

# 传输层概述

---

# 传输层的作用

- 网络层提供点到点的连接
- 传输层提供端到端的连接



# 传输层的协议

- TCP ( Transmission Control Protocol )

- 传输控制协议
- 可靠的、面向连接的协议
- 传输效率低

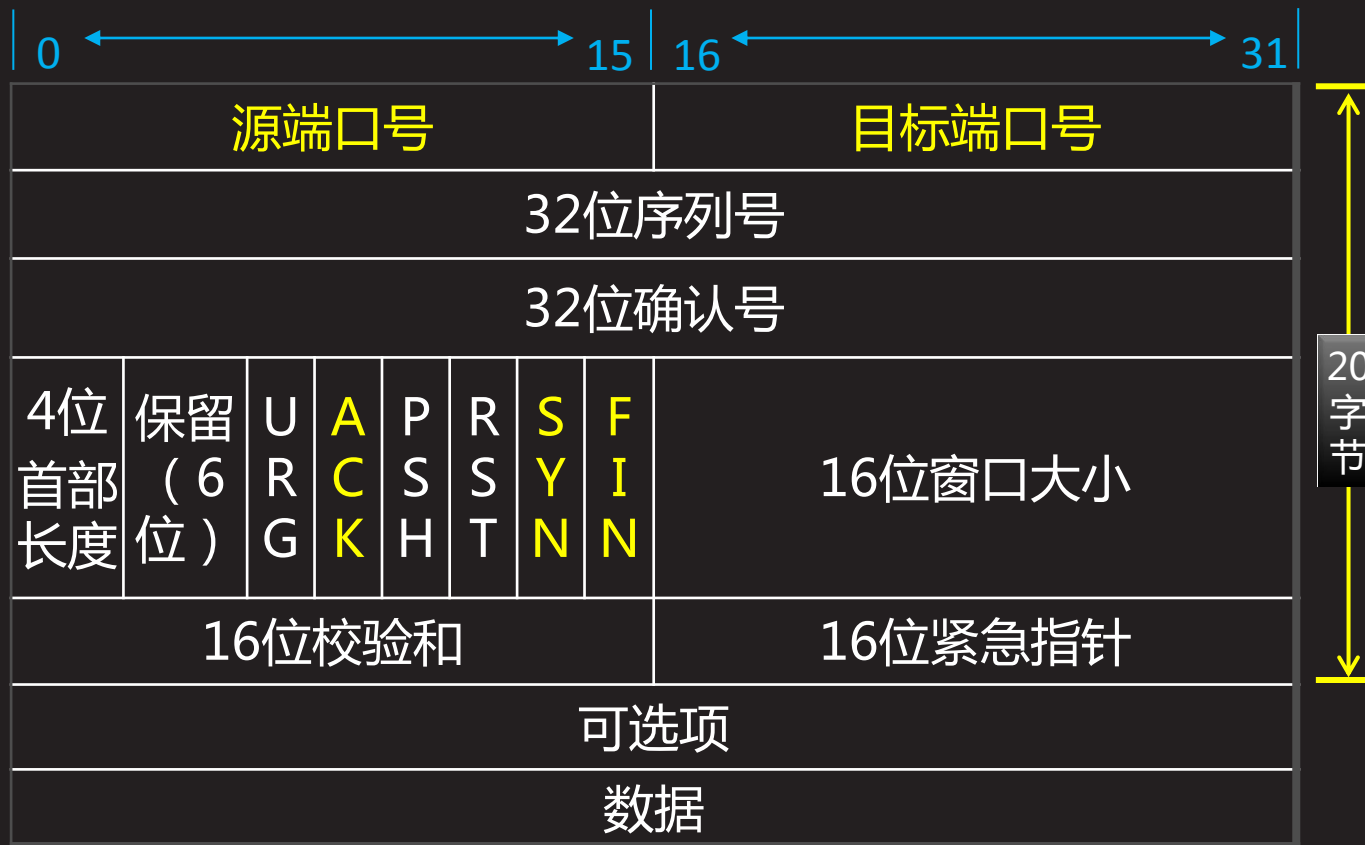


- UDP ( User Datagram Protocol )

