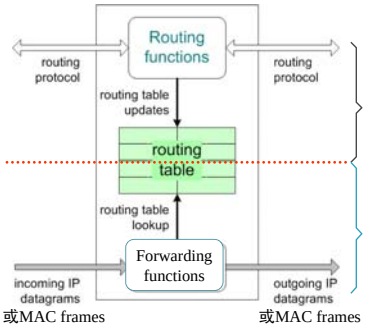
第五章 寻址与路由技术

寻址与路由的关系



控制平面:
路由协议(RIP, OSPF, BGP, IEEE 802.1d/802.1q)
控制路由表

数据平面:
入口分组经路由表查找
再转发至出口

第五章 寻址与路由技术

- 5.1 距离矢量路由
- 5.2 链路状态路由
- 5.3 MAC广播生成树
- 5.4 路由表查找算法
- 5.5 标签交换路由

距离矢量（DV）算法

距离矢量算法来源

- **Bellman-Ford算法**
  - 针对加权有向图，计算源点去往一组(即vector)其他点最短距离和路径；
  - 存在负权回路时，无法解算完成；
  - 存在无穷计数 (count-to-infinity)问题；
  - 在RIP(v1,v2)和IGRP被采用。

Richard Bellman
Aug.26,1920-Mar.19, 1984

Lester Ford, Jr.
(1886-1967)

Bellman, Richard. On a routing problem. *Quart. Appl. Math.* ,1958, 16:87–90.

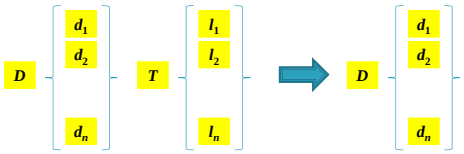
BF算法伪代码，*s*为源节点

```
1: for each vertex v ∈ V[G] do
2:     d[v] ← ∞
3: end for
4: d[s] ← 0
5: for i ← 1 to |V[G]| do
6:     relaxed ← FALSE
7:     for each edge (u, v) ∈ E[G] do
8:         if d[v] > d[u] + w(u, v) then    ▷ Relaxation call
9:             d[v] ← d[u] + w(u, v)    ▷ Relaxation step
10:            relaxed ← TRUE
11:        end if
12:    end for
13:    if relaxed = FALSE then
14:        exit the loop
15:    end if
16: end for
17: for each edge (u, v) ∈ E[G] do
18:    if d[v] > d[u] + w(u, v) then
19:        return FALSE
20:    end if
21: end for
22: return TRUE
```

总循环次数：
O(|*E*||*V*|)

内循环次数：|*E*|

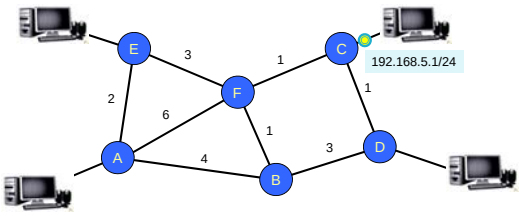
外循环次数：|*V*|

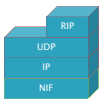


RIP的收敛过程

$$if\ (d_i > l_i + w_t)$$
$$d_i = l_i + w_t$$

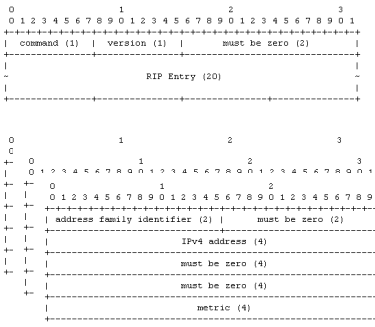
C所知目标，注入A的路由表





Port：520
IPv4 Addr.：224.0.0.9

RIP消息格式(RFC2453)

