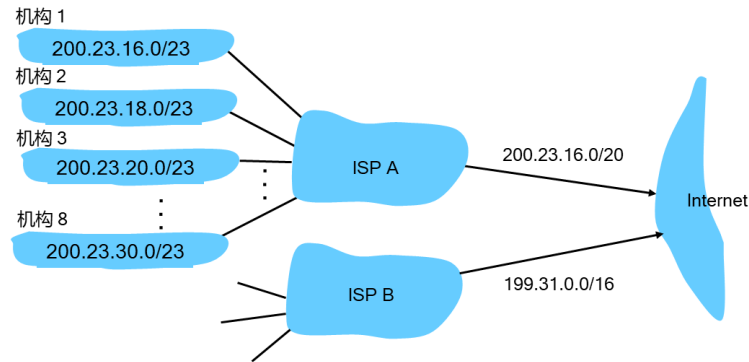


CIDR路由聚合

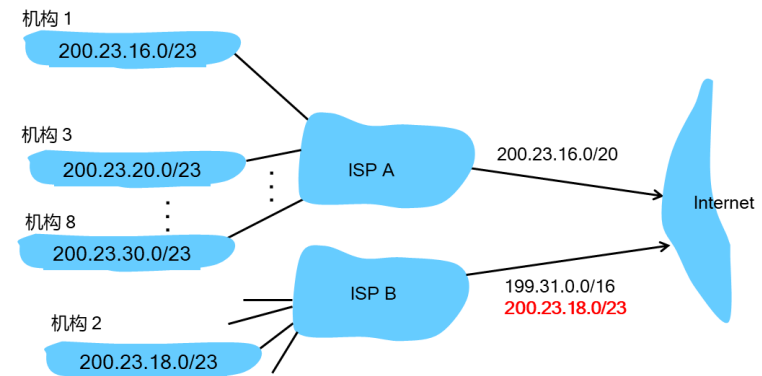


2022/11/25

计算机网络与信息安全研究室

36

CIDR路由聚合 (续)

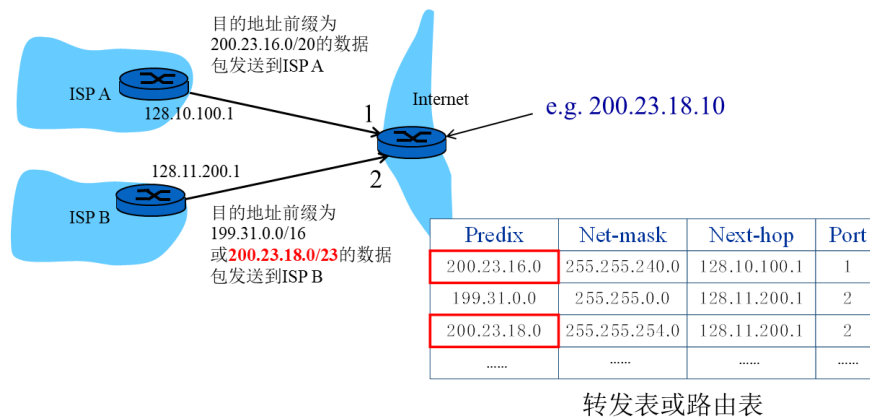


2022/11/25

计算机网络与信息安全研究室

37

最长匹配原则



2022/11/25

计算机网络与信息安全研究室

38

单选题 5分

有地址块202.113.16.0~202.113.31.255（16个C类地址），聚合成的地址和掩码为：

- A 202.113.16.0（掩码：255.255.255.0）
- B 202.113.16.0（掩码：255.255.240.0）
- C 202.113.16.0（掩码：255.255.32.0）
- D 202.113.16.0（掩码：255.255.16.0）