- 用户数据报协议
- 不可靠的、无连接的服务
- 传输效率高

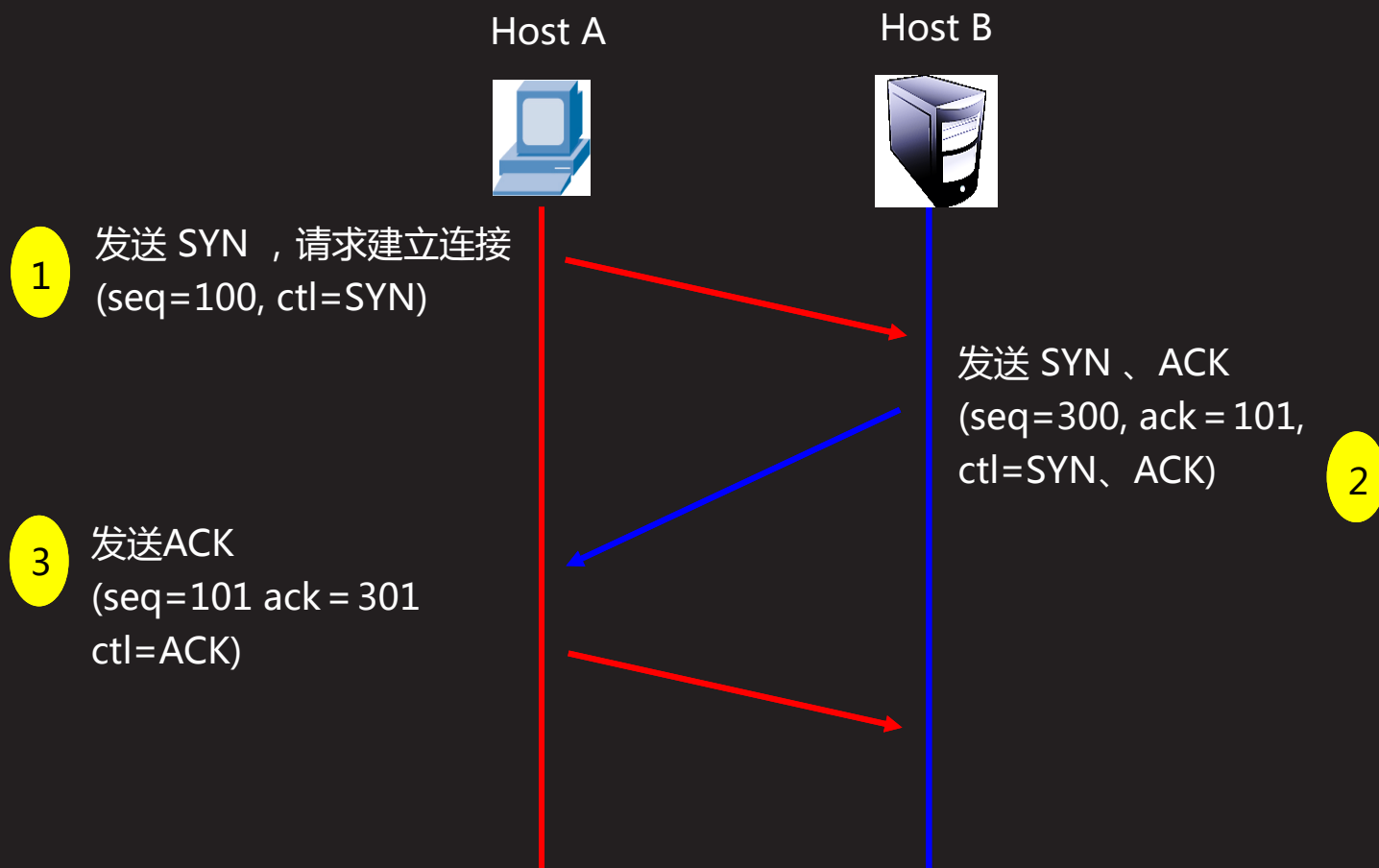


# TCP的封装格式



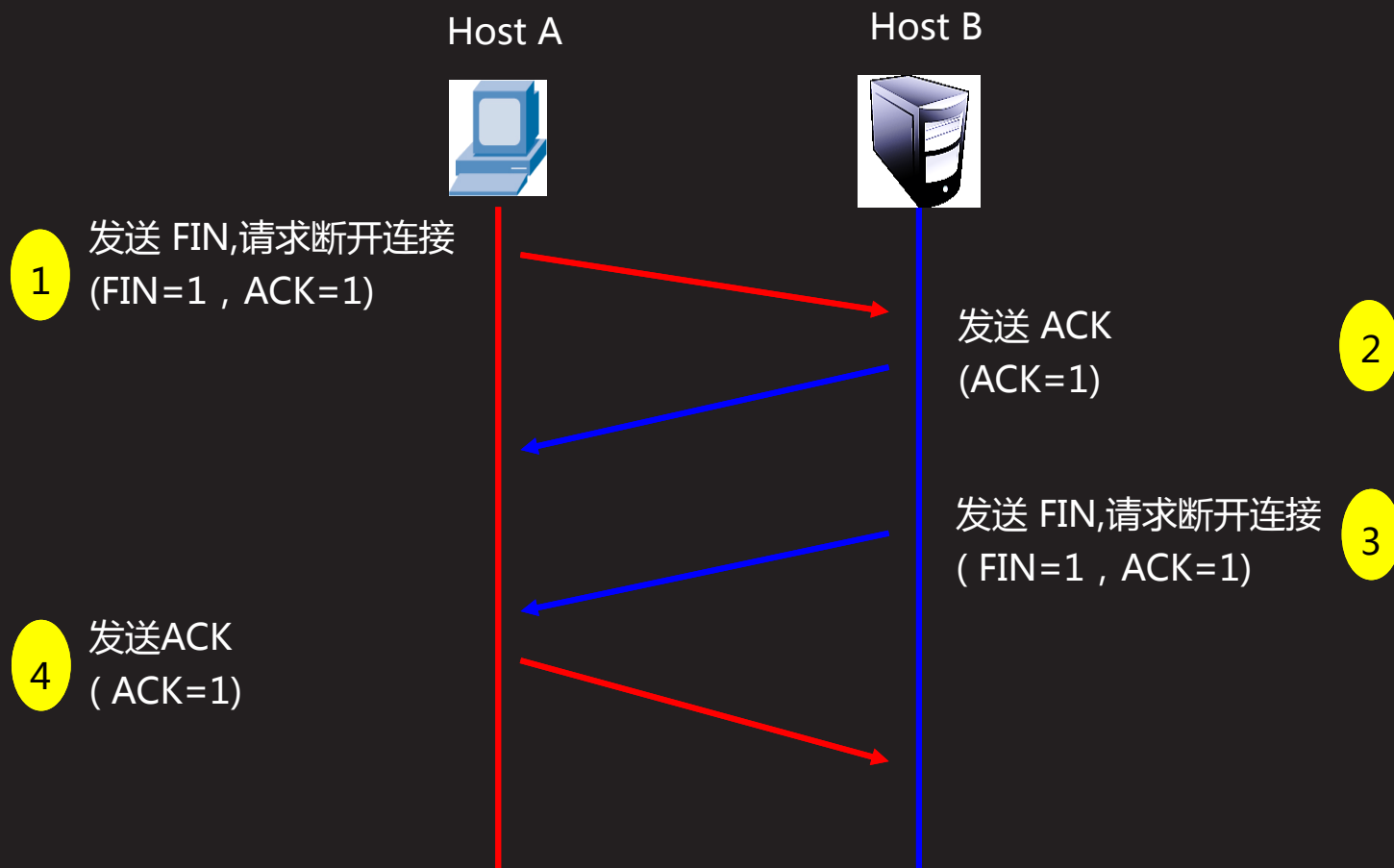
# TCP的连接与断开

- TCP的连接 - 三次握手



# TCP的连接与断开（续1）

- TCP的四次断开



# TCP的应用

端口	协议	说明
21	FTP	文件传输协议，用于上传、下载
23	Telnet	用于远程登录，通过连接目标计算机的这一端口，得到验证后可以远程控制管理目标计算机
25	SMTP	简单邮件传输协议，用于发送邮件
53	DNS	域名服务，当用户输入网站的名称后，由DNS负责将它解析成IP地址，这个过程中用到的端口号是53
80	HTTP	超文本传输协议，通过HTTP实现网络上超文本的传输