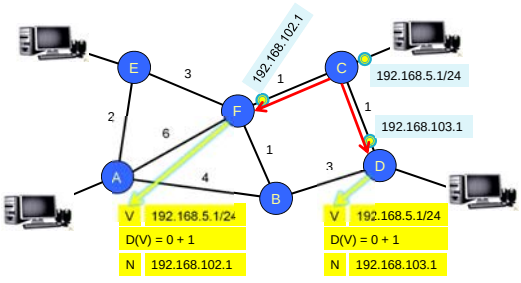


1~25个表入口

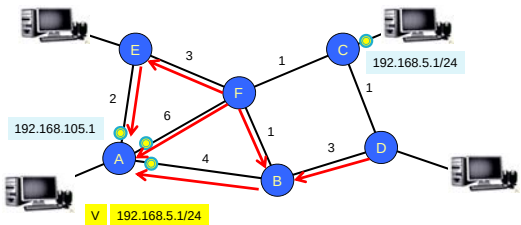
目标地址

权重度量值

节点C组播路由表项



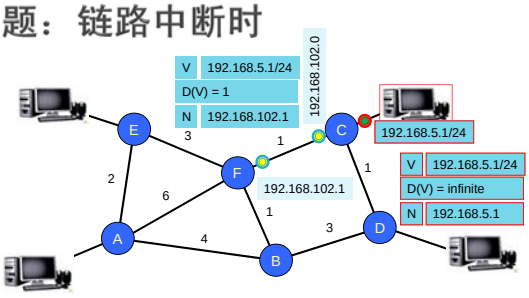
节点A的更新计算



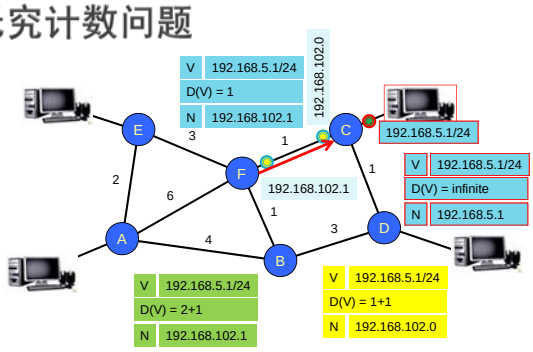
V 192.168.5.1/24
D(V) = 1 + 3 + 2
N 192.168.105.1

距离相等路径的解决方法
1)先到先用;
2)组播更新周期一半之前的优先。

问题：链路中断时



无究计数问题

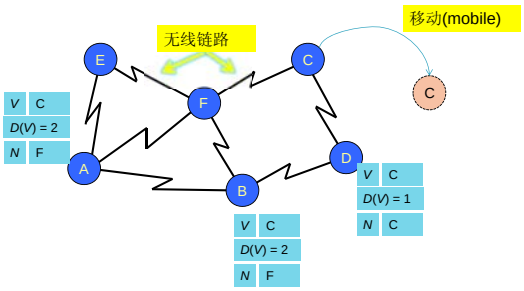


[Prob]如此循环，以至无穷

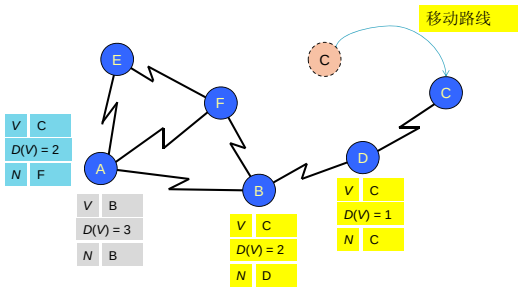
DSDV：RIP的MANET扩展

MANET: Mobile Ad hoc Network
(IETF working group)

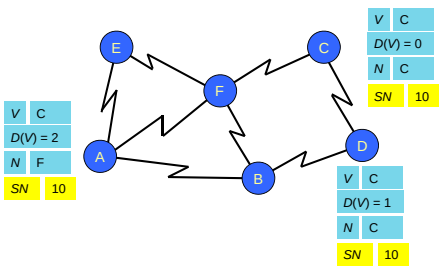
MANET示例



不考虑时效性，更新慢

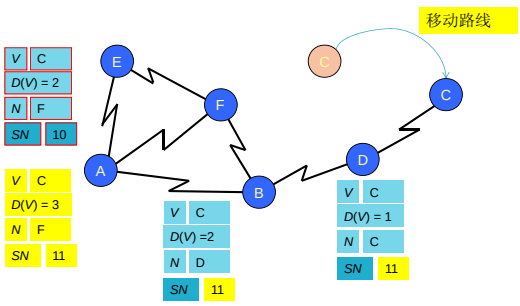


Destination-Sequenced Distance-Vector（DSDV）



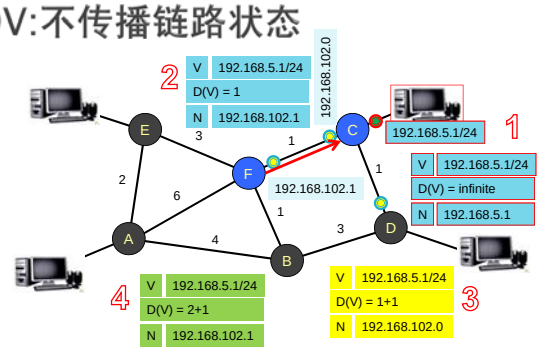
http://www.cs.virginia.edu/~cl7v/cs851-papers/dsdv-sigcomm94.pdf

