理接口中最大的IP地址。
- ❑ 除非路由器重启、所选接口故障或关闭或IP地址改变、重新执行了`router-id`命令，RID将保持不变。

LSA定时器*

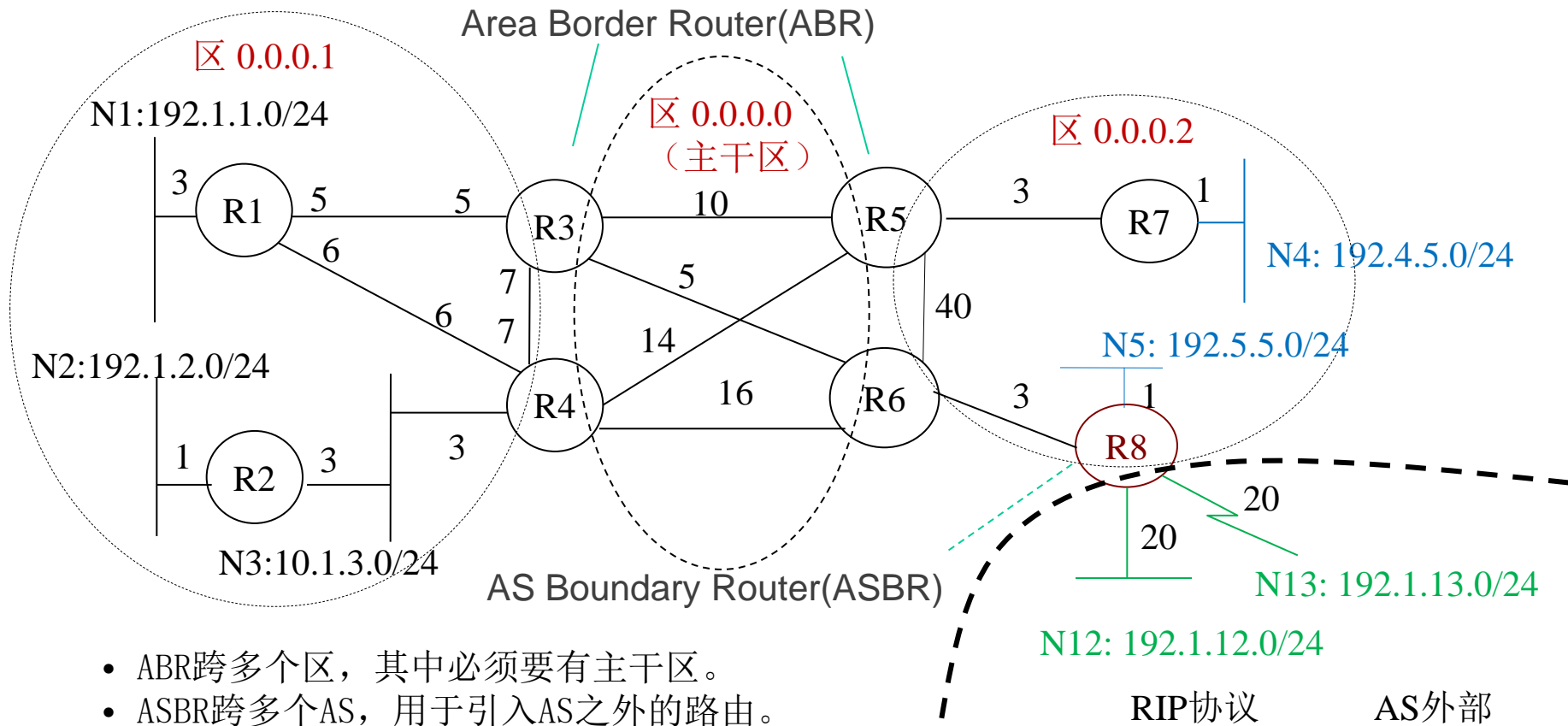
- ❑ 每10秒(Hello Interval)向邻居发送一次Hello，4倍的hello interval (Dead Interval, 40秒)没有收到邻居的Hello就认为邻居失效。
- ❑ 每30分钟会产生新的LSA，最小间隔时间为5秒。
- ❑ 每个LSA都有年龄字段(age)，发给邻居时被设置为0，在链路状态数据库中age会不断增长，增长到Max Age(默认为60分钟)时LSA被标记为失效。失效的LSA会被扩散到整个AS，令AS的所有路由器把该LSA从链路状态数据库中移除。
- ❑ 存储在链路状态数据库中的LSA每10分钟会被计算校验和，如果有错将被删除。
- ❑ 接收来自邻居的LSA的最小间隔时间为1秒。
- ❑ 计算最短路径的最小间隔时间为10秒。