2022/11/25

计算机网络与信息安全研究室

39



1	202.113.00010000	00000000	202.113.00010000	11111111
2	202.113.00010001	00000000	202.113.00010001	11111111
3	202.113.00010010	00000000	202.113.00010010	11111111
4	202.113.00010011	00000000	202.113.00010011	11111111
5	202.113.00010100	00000000	202.113.00010100	11111111
6	202.113.00010101	00000000	202.113.00010101	11111111
7	202.113.00010110	00000000	202.113.00010110	11111111
8	202.113.00010111	00000000	202.113.00010111	11111111
9	202.113.00011000	00000000	202.113.00011000	11111111
10	202.113.00011001	00000000	202.113.00011001	11111111
11	202.113.00011010	00000000	202.113.00011010	11111111
12	202.113.00011011	00000000	202.113.00011011	11111111
13	202.113.00011100	00000000	202.113.00011100	11111111
14	202.113.00011101	00000000	202.113.00011101	11111111
15	202.113.00011110	00000000	202.113.00011110	11111111
16	202.113.00011111	00000000	202.113.00011111	11111111

$2^8 \times 2^4 = 2^{12}$ 255.255.11110000.00000000

2022/11/25

计算机网络与信息安全研究室

40

40

单选题 5分

有地址块202.113.224.0-202.113.255.255（32个C类地址），聚合成的地址和掩码为：

- ☐ A 202.113.224.0（掩码：255.255.255.0）
- ☐ B 202.113.224.0（掩码：255.255.240.0）
- ☒ C 202.113.224.0（掩码：255.255.224.0）
- ☐ D 202.113.224.0（掩码：255.255.32.0）

2022/11/25

计算机网络与信息安全研究室

41



多选题 5分

有地址块222.30.32.0-222.30.61.255（30个C类地址），聚合成的地址和前缀长度包括：

- ☒ A 222.30.32.0/20
- ☒ B 222.30.48.0/21
- ☒ C 222.30.56.0/22
- ☒ D 222.30.60.0/23

2022/11/25

计算机网络与信息安全研究室

42

42

222.30.00100000.0	222.30.00110000.0
222.30.00100001.0	222.30.00110001.0
222.30.00100010.0	222.30.00110010.0
222.30.00100011.0	222.30.00110011.0
222.30.00100100.0	222.30.00110100.0
222.30.00100101.0	222.30.00110101.0
222.30.00100110.0	222.30.00110110.0
222.30.00100111.0	222.30.00110111.0
222.30.00101000.0	222.30.00111000.0
222.30.00101001.0	222.30.00111001.0
222.30.00101010.0	222.30.00111010.0
222.30.00101011.0	222.30.00111011.0
222.30.00101100.0	222.30.00111100.0
222.30.00101101.0	222.30.00111101.0
222.30.00101110.0	
222.30.00101111.0	