时效优先更新



第五章 寻址与路由技术

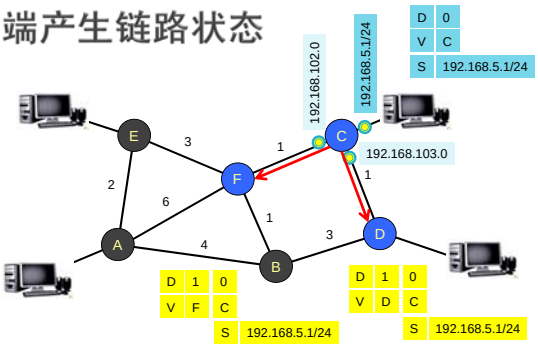
DV:不传播链路状态



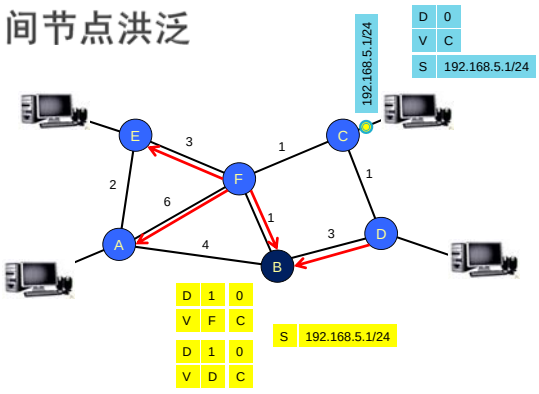
[Prob]如此循环，以至无穷

链路状态及通告

源端产生链路状态

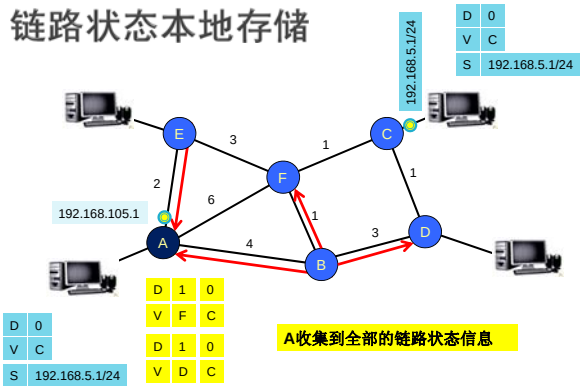


中间节点洪泛



LS(OSPF)计算

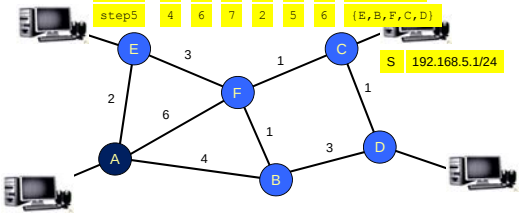
链路状态本地存储



A收集到全部的链路状态信息

A独立计算最短路由

	B	C	D	E	F	S	
step1	4	∞	∞	2	6	∞	{E}
step2	4	∞	∞	2	5	∞	{E,B}
step3	4	∞	7	2	5	∞	{E,B,F}
step4	4	6	7	2	5	6	{E,B,F,C}
step5	4	6	7	2	5	6	{E,B,F,C,D}



算法

Input: Graph (N,E) with

N the set of nodes and $E \subseteq N \times N$ the set of edges

d_{vw} link cost (d_{vw} = infinity if $(v,w) \notin E$, $d_{vv} = 0$)

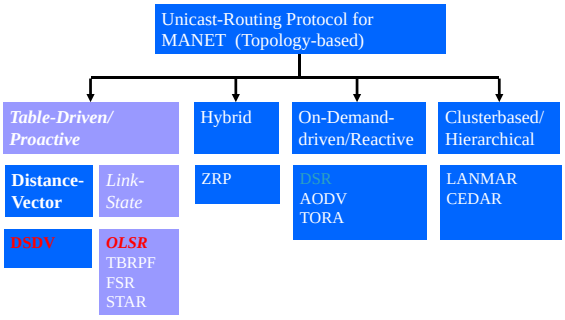
s source node.

Output: D_n cost of the least-cost path from node s to node n

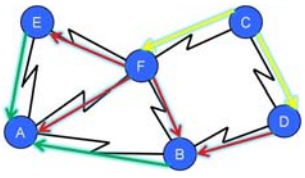
```
M = {s};
for each  n ∉ M
        D_n  = d_sn;
while (M ≠ all nodes) do
        Find w ∉ M for which D_w = min{D_j ; j ∉ M};
        Add w to M;
        for each  n ∉ M
                D_n = min_w [ D_n, D_w + d_wn ];
                Update route;
enddo
```