指定路由器*

- ❑ 当多路访问网络重启时，选择**DR**的过程就开始了。在等待时间结束 (Wait Time, Dead Interval, 40秒)时，带有最高和次高优先权的路由器分别成为**DR**和 **BDR(Backup DR)**。如果优先权相同，**RID**更大的成为**DR**，次大的成为**BDR**。
- ❑ 如果路由器不希望参与选举，则应该把优先权设置为**0**。如果优先权相同，具有更高**RID**的路由器成为**DR**。如果收到的**Hello**列出了**DR(RID不是0.0.0.0)**，路由器成为**DR**。
- ❑ 如果一个新的路由器在选举之后到达或者有路由器修改为更高的优先权，它也不可能抢占现存的 **DR (或BDR)**和变为**DR(或BDR)**。
- ❑ 当**DR**失效时，**BDR**成为**DR**，将开始一个新的选举过程来选出**BDR**。
- ❑ 一个多路访问网络中的**OSPF**路由器只与**DR**和**BDR**建立相邻关系。
- ❑ 收到一个**LSA**后，一个多路访问网络中的**OSPF**路由器将把它首先多播(224.0.0.6)给**DR**和**BDR**，然后 **DR**再把它多播 (224.0.0.5)给所有**OSPF**路由器。

分区OSPF*

- ❑ 一个**AS**可以划分为多个区以改进效率和安全性。
- ❑ 每个**AS**必须要有主干区(backbone area)，即区0。
- ❑ 所有其它**OSPF**区必须与主干区相邻，即每个区必须有路由器也属于主干区。这样的路由器称为区边界路由器(area border router, **ABR**)。
- ❑ **ABR**需要为其所在的每个区保存一个独立的链路状态数据库。
- ❑ 一个区的链路状态数据库在区内所有路由器中必须相同。
- ❑ **ABR**会把非主干区的网络用Network Summary LSA转入主干区，或者把主干区的网络转入非主干区。
- ❑ 连到**AS**外部的路由器称为**AS**边界路由器(**AS** Boundary Router, **ASBR**)。
- ❑ **ASBR**用**ASBR** Summary LSA和**AS-External** LSA把位于**AS**外部的路由引入**AS**内。
- ❑ 如果一个网络有多条路由，其选择的优先次序：区内路由，区外路由，**AS**内外路由。对于相同类型的多条路由，选择开销小的。这个规则可能导致选择一个非最短距离的路由。