# UDP协议工作原理

---



# UDP的封装格式



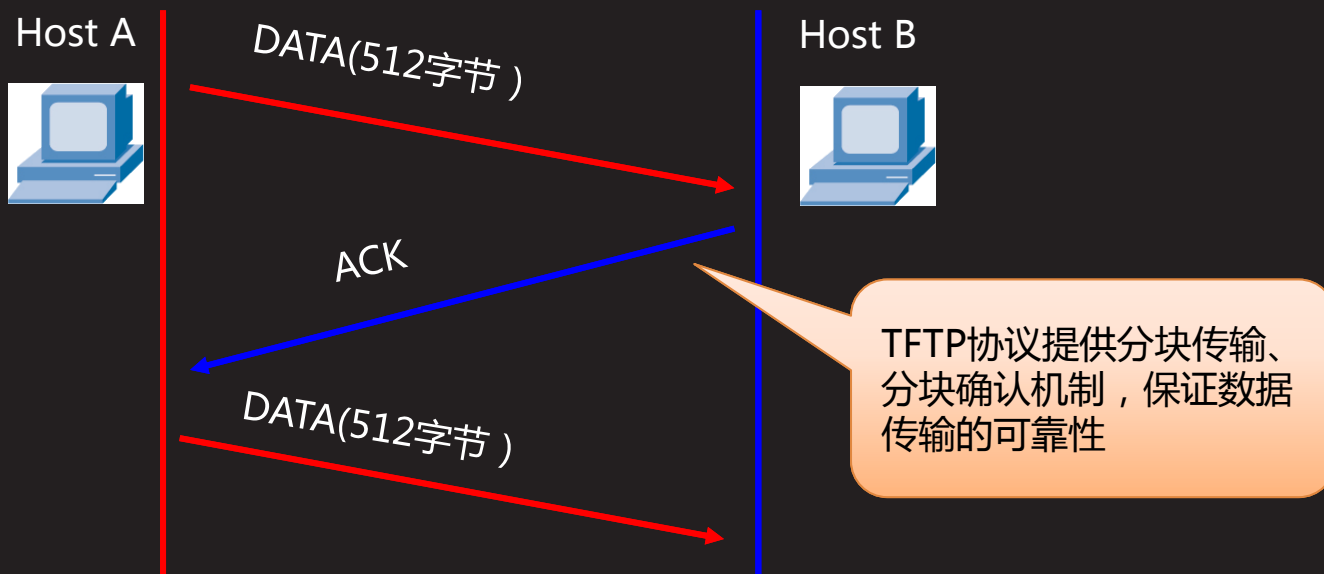
# UDP的应用

端 口	协 议	说 明
69	TFTP	简单文件传输协议
53	DNS	域名服务
123	NTP	网络时间协议



# UDP的流控和差错控制

- UDP缺乏可靠机制
- UDP只有校验和来提供差错控制
  - 需要上层协议来提供差错控制：例如TFTP协议

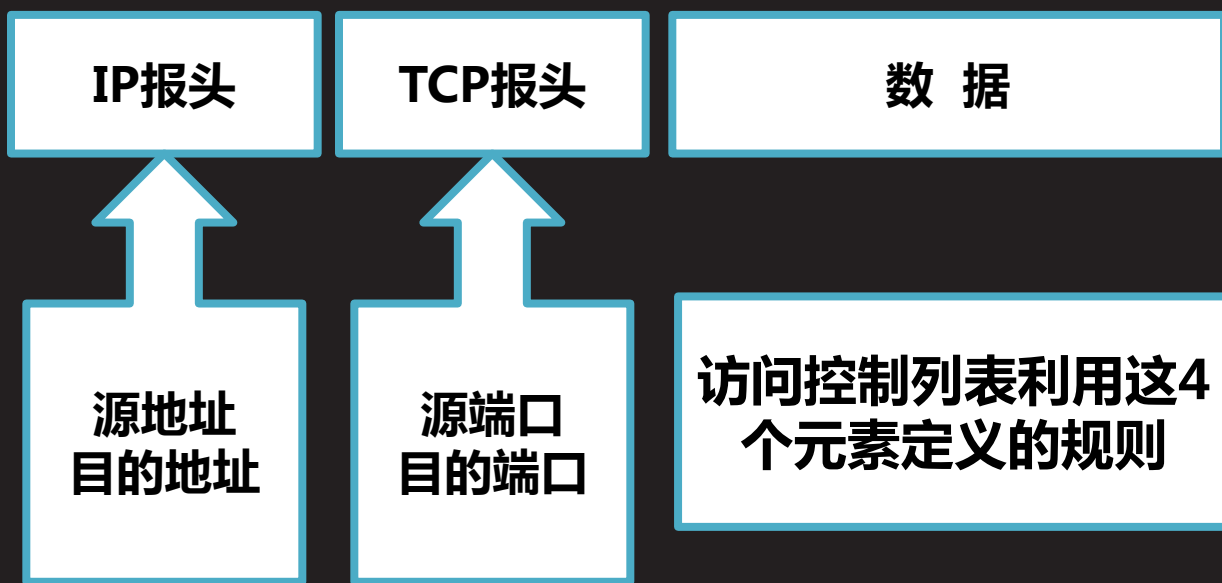


# 访问控制列表概述

---

# 访问控制列表作用

- 访问控制列表 ( ACL )
  - 读取第三层、第四层 头部信息
  - 根据预先定义好的规则对数据进行过滤



# 访问控制列表的工作原理

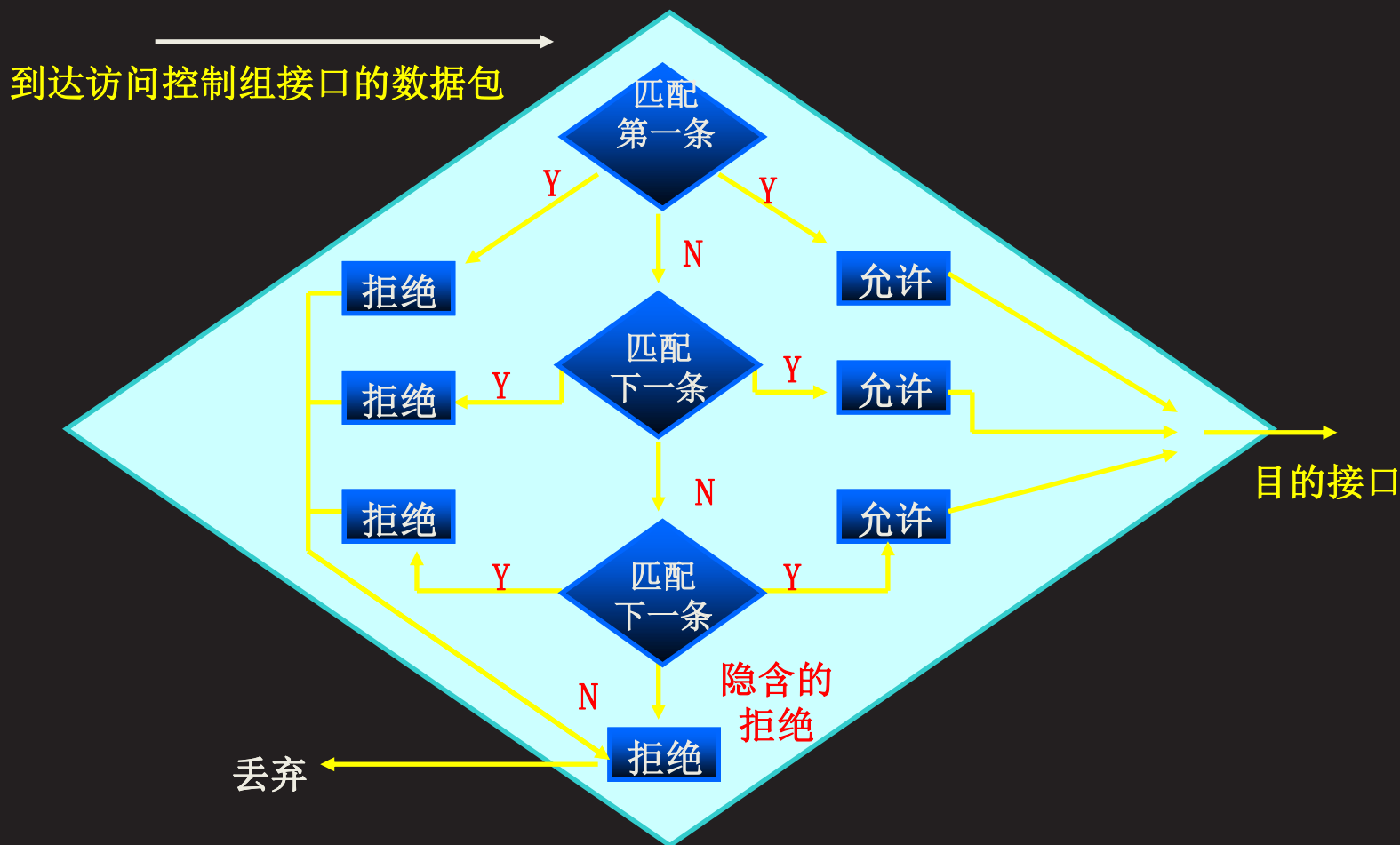
- 访问控制列表在接口应用的方向
  - 出：已经过路由器的处理，正离开路由器接口的数据包
  - 入：已到达路由器接口的数据包，将被路由器处理



- 列表应用到接口的方向与数据方向有关

# 访问控制列表的工作原理（续1）

- 访问控制列表的处理过程



# 访问控制列表的类型

- 标准访问控制列表
  - 基于源IP地址过滤数据包
  - 标准访问控制列表的访问控制列表号是1 ~ 99
- 扩展访问控制列表
  - 基于源IP地址、目的IP地址、指定协议、端口来过滤数据包
  - 扩展访问控制列表的访问控制列表号是100 ~ 199



# 标准ACL配置

---

# 标准访问控制列表的配置

- 创建ACL

```
Router(config)#access-list access-list-number  
{ permit | deny } source [ source-wildcard ]
```

允许数据包通过      拒绝数据包通过

# 标准访问控制列表的配置（续1）

- 应用实例

```
Router(config)# access-list 1 permit 192.168.1.0 0.0.0.255
```

```
Router(config)# access-list 1 permit 192.168.2.2 0.0.0.0
```

- 允许192.168.1.0/24和主机192.168.2.2的流量通过

# 标准访问控制列表的配置（续2）

- 隐含的拒绝语句

```
Router(config)# access-list 1 deny 0.0.0.0 255.255.255.255
```

- 关键字
  - host
  - any

# 标准访问控制列表的配置（续3）

- 将ACL应用于接口

```
Router(config-if)# ip access-group access-list-number {in  
|out}
```

- 在接口上取消ACL的应用

```
Router(config-if)# no ip access-group access-list-number  
{in |out}
```