OLSR：LS的MANET扩展

MANET路由协议类别



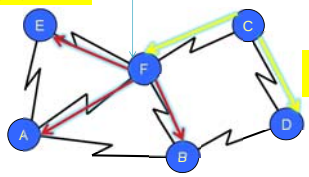
LSA通告



源自C的LSA消息总数为8条

OLSR(优化后LS路由)通告

MPR(Multi-Point Relay)
负责2跳的单跳邻近节点

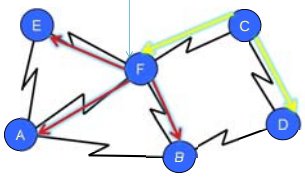


[Prob]
节点D为何不是MPR

源自C的OLSR消息总数为5条
所以，节能3/8

MPR选择（以节点C为例）

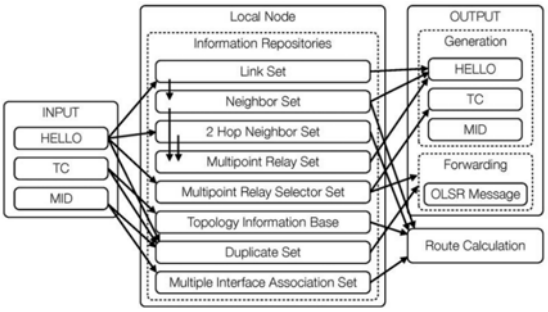
1）2跳邻接节点最多的单跳邻居



2）删除已选的MPR及对应的2跳邻接节点

3）重复1）和2），甚至所有2跳节点全部删除

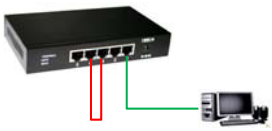
OLSR消息类型与数据流



MID：Multiple Interface Declaration
TC：Topology Control

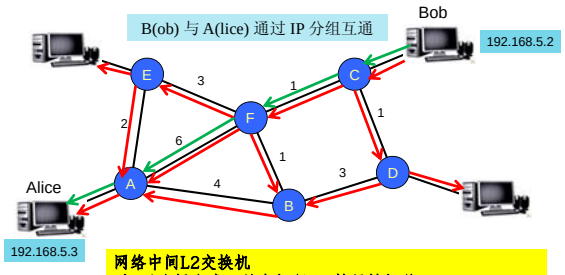
第五章 寻址与路由技术

5.1 距离矢量路由
5.2 链路状态路由
5.3 MAC广播生成树
5.4 路由表查找算法
5.5 标签交换路由



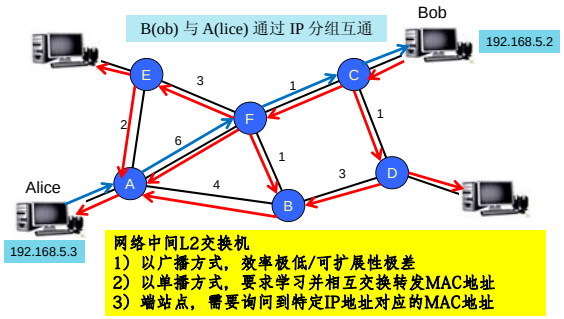
局域网广播风暴

L2单播与广播的效率对比



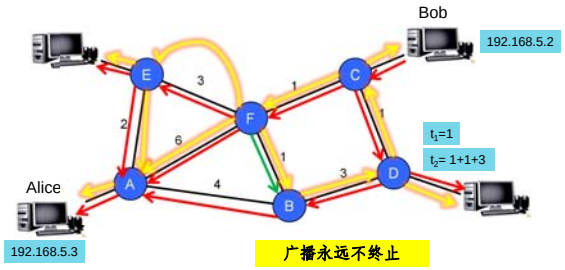
网络中间L2交换机
1）以广播方式，效率极低/可扩展性极差

广播以获取L2单播地址



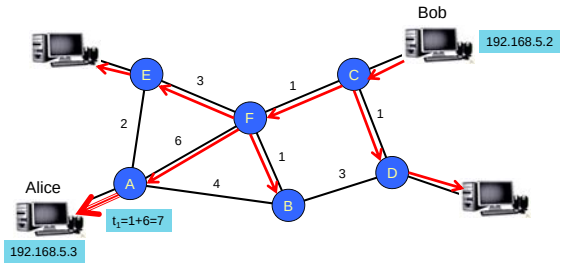
网络中间L2交换机
1）以广播方式，效率极低/可扩展性极差
2）以单播方式，要求学习并相互交换转发MAC地址
3）端站点，需要询问到特定IP地址对应的MAC地址

L2风暴路径，以(F,B)为例

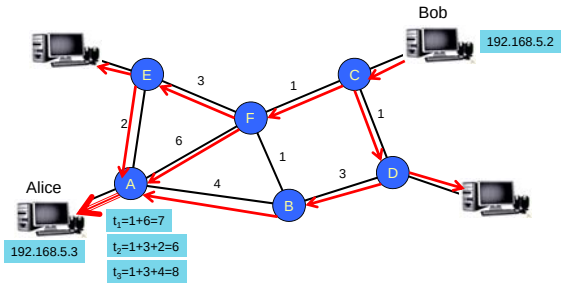


广播永远不终止

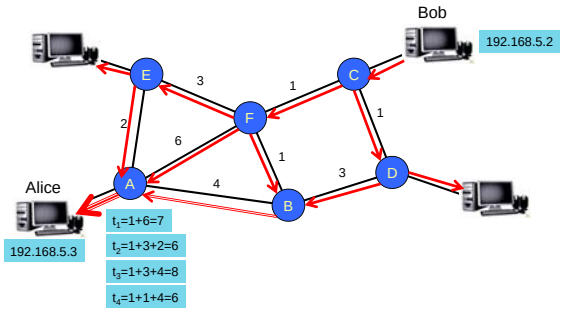
重复广播的时间间隔



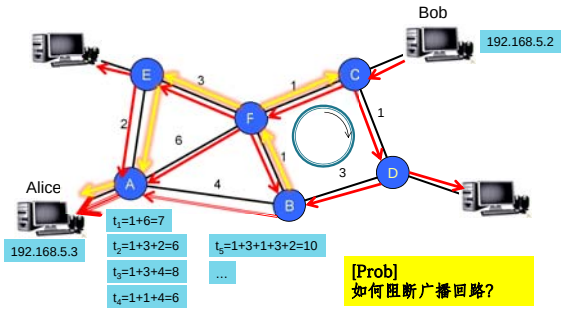
重复广播的时间间隔



重复广播的时间间隔



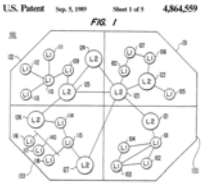
重复广播的时间间隔

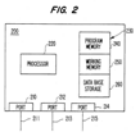


[Prob]
如何阻断广播回路?