如果用222.30.32.0/19，需要排除哪个地址块？

2022/11/25

计算机网络与信息安全研究室

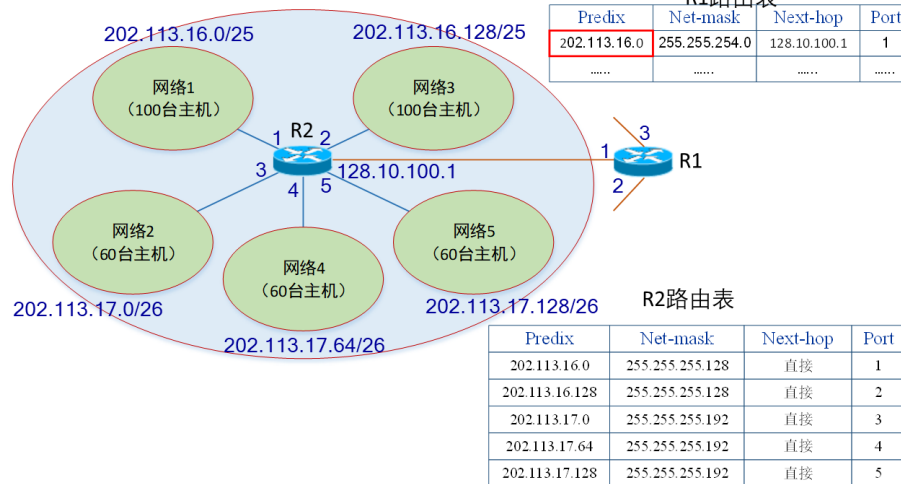
43

4.2 IP协议—IPv4地址问题及解决策略



地址分配示例

例如：分配202.113.16.0/23地址块



2022/11/25

计算机网络与信息安全研究室

44

4.2 IP协议—IPv4地址问题及解决策略



私有IP地址与NAT

■ 私有IP地址的使用

- ▶ 可以在私有网络（本地网络）中自由使用，但不能出现在公网上
- ▶ 如果数据包要进入公网，需要进行私有地址到公有地址的转换
- ▶ 承担地址转换的设备称为**NAT**（Network Address Translation）

■ 本地网络使用私有IP地址，**NAT**面向公网侧使用公网IP地址

- ▶ 只需要一个或少量的公网IP地址：**解决地址短缺**
- ▶ 公网IP地址变化，不影响本地网络地址配置
- ▶ 安全性增强：隐藏本地网络的细节

地址范围	网络个数	地址类别
10.0.0.0~10.255.255.255	1	A类
172.16.0.0~172.31.255.255	16	B类
192.168.0.0~192.168.255.255	256	C类

2022/11/25

计算机网络与信息安全研究室

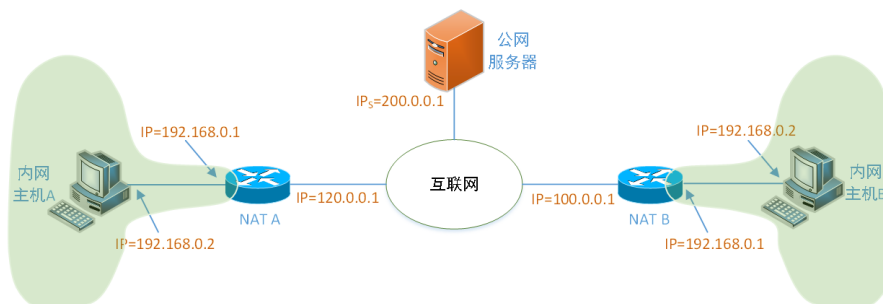
45

4.2 IP协议—IPv4地址问题及解决策略



网络地址转换（NAT）

■ NAT功能：实现私有IP地址与公网IP地址之间的转换



2022/11/25

计算机网络与信息安全研究室

46

4.2 IP协议—IPv4地址问题及解决策略



网络地址转换（NAT）

NAPT: Network Address/Port Translation

■ NAT的基本实现方法

- ▶ **离开本地网络的数据包**：用（**NAT的IP地址+新端口号**）代替（源IP地址+端口号），远端的主机返回的数据包将使用（**NAT的IP地址+新端口号**）作为目的IP地址和目的端口
 - NAT在**转换表**中记录（**NAT的IP地址+新端口号**）到（**源IP地址+端口号**）的映射
- ▶ **进入本地网络的数据包**：用转换表中的（**源IP地址+端口号**）替换数据包中的目的IP地址和端口号（**NAT的IP地址+新端口号**）

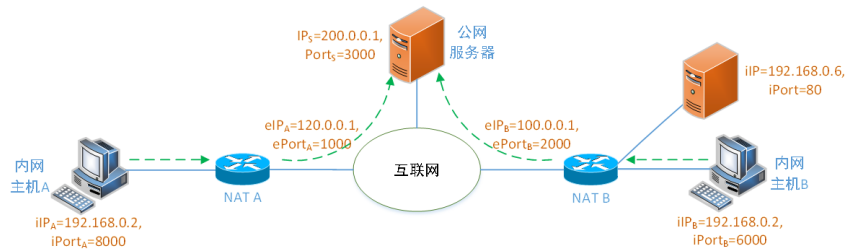
2022/11/25

计算机网络与信息安全研究室

47

网络地址转换 (NAT)