# 标准访问控制列表的配置（续4）

- 查看访问控制列表

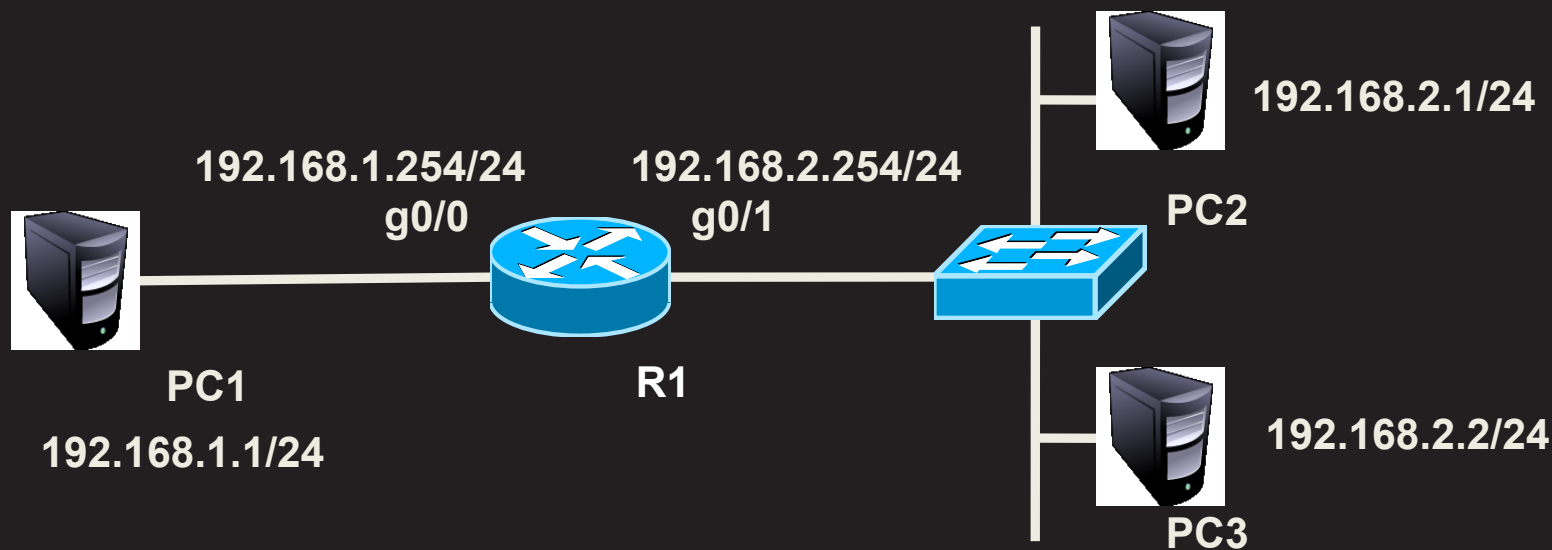
Router(config)# Show access-lists

- 删除ACL

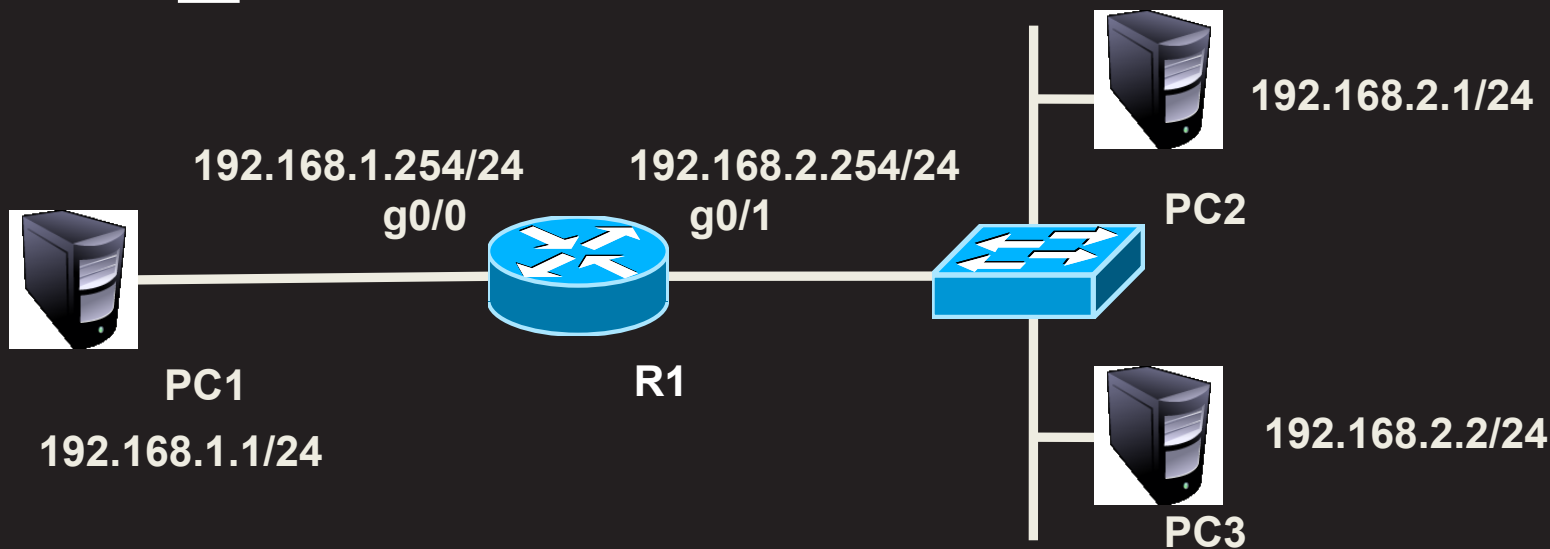
Router(config)# no access-list *access-list-number*

# 标准ACL的配置

- 需求描述
  - 禁止主机PC2访问主机PC1，而允许所有其他的流量



- 需求描述
  - 只允许主机PC2访问主机PC1，而禁止所有其他的流量





# 扩展访问控制列表的配置

- 创建ACL

```
Router(config)# access-list access-list-number { permit | deny }  
      protocol { source source-wildcard destination destination-  
      wildcard } [ operator operan ]
```

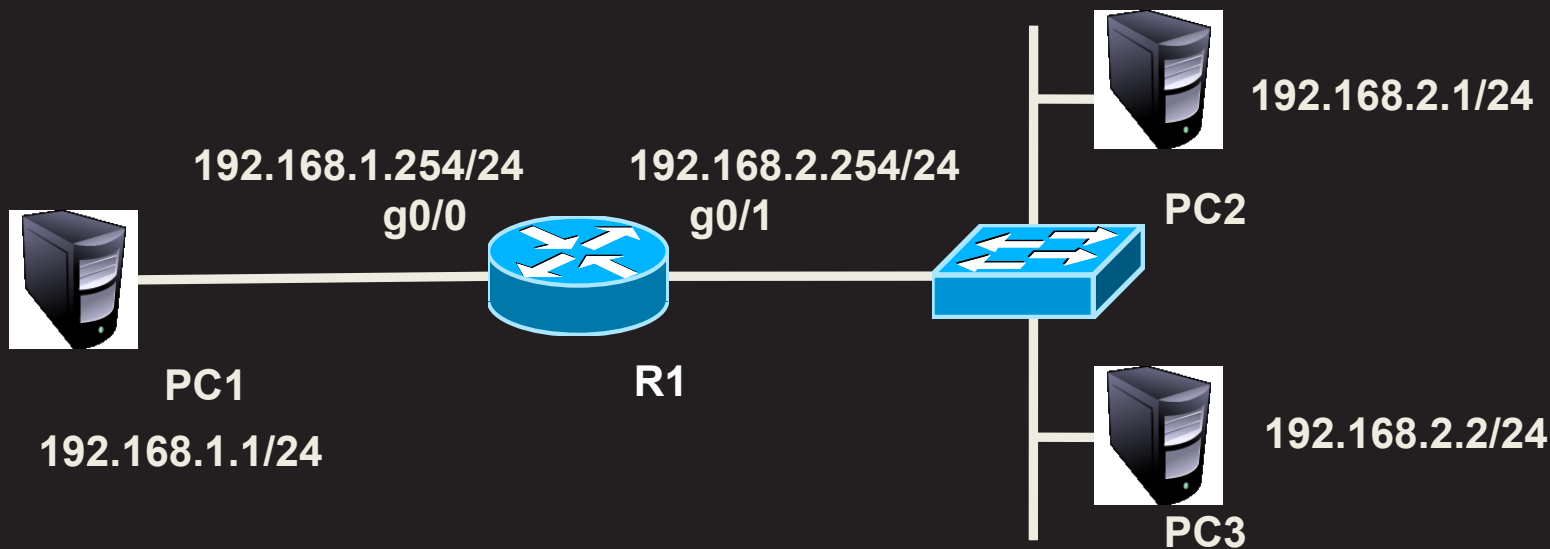
- 应用实例

```
Router(config)# access-list 101 deny tcp 192.168.1.0 0.0.0.255  
      host 192.168.2.2 eq 80
```

```
Router(config)# access-list 101 permit ip any any
```