STP/RSTP

Radia Perlman算法



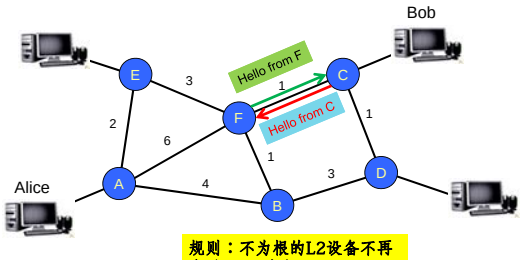


1) L2之间通过HELLO消息
   通告ID、成本和链路ID
2) 选取根
   ID最小者自动选出
3) 选取根端口
   L2至根成本最小的端口
4) 确定指定端口
   L1经该口至根最小的端口

1951－

Perlman R. An algorithm for distributed computation of a spanning tree in an extended LAN[C]. ACM SIGCOMM Computer Communication Review, 1985, 15(4): 44-53.

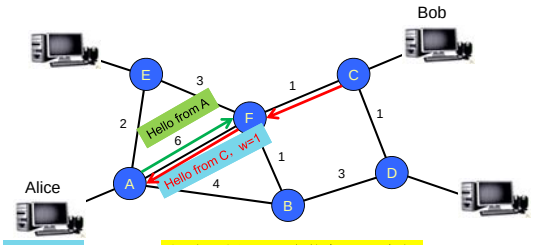
STP操作示例，C可能为根



规则：不为根的L2设备不再发送Hello消息

所以，F不以根发送Hello

STP操作示例，C不能为根

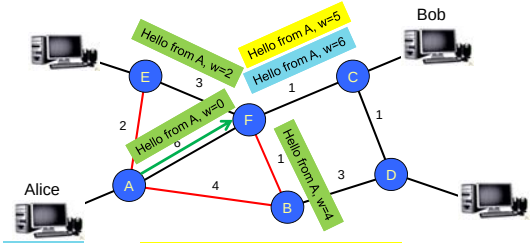


192.168.5.3

规则：非根L2设备转发Hello消息，并附加至根的成本

A作为根发出的Hello被F转发给C

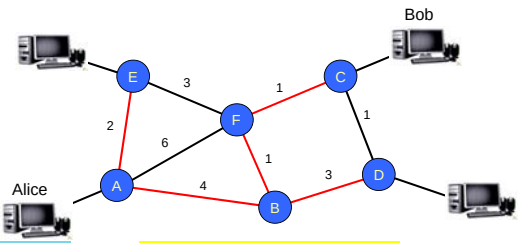
STP操作示例，成本最低端口



192.168.5.3

对于F：
w(A,E,F)=2+3=5
w(A,F)=6
w(A,B,F)=5
相同成本，以ID最小，即B，为父

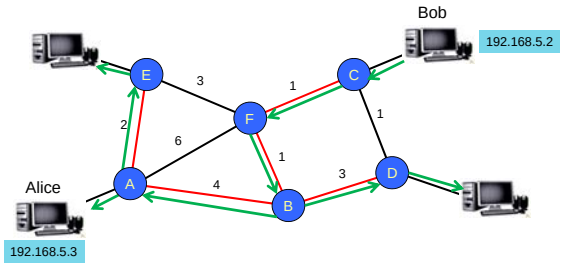
STP操作示例，成本最低端口



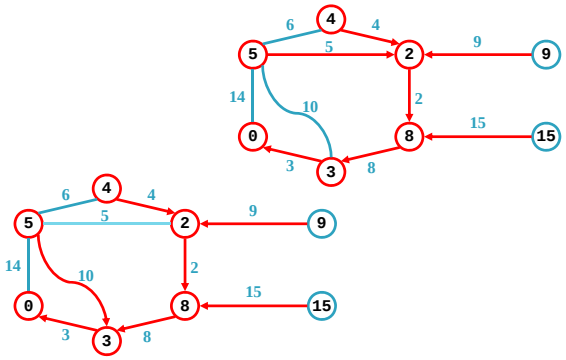
192.168.5.3

对于B：
(B,F)，(B,D)为指定端口
对于F：
(F,C)为指定端口

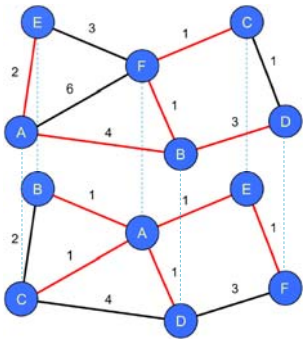
STP的广播路径



与Ch4.Prim算法对比



MSTP实现负载均衡



MSTP(0)
如果(A,E)和(A,B)链路中断

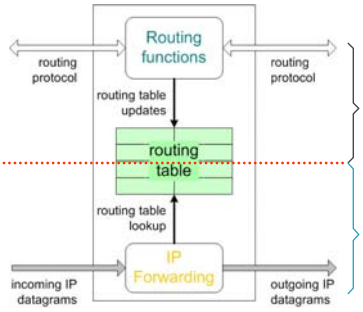
业务流临时转移到
MSPT(1)