■ NAT地址转换与NAT地址转换表



地址转换表 (NAT B)

外侧IP	外侧端口	内部IP	内部端口
100.0.0.1	2000	192.168.0.2	6000

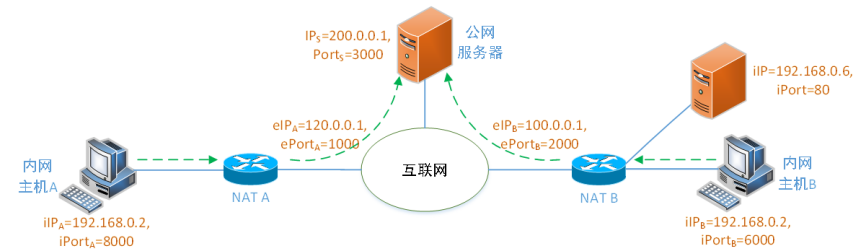
2022/11/25

计算机网络与信息安全研究室

48

网络地址转换 (NAT)

■ NAT地址转换与NAT地址转换表



地址转换表 (NAT B)

外侧IP	外侧端口	内部IP	内部端口
100.0.0.1	2000	192.168.0.2	6000
100.0.0.1	80	192.168.0.6	80

2022/11/25

计算机网络与信息安全研究室

49

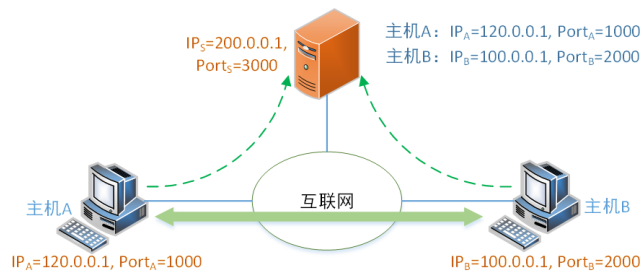
网络地址转换 (NAT)

■ NAT问题讨论

▶ 破坏端到端原则，应用部署困难

— 例如：服务器辅助的对等通信

- ✓ 地址端口代换、影响网络性能
- ✓ 安全溯源、网络监管困难



2022/11/25

计算机网络与信息安全研究室

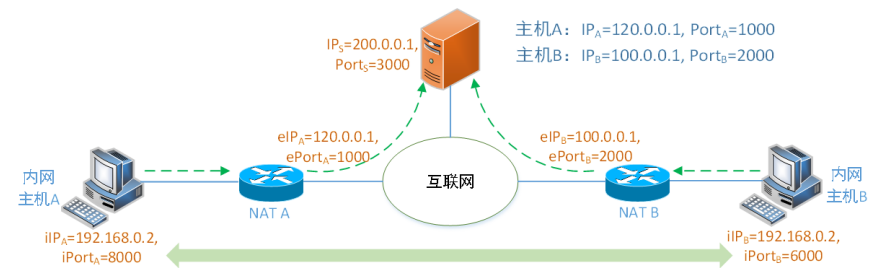
50

网络地址转换 (NAT)

■ NAT问题讨论

▶ 破坏端到端原则，应用部署困难

— 例如：服务器辅助的对等通信



讨论问题：考虑服务器辅助的对等通信，工作流程如何？

2022/11/25

计算机网络与信息安全研究室

51



IPv4局限性

■ IPv4地址资源枯竭

- 2³²个地址，2011年2月3号，IANA宣布IPv4地址耗尽

■ 路由成为互联网的瓶颈

- 网络数目增加导致路由表庞大
- 地址层次性差，查表时间长
- 数据包首部长度可变，不利于路由转发处理

■ 缺乏服务质量保证

- 服务质量欠缺：主要依赖IP报头中的“服务类型”字段
- 安全性缺乏：IPSec只是IPv4的一个选项

■ 配置较为繁琐

2022/11/25

计算机网络与信息安全研究室

52



IPv4的改进措施

■ 无类型域间选路（CIDR）

- CIDR一般采用13~27位可变前缀
- 容纳主机范围：32~500000台
- 充分利用IPv4地址，减缓IPv4地址的消耗速度
- 实现路由聚合，缓解主干路由表的激增问题