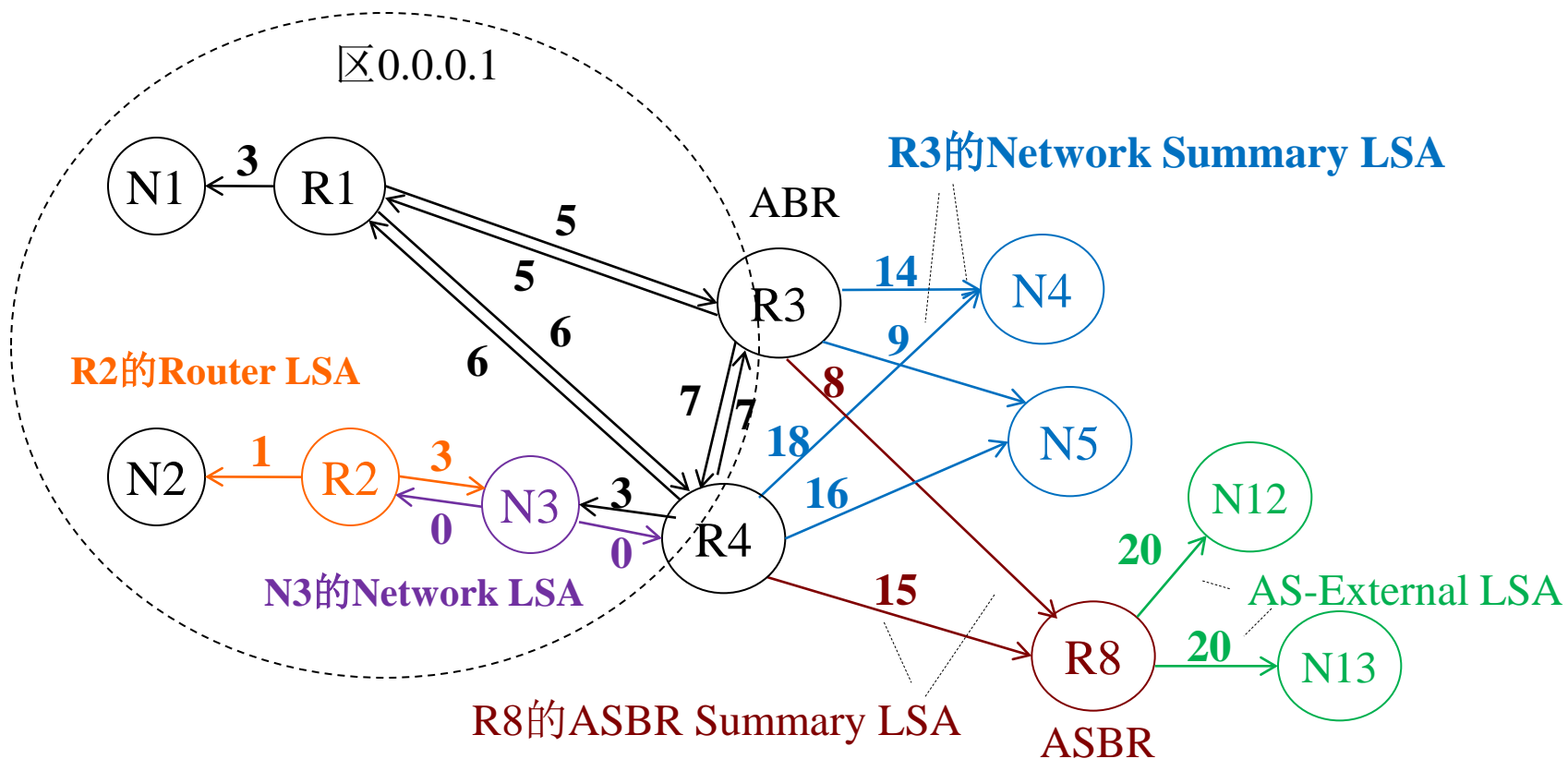


- ABR跨多个区，其中必须要有主干区。
- ASBR跨多个AS，用于引入AS之外的路由。

区1的拓扑结构图包含五种弧：

- (1) 从路由器到邻居的弧；
- (2) 从多路访问网络到邻居的弧；
- (3) 从ABR到本AS的区外网络的弧；
- (4) 从ABR到AS边界路由器(ASBR)的弧；
- (5) 从ASBR到AS外部网络的弧。

✓ 区1的所有路由器都保存以下拓扑结构图(LS 数据库):