- 通过配置扩展acl禁止pc2访问pc1的ftp服务，禁止pc3访问pc1的www服务器，所有主机的其他服务不受任何限制



# NAT概述

---

# NAT作用

- NAT
  - Network Address Translation , 网络地址转换
- 作用
  - 通过将内部网络的私有IP地址翻译成全球唯一的公网IP地址, 使内部网络可以连接到互联网等外部网络上。

## 私有ip地址分类

- A类 10.0.0.0~10.255.255.255
- B类 172.16.0.0~172.31.255.255
- C类 192.168.0.0~192.168.255.255



# NAT的特性

- NAT的优点
  - 节省公有合法IP地址
  - 处理地址重叠
  - 安全性



# NAT的特性（续1）

- NAT的缺点
  - 延迟增大
  - 配置和维护的复杂性



# NAT实现方式

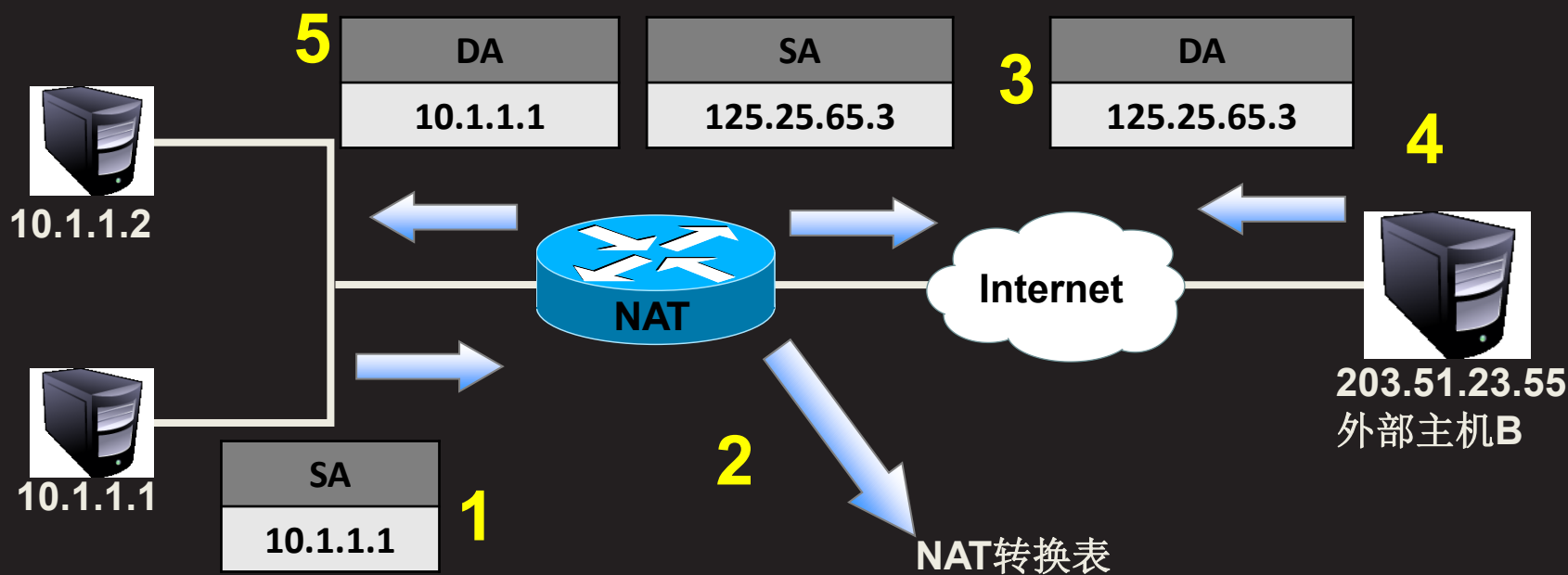
- NAT实现方式
  - 静态转换 ( Static Translation )
  - 端口多路复用 ( Port Address Translation , PAT )