第五章 寻址与路由技术

- 5.1 距离矢量路由
- 5.2 链路状态路由
- 5.3 MAC广播生成树
- 5.4 路由表查找算法
- 5.5 标签交换路由

路由表查找的问题

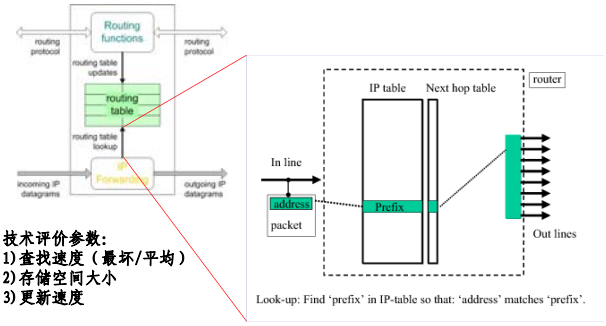
寻址与路由的关系



控制平面:
运行路由协议
(RIP, OSPF, BGP)

数据平面:
将入口分组寻址转发到出口

路由表查找的功能

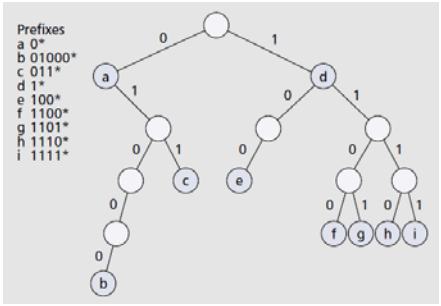


技术评价参数:
1)查找速度（最坏/平均）
2)存储空间大小
3)更新速度

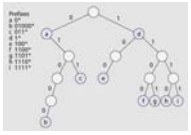
二叉树查找

根节点度不大于2，
中间节点度不大于3，
叶节点度为1。

二叉构结构，查找和更新均为O(W)



节点结构及树构造C示例

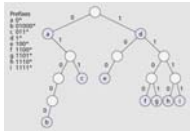


```
struct Node {
   Node  *left;
   Node  *right;
   int32 portIndex;
} **root;
```

```
struct IPPref {
   int8 len;
   int32 addr;
};
```

```
Insert(Node **r, int port, IPPref pref, int of = 0) {
   if (*r == NULL) *r = new Node(0,0,-1);
   if (offset >= pref.len)
      (*r)->portIndex = port;
   else {
      if ( (pref.addr << of) >> (32-of) )
           r = &((*r)->right);
      else
           r = &((*r)->left);
      Insert(r, port, pref, ++of);
   }
}
```


查找操作

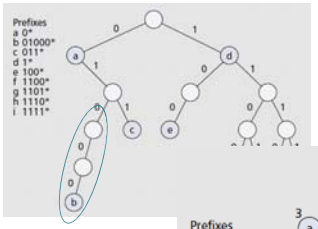


```
struct Node {
   Node  *left;
   Node  *right;
   int32 portIndex;
} **root;
```

```
struct IPPref {
   int8 len;
   int32 addr;
};
```

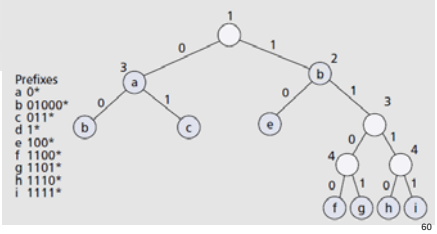
```
int Lookup(Node **r, IPAddr addr, int of = 0) {
   int ret = -1;
   if ( (addr<<of)>>(32-of) && (*r)->right)
      ret = Lookup(&((*r)->right), addr, ++of);
   if (! (addr<<of)>>(32-of) && (*r)->left)
      ret = Lookup(&((*r)->left), addr, ++of);
   if (ret == -1) ret = (*r)->portIndex;
   return ret;
}
```