(1) R3's Network Summary LSA

	R3 (From)
N4	14
N5	9

(2) R8's ASBR Summary LSA

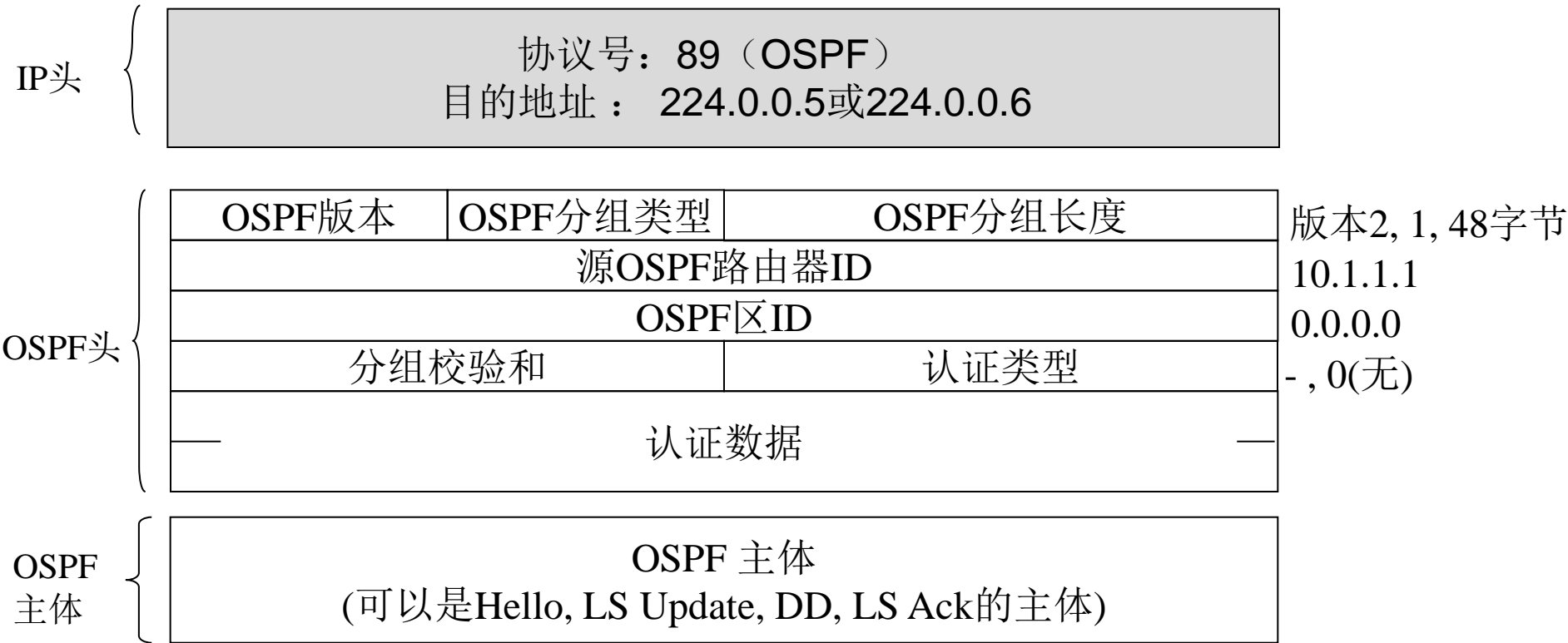
	R3 (From)
R8	8

(3) R8's AS External LSA

	R8 (From)
N12	20
N13	20

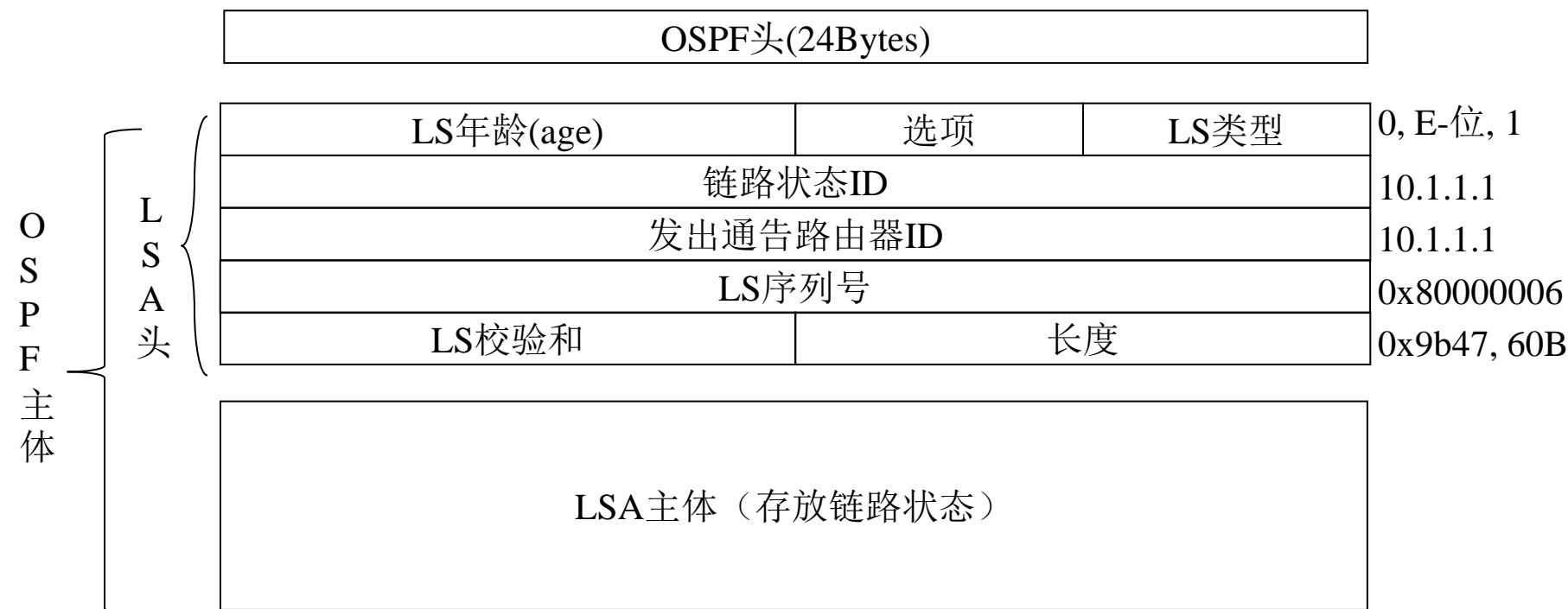
OSPF分组格式

224.0.0.5 -- All OSPF Router
224.0.0.6 -- OSPF DR or BDR



- OSPF分组类型：
- 1--Hello Packet
 - 2--Database Description Packet
 - 3--Link State Request Packet
 - 4--Link State Update Packet
 - 5--Link-State Acknowledge Packet

LS Update分组



- 发出通告路由器ID为产生本分组的路由器ID。
 - 链路状态ID用于区分不同的。Router LSA的链路状态ID与发出通告路由器ID相同，Network LSA的链路状态ID为网络号。
 - LS类型：

1--Router LSA	2--Network LSA
3--Network Summary LSA	4--ASBR Summary LSA(E1和E2)
5--AS-External LSA	6--Group Membership LSA
7--NSSA External LSA(N1和N2)	
- * type1~4都被限制在本区扩散。

Dijkstra最短路径算法

Notation:

- $c(x,y)$: 从 x 到 y 的链路开销; 如果不是直接邻居, 则为 ∞
- $D(v)$: 从源节点到目的节点 v 的当前路径开销。
- $p(v)$: 从源节点到目的节点 v 的路径上最靠近 v 的上一个节点。
- N' : 已经知道最短路径的节点集合

1 Initialization:

```
2   $N' = \{u\}$   
3  for all nodes  $v$   
4    if  $v$  adjacent to  $u$   
5      then  $D(v) = c(u,v)$   
6    else  $D(v) = \infty$ 
```