■ 网络地址转换（NAT）

- 私有地址和公有地址转换，“增加”可用的地址空间
- 有效地解决了IPv4地址紧张状况和匮乏危机
- **问题**：破坏了端到端特性，影响点对点应用，安全溯源困难，网络应用性能下降等等

2022/11/25

计算机网络与信息安全研究室

53



IPv6互联网的优势

■ 解决地址耗尽问题：更大的地址空间（32位→128位）

- 340,282,366,920,938,463,463,374,607,431,768,211,456个IP地址
- 地球上每平方米范围： 6.67×10^{23} 个IP地址

■ 自动配置的支持（即插即用）

- 无状态地址自动配置
- 有状态地址自动配置：对DHCP协议改进和扩展，网络管理更加方便和快捷

■ 改善网络性能

- IP包头的合理改善，提高了路由器对数据包的处理速率
- 路由聚合，使路由表更小，提高转发效率

■ 方便各项业务开展

- 无需使用NAT设备

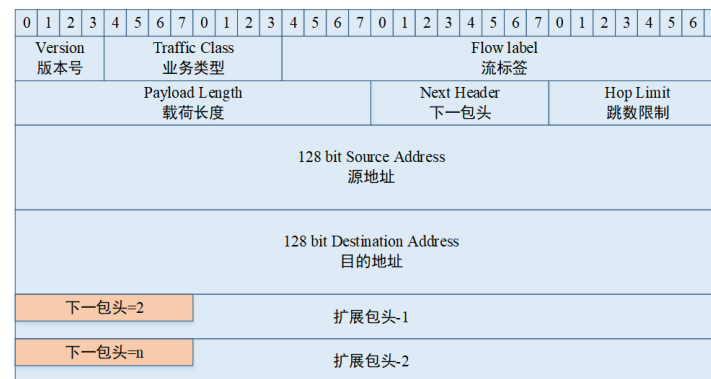
2022/11/25

计算机网络与信息安全研究室

54



IPv6数据包格式（RFC 2460）



■ IPv6数据包由一个IPv6基本头、多个扩展头和上层数据单元组成

■ IPv6基本头：40字节固定长度，包含了发送和转发该数据包必须处理的一些字段

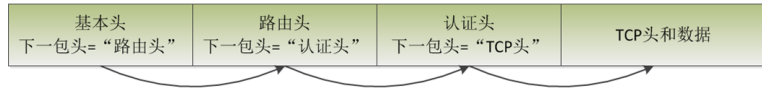
2022/11/25

计算机网络与信息安全研究室

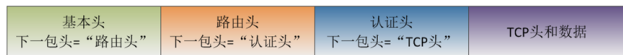
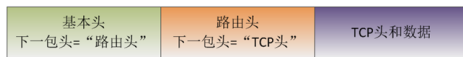
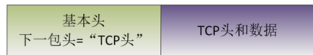
55

IPv6扩展头

- IPv6数据包可以包含0个或多个扩展头，扩展头位于基本头之后
- 基本头和扩展头中“下一包头”字段指出下一个扩展头的类型。最后一个扩展头中的“下一包头”字段指出高层协议的类型