# NAT的工作过程

- 静态

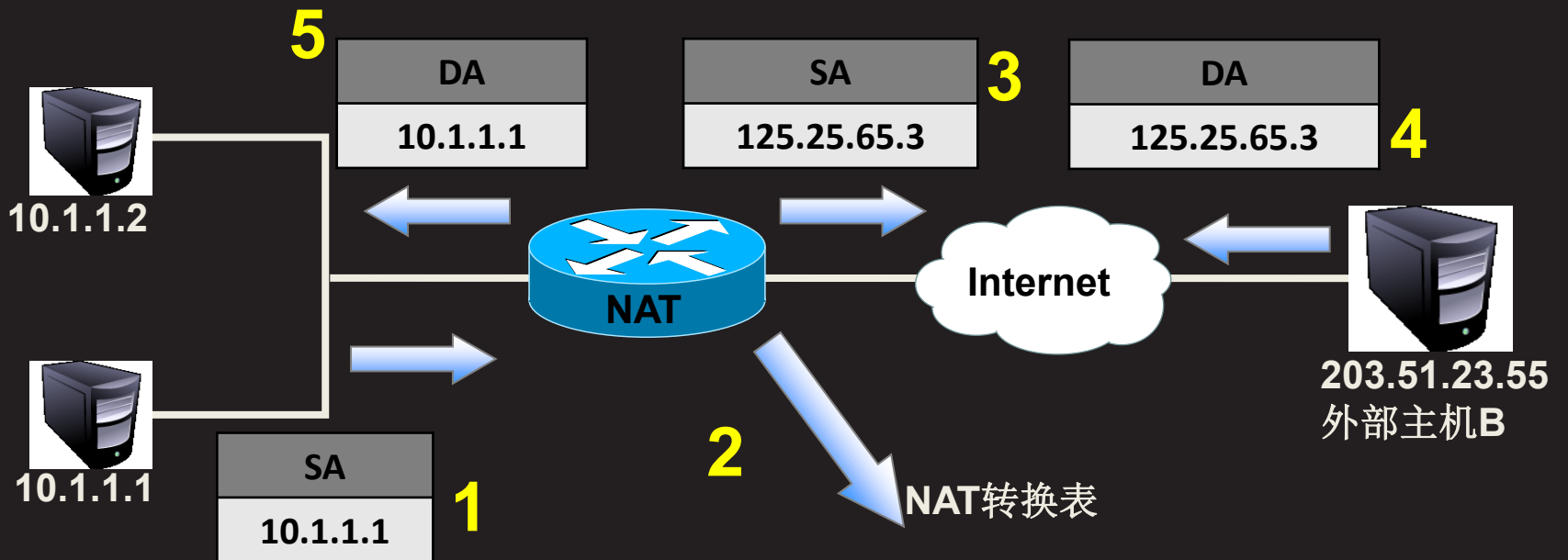


协议	内部局部IP地址	内部全局IP地址	外部IP地址
TCP	10.1.1.1	125.25.65.3	203.51.23.55

# NAT的工作过程（续1）

- PAT

知识讲解



协议	内部局部IP地址	内部全局IP地址	外部IP地址
TCP	10.1.1.1:1492	125.25.65.3:1492	203.51.23.55:23
TCP	10.1.1.2:1493	125.25.65.3:1493	203.51.23.55:80



# 静态转换

---

# 静态NAT

- 静态转换
  - IP地址的对应关系是一一对一，而且是不变的，借助静态转换，能实现外部网络对内部网络中某些特设定服务器的访问。



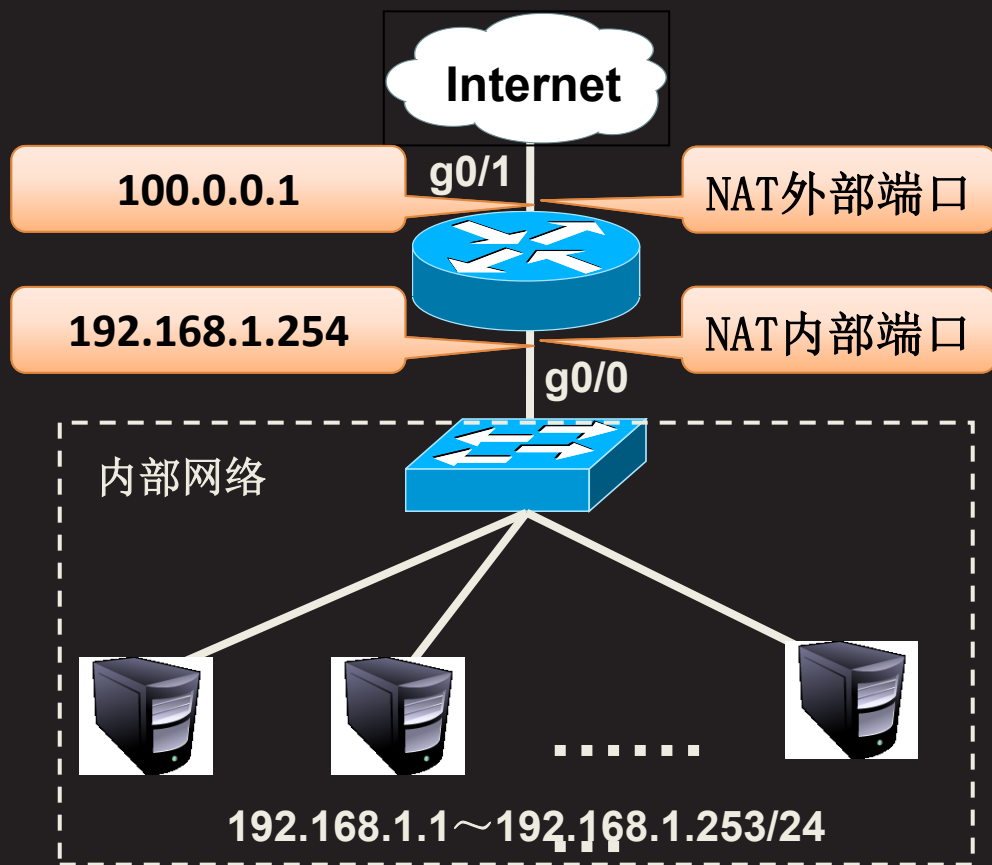
# 静态NAT的配置

- 静态NAT配置步骤
  - 接口IP地址配置
  - 决定需要转换的主机地址
  - 决定采用什么公有地址
  - 在内部和外部端口上启用NAT

Router(config)#ip nat inside source static *local-ip global-ip*

## 静态NAT的配置（续1）

- 将内网地址192.168.1.1静态转换为合法的外部地址100.0.0.2以便访问外网。



## 静态NAT配置（续2）

- 设置外部端口的IP地址：

```
Router(config)#interface g0/1
```

```
Router(config-if)#ip address 100.0.0.1 255.0.0.0
```

```
Router(config-if)#no shut
```

- 设置内部端口的IP地址：

```
Router(config)#interface g0/0
```

```
Router(config-if)#ip address 192.168.1.254 255.255.255.0
```

```
Router(config-if)#no shut
```

- 建立静态地址转换

```
Router(config)#ip nat inside source static 192.168.1.1 100.0.0.2
```