路压缩二叉树，PATRICIA

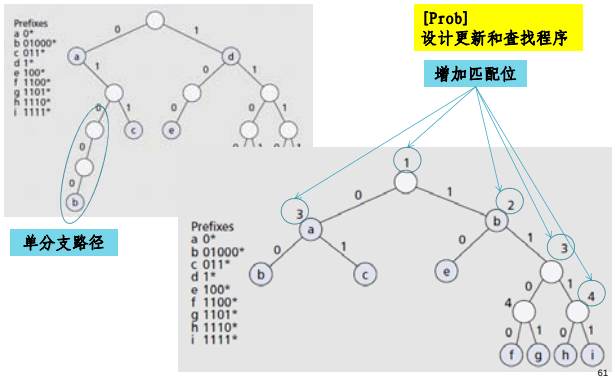


单分支路径

平衡二叉树:
除叶和根外，
节点度均为3



路压缩二叉树，增加比特位指示

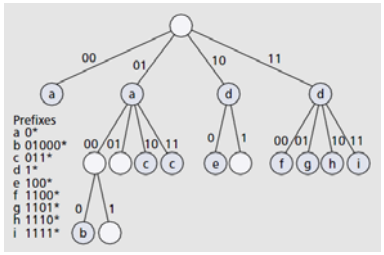


[Prob]
设计更新和查找程序

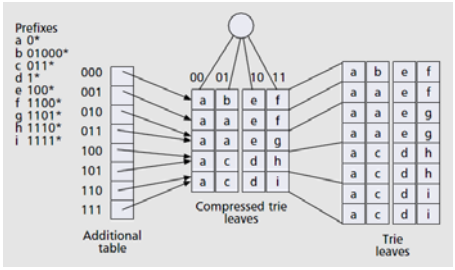
增加匹配位

单分支路径

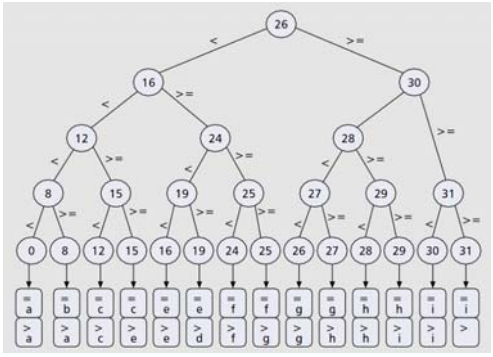
多比特树



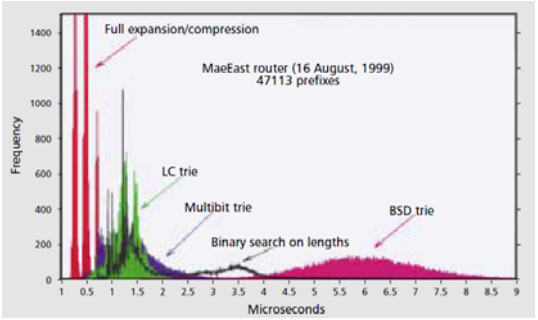
先扩充再并行压缩树



前缀长度分区树

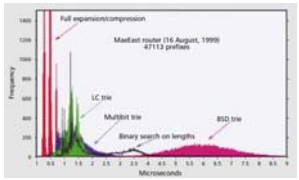


仿真实验结果



Ruiz-Sánchez M Á, Biersack E W, Dabbous W. Survey and taxonomy of IP address lookup algorithms[J]. Network, IEEE, 2001, 15(2): 8-23.

能否满足高吞吐性能



平均分组查找时间T
最长分组长度L = 1500B = 12000b

吞吐能性：L/T

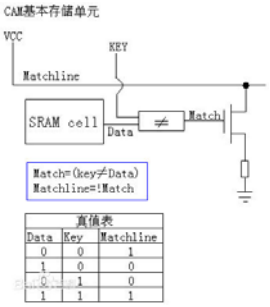
T(us)	S(Mb/s)
6	2000
3.5	3429
1.5	8000
0.5	24000

路由器端口数：N~100
端口宽带：BW~10Gb/s
最低吞吐性能：S~1000Gb/s

[Prob]
如何缩小缺口

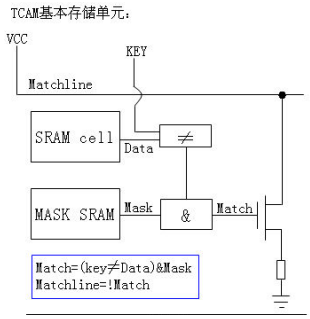
基于硬件的查找

内容寻址存储器(CAM)



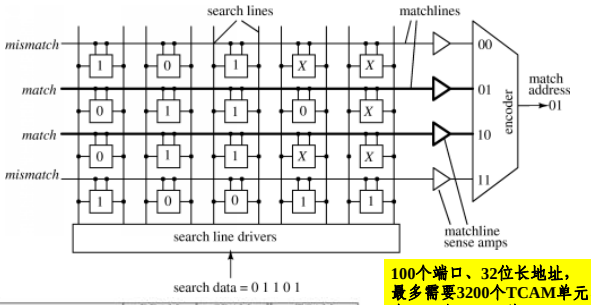
真值表		
Data	Key	Matchline
0	0	1
1	0	0
0	1	0
1	1	1

三态CAM(TCAM)



真值表				
Key	Mask	Data	Actual Data	Matchline
0	0	0	Don't care	1
0	0	1	Don't care	1
0	1	0	0	1
0	1	1	1	0
1	0	0	Don't care	1
1	0	1	Don't care	1
1	1	0	0	0
1	1	1	1	1

5位长地址、4出口的TCAM逻辑



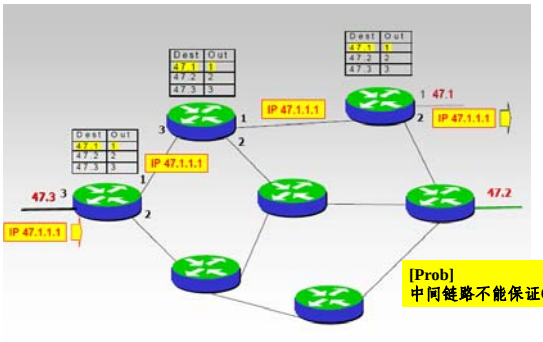
	DRAM	SRAM	TCAM
Wire speed	20 ns	1 ns	360 Msps
Read latency	20 ns	2 ns	75 ns
Cost per 10 Mbit	$1	$10	$50