IPv6数据包示例



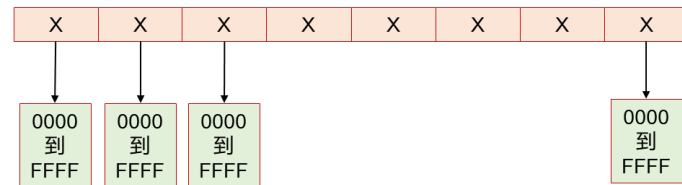
2022/11/25

计算机网络与信息安全研究室

56

IPv6地址的表示方法

- 128位地址：由冒号分开的8组十六进制字段组成



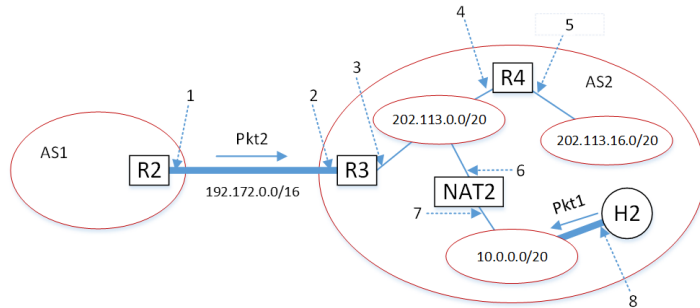
- 完全形式：1080:0000:0000:0000:0008:0800:200C:417A
- 压缩形式：1080:0:0:0:8:800:200C:417A / 1080::8:800:200C:417A
 - 地址段中有时会出现连续的几组0，这时这些0可以用“::”代替，但一个地址中只能出现一次“::”，例如：FF01:0:0:0:0:0:0:101=FF01::101, 0:0:0:0:0:0:1::1
- 内嵌IPv4地址的IPv6地址：0:0:0:0:0:0:61.1.133.1 或::61.1.133.1
- URL的IPv6地址表示：[https://\[2001:410:0:1:250:fcee:e450:33ab\]:8443/abc.html](https://[2001:410:0:1:250:fcee:e450:33ab]:8443/abc.html)

2022/11/25

计算机网络与信息安全研究室

57

4.2 IP协议—练习



- 问题1：端口4、5、7应该分别分配什么样的IP地址？
- 问题2：如果路由到AS2中的所有网络，R2路由表中应至少包含什么样的路由信息？
- 问题3：如果Pkt2是发往H2的数据包，其目的IP地址是什么？

2022/11/25

计算机网络与信息安全研究室

58

4.3 ICMP协议

ICMP: 互联网控制消息协议

类型	代码	描述
0	0	回声请求 (Ping命令)
3	0	目的网络不可达
3	1	目的主机不可达
3	2	目的协议不可达
3	3	目的端口不可达
3	6	目的网络不可知
3	7	目的主机不可知
8	0	回声应答 (Ping命令)
9	0	路由通告
10	0	路由发现
11	0	TTL超期
12	0	IP数据包首部出错

2022/11/25

计算机网络与信息安全研究室

59

ICMP示例: Traceroute (Tracert)

```

命令提示符

通过最多 30 个跃点跟踪
到 www.a.shifen.com [110.242.68.4] 的路由:

 1    1 ms    1 ms    1 ms    192.168.3.1
 2    3 ms    2 ms    2 ms    192.168.18.1
 3    6 ms    4 ms    4 ms    111.165.48.1
 4    6 ms    5 ms    5 ms    117.8.5.9
 5    *      *      *      请求超时。
 6    6 ms    6 ms    6 ms    219.158.107.17
 7    11 ms   10 ms   10 ms   110.242.66.166
 8    14 ms   10 ms   10 ms   221.194.45.130
 9    *      *      *      请求超时。
10    *      *      *      请求超时。
11    *      *      *      请求超时。
12   11 ms   10 ms   10 ms   110.242.68.4

跟踪完成。

```

2022/11/25

计算机网络与信息安全研究室

60

路由算法概述

- **目标**: 确定从源主机到目的主机的最优路径
- **最优路径**: 一般为**代价最小**的路径
 - ▶ 代价: 有不同的定义方法, 例如: 距离、时延、费用、拥塞等
- **路由算法的分布式实现**
 - ▶ 路由器之间交互**路由信息**或**链路状态信息**
 - ▶ 每台路由器独立计算最优路径