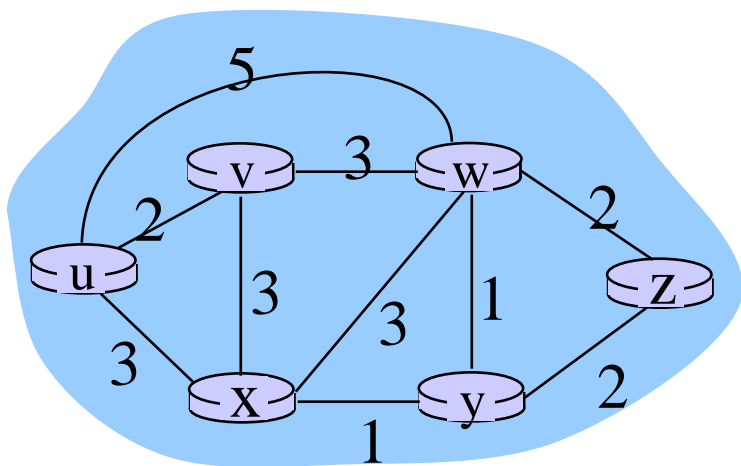
7

8 Loop

```
9  find  $w$  not in  $N'$  such that  $D(w)$  is a minimum  
10 add  $w$  to  $N'$   
11 update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$  :  
12    $D(v) = \min( D(v), D(w) + c(w,v) )$   
13 /* new cost to  $v$  is either old cost to  $v$  or known  
14 shortest path cost to  $w$  plus cost from  $w$  to  $v$  */  
15 until all nodes in  $N'$ 
```

举例：

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



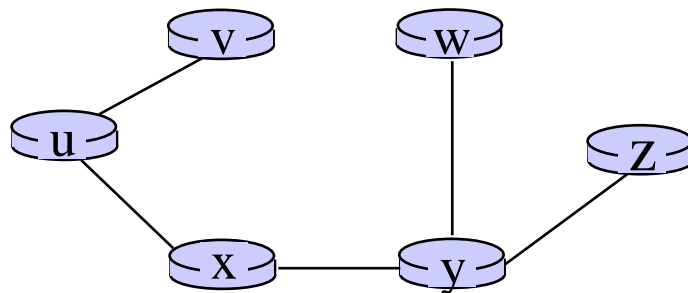
算法复杂性: n 个节点

- 每次循环需要检查所有不在N中的节点
 $n(n+1)/2$ 次比较: $O(n^2)$
- 更有效地算法: $O(n \log n)$

震荡的可能性:

- 例如, 链路开销=实际流量大小。

从u出发的最短路径树:



u的转发表:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

OSPF的特点

- ❑ 所有的OSPF消息都要认证 (防止恶意入侵)。
- ❑ 路由表中允许多个相同开销的路经存在(RIP只允许一条路经)，可以实现负载均衡。
- ❑ 对于每条链路，允许同时有多个(TOS)开销。
- ❑ 多播OSPF (MOSPF)使用与OSPF相同的链路状态数据库 (思科路由器不支持)
- ❑ 在大型路由选择域中OSPF可以分区。
- ❑ 比RIP收敛快而且更安静。
- ❑ 实现起来更复杂，需要更多的计算开销。

LS算法和DV算法比较

消息复杂性

- LS: n 个节点, E 条链路, 要发送 $O(nE)$ 条消息
- DV: 只在邻居之间交换消息

收敛速度

- LS: $O(n^2)$ 算法需要 $O(nE)$ 条消息
 - ❖ 可能会震荡
- DV: 收敛时间变化
 - ❖ 可能出现路由循环
 - ❖ 计数到无穷问题

健壮性: 路由器失效时会出现什么情况?

LS:

- ❖ LS节点可能通告不正确的链路开销
- ❖ 每个节点只计算自己的路由表

DV:

- ❖ DV节点可能通告不正确的路径开销
- ❖ 每个节点的路由表被其它节点所用
 - 错误会通过网络传播开

总结

- ❑ 概述
- ❑ 把网络转变为图
- ❑ OSPF协议的详细描述
- ❑ OSPF开销
- ❑ 数据库同步
- ❑ 路由器ID
- ❑ 指定路由器
- ❑ 分区OSPF
- ❑ OSPF分组格式
- ❑ LS Update分组
- ❑ Dijkstra最短路经算法