100个端口、32位长地址，最多需要3200个TCAM单元，含6400个SRAM单元

第五章 寻址与路由技术

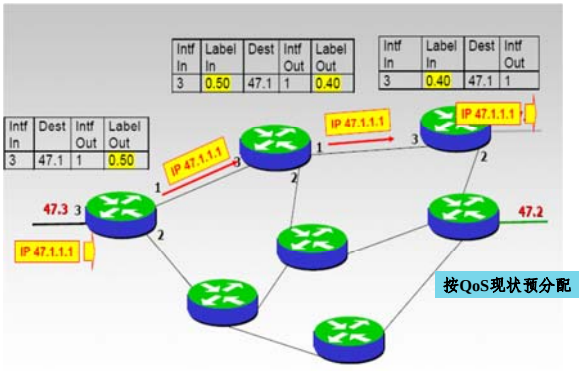
多协议标签交换技术

IP路由选择方式



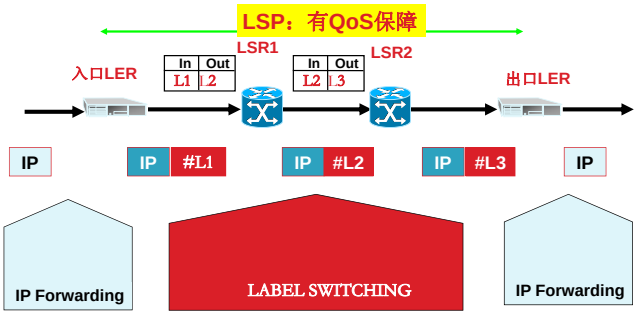
[Prob]
中间链路不能保证QoS

MPLS转发



按QoS现状预分配

MPLS的系统组成



ROUTE AT EDGE, SWITCH IN CORE

QoS参数

路径$Pa(s,t)$，成本c，测度$m \propto M$

$$c_m Pa(s,t) = f(m(v_s, v_{s+1}), \ldots, m(v_i, v_{i+1}), \ldots, m(v_{t-1}, v_t))$$ 一般式

$$c_{bw} Pa(s,d) = \min_{(i,j) \in Pa(s,d)} b(i,j) \quad < bandwidth >$$

$$c_{bf} Pa(s,d) = \min_{(i,j) \in Pa(s,d)} bf(i,j) \quad < buffer >$$

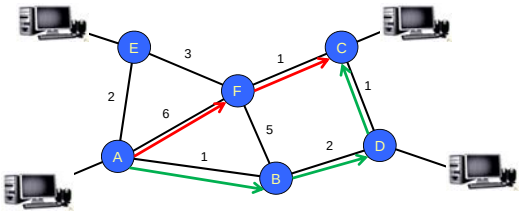
$$c_h Pa(s,d) = \#edges\ of\ Pa(s,d) \quad < hop\ count >$$

$$c_d Pa(s,d) = \sum_{(i,j) \in Pa(s,d)} d(i,j) \quad < delay >$$

$$c_j Pa(s,d) = \sum_{(i,j) \in Pa(s,d)} j(i,j) \quad < jitter >$$

$$c_l Pa(s,d) = 1 - \prod_{(i,j) \in Pa(s,d)} [1 - l(i,j)] \quad < packet\ loss >$$

最小跳（红）、最小延时（绿）

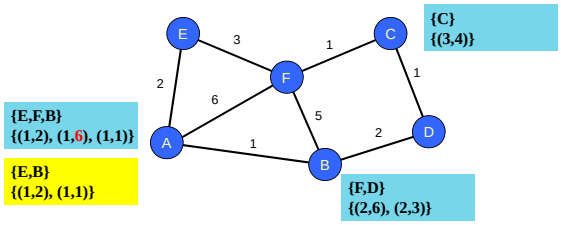


Constrained Shortest Path算法

测度m <M、跳数最小路径

Delay, m<=5

(3,4): （跳数，m）



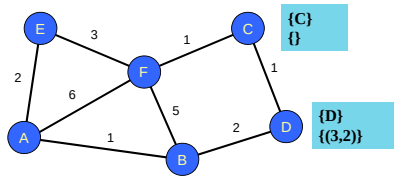
{C}
{(3,4)}

{E,F,B}
{(1,2), (1,6), (1,1)}

{E,B}
{(1,2), (1,1)}

{F,D}
{(2,6), (2,3)}

BW, m>=2



{C}
{}

{E,F,B}
{(1,2), (1,6), (1,1)}

{D}
{(3,2)}

{E,F}
{(1,2), (1,6)}

{B}
{(2,5)}

第五章 寻址与路由技术

5.1 对比分析RIP与OSPF协议的异同点。

5.2 如何解决LS路由的无究计数问题？

5.3 实验观测并报告交换机自环前后的网络流量变化。

5.4 设计PATRICIA路由表的更新和查找代码。

5.5 给出Bellman-Ford算法的QoS约束改进思路。

Cavendish D, Gerla M. Internet QoS routing using the Bellman-Ford algorithm[C]//High Performance Networking. Springer US, 1998: 627-646.