注: 相对于数据包转发, 所有路由协议的开销都属于额外管理开销

2022/11/25

计算机网络与信息安全研究室

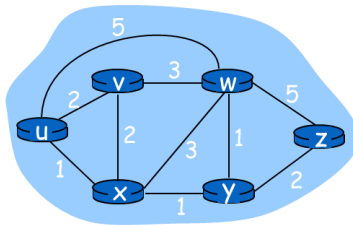
61

4.4 路由算法

路由算法概述 (续)

■ 网络结构的图抽象

- ▶ **路由器**: 图中的点
- ▶ **物理链路**: 图中的边
- ▶ **代价**: 图中边的权值

■ 用 $G = (N, E)$ 表示图

- ▶ $N = \{u, v, w, x, y, z\}$, 路由器集合
- ▶ $E = \{(u, v), (u, x), (v, x), (v, w), (x, w), (x, y), (w, y), (w, z), (y, z)\}$, 链路的集合

2022/11/25

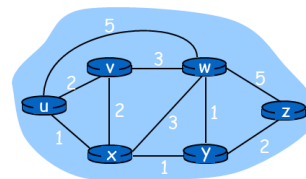
计算机网络与信息安全研究室

62

4.4 路由算法

路由算法概述 (续)

- **链路代价**: 相邻节点之间的代价值, 表示为 $c(x_i, x_j)$
 - ▶ 如果 x_i 和 x_j 不相邻, 链路代价记为无穷 ∞
 - ▶ 例如, $c(w, z) = 5$
- **路径代价**: 路径上所有链路代价和, 例如, 路径 $(x_1, x_2, x_3, \dots, x_p)$ 的代价 $= c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$



2022/11/25

计算机网络与信息安全研究室

63

链路状态路由算法

- 假设网络中的所有节点都已知网络拓扑和链路代价
- 各个节点利用Dijkstra算法计算最小代价路径
- 根据计算结果产生路由表

符号定义:

- $c(i,j)$: 节点i到节点j的链路代价, 不相邻节点之间链路代价为无穷
- $D(k)$: 从计算节点到目的节点k当前路径代价
- $p(k)$: 从计算节点到目的节点k的路径中k节点的前继节点
- N : 最小代价路径已知的节点集合

难点: 网络拓扑和链路代价如何获取?

2022/11/25

计算机网络与信息安全研究室

64

Dijkstra算法

计算复杂性: $O(n^2) \rightarrow O(n \log n)$

```

1 Initialization:
2  $N = \{u\}$  //计算u节点到所有其他节点的最优路径
3 for all nodes k
4   if k adjacent to u
5     then  $D(k) = c(u,k)$ 
6     else  $D(k) = \infty$ 
7
8 Loop
9   find m not in N such that  $D(m)$  is a minimum
10  add m to N
11  update  $D(k)$  for all k adjacent to m and not in N:
12     $D(k) = \min(D(k), D(m) + c(m,k))$ 
13    /* new cost to k is either old cost to k or known
14       shortest path cost to m plus cost from m to k */
15 until all nodes in N
  
```

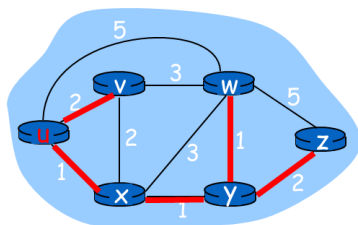
2022/11/25

计算机网络与信息安全研究室

65

Dijkstra算法示例

步骤	集合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					



目的	下一跳	代价
v	v	2
w	x	3
x	x	1
y	x	2
z	x	4

2022/11/25

计算机网络与信息安全研究室

66

Bellman-Ford公式

- 假设: $D_x(y)$ = 从x到y最小代价路径的代价值

- 则: $D_x(y) = \min \{c(x,m) + D_m(y)\}$

► m为x的邻居

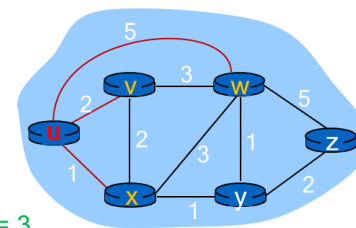
- 示例:

► 已知: $D_v(z) = 5, D_x(z) = 3, D_w(z) = 3$

► $D_u(z) = \min \{c(u,v) + D_v(z), c(u,x) + D_x(z), c(u,w) + D_w(z)\}$

$= \min \{2 + 5, 1 + 3, 5 + 3\}$

$= 4$



2022/11/25

计算机网络与信息安全研究室

67

距离向量路由算法

■ 符号定义

- ▶ $D_x(y)$ = 从x到y最小代价路径的代价值
- ▶ 节点x可以获知到每个邻居的链路代价 $c(x, m)$
- ▶ 节点x维护自己的距离向量 $D_x = [D_x(y): y \in N]$
- ▶ 节点x维护其邻居的距离向量, 对于每个邻居m, x维护

$$D_m = [D_m(y): y \in N]$$

2022/11/25

计算机网络与信息安全研究室

68

距离向量路由算法

■ 基本思想

- ▶ 当节点的距离向量值发生变化时, 向邻居节点发送自己的距离向量
- ▶ 如果节点x接收到邻居节点发送的距离向量, 使用Bellman-Ford公式重新计算到所有其他节点的路径代价

$$D_x(y) \leftarrow \min_m \{c(x, m) + D_m(y)\} \quad \text{每个节点 } y \in N$$

- ▶ 经过反复迭代, $D_x(y)$ 会逐渐收敛到实际的最小路径代价值

2022/11/25

计算机网络与信息安全研究室

69

距离向量路由算法

对于节点 x:

```

01 Initialization:
02   for all destinations y in N:
03      $D_x(y) = c(x, y)$       /* if y is not a neighbor then  $c(x, y) = \infty$  */
04   for each neighbour m
05      $D_m(y) = \infty$  for all destinations y in N
06   for each neighbour m
07     send distance vector  $D_x = [D_x(y): y \in N]$  to m
08 loop
09   wait (until I see a link cost change to some neighbor m
10        or until I receive a distance vector from some neighbor w)
11   for each y in N
12      $D_x(y) = \min_m \{c(x, m) + D_m(y)\}$ 
13   if  $D_x(y)$  changed for any destination y
16     send distance vector  $D_x = [D_x(y): y \in N]$  to all neighbors
17 forever
  
```

2022/11/25

计算机网络与信息安全研究室

70

node x table

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

node y table

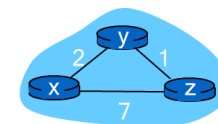
		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

node z table

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

$$D_x(y) = \min\{c(x, y) + D_y(y), c(x, z) + D_z(y)\} = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x, y) + D_y(z), c(x, z) + D_z(z)\} = \min\{2+1, 7+0\} = 3$$



2022/11/25

计算机网络与信息安全研究室

71

4.4 路由算法—距离向量算法



node x table

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

node y table

		cost to		
		x	y	z
from	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

node z table

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

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

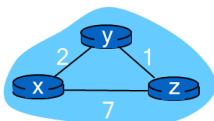
		cost to		
		x	y	z
from	x	0	2	7
	y	2	0	1
	z	7	1	0

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

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

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

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



2022/11/25

计算机网络与信息安全研究室

72

4.4 路由算法—距离向量算法

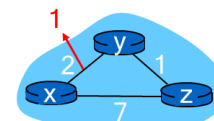


距离向量路由算法—链路代价改变示例

注：节点可以检测到链路代价的改变

- t_0 时刻， y 检测到链路代价的改变，计算并修改自己的DV，通知邻居 x 和 z
- t_1 时刻， z 接收到来自 y 的DV，计算并修改自己的DV，通知邻居 x 和 y
- t_2 时刻， y 接收到来自 z 的更新，并重新计算自己的DV， y 的最小代价值未发生变化，不再向 z 发送DV

“好消息传播的快”



2022/11/25

计算机网络与信息安全研究室

73