## 静态NAT配置（续3）

- 在内部和外部端口上启用NAT

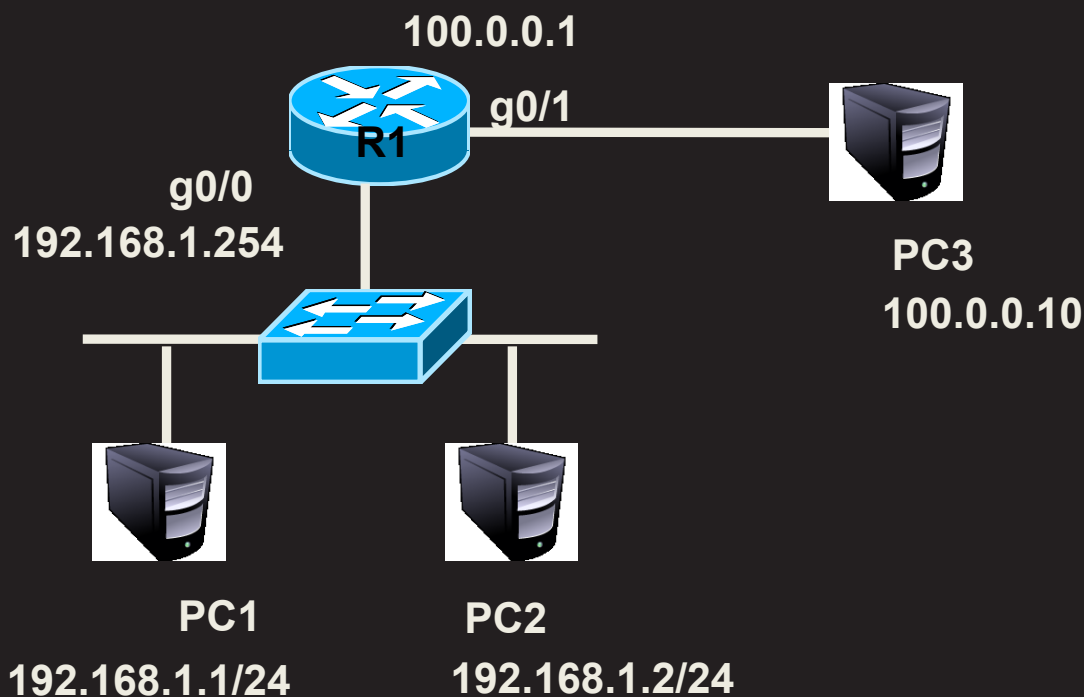
```
Router(config)#interface g0/1
Router(config-if)#ip nat outside
Router(config)#interface g0/0
Router(config-if)#ip nat inside
```





# 案例1：配置静态NAT

在R1上配置静态NAT使192.168.1.1转换为100.0.0.2，  
192.168.1.2转换为100.0.0.3，实现外部网络访问。



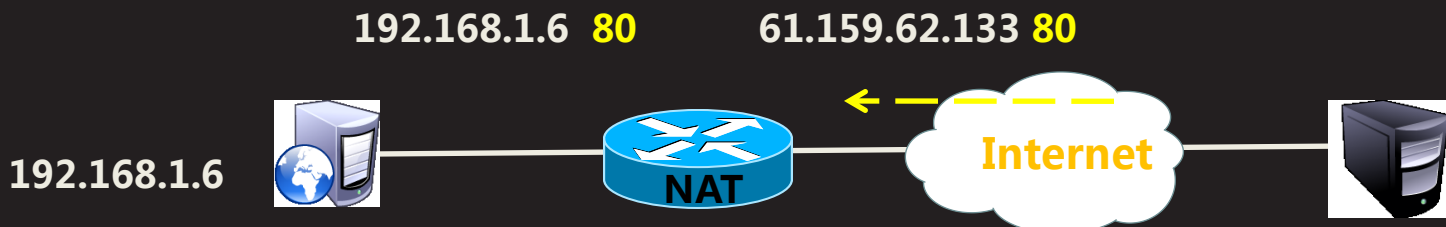
# NAT端口映射

---

# NAT端口映射配置

- 建立NAT端口映射关系
- 配置实例

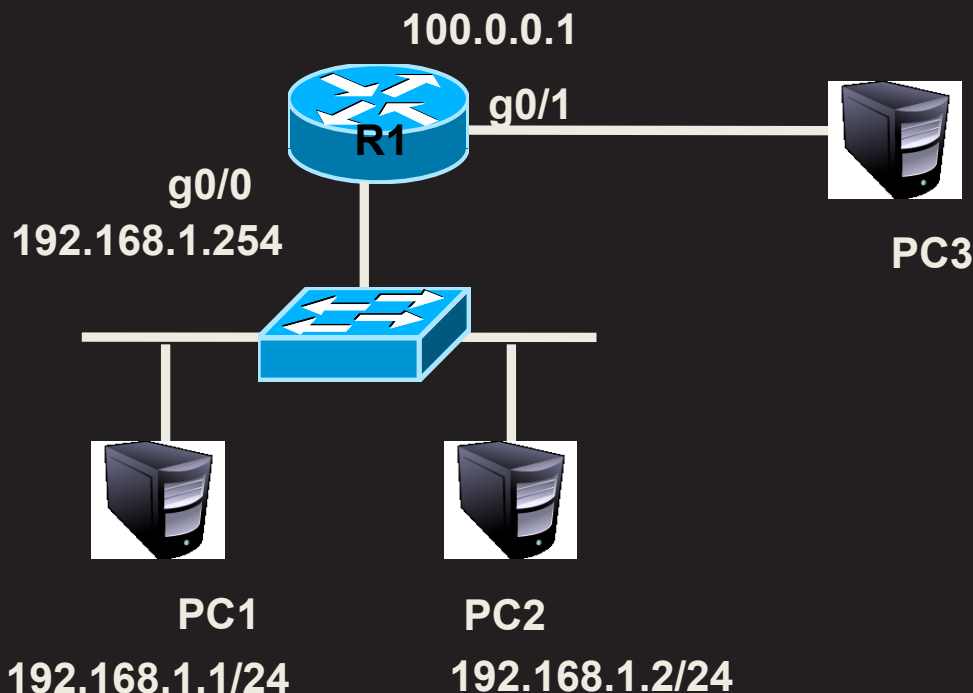
```
Router(config)#ip nat inside source static tcp 192.168.1.6 80  
61.159.62.133 80
```



协议	内部用局部IP地址	内部用全局IP地址	外部用全局IP地址
TCP	192.168.1.6:80	61.159.62.133:80	155.34.2.3

## 案例2：配置端口映射

在R1上配置端口映射将192.168.1.1的80端口映射为100.0.0.2的80端口，将其web服务发布到Internet。



# 端口多路复用(PAT)

A thick, solid red horizontal bar spanning most of the width of the slide, positioned below the title.

# PAT

- PAT(端口多路复用)
  - 通过改变外出数据包的源IP地址和源端口并进行端口转换，内部网络的所有主机均可共享一个合法IP地址实现互联网的访问，节约IP。



# PAT的配置

- PAT配置步骤
  - 接口IP地址配置
  - 使用访问控制列表定义哪些内部主机能做PAT
  - 确定路由器外部接口
  - 在内部和外部端口上启用NAT



# PAT的配置（续1）

- 定义内部ip地址

**Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255**

- 设置复用动态IP地址转换

外部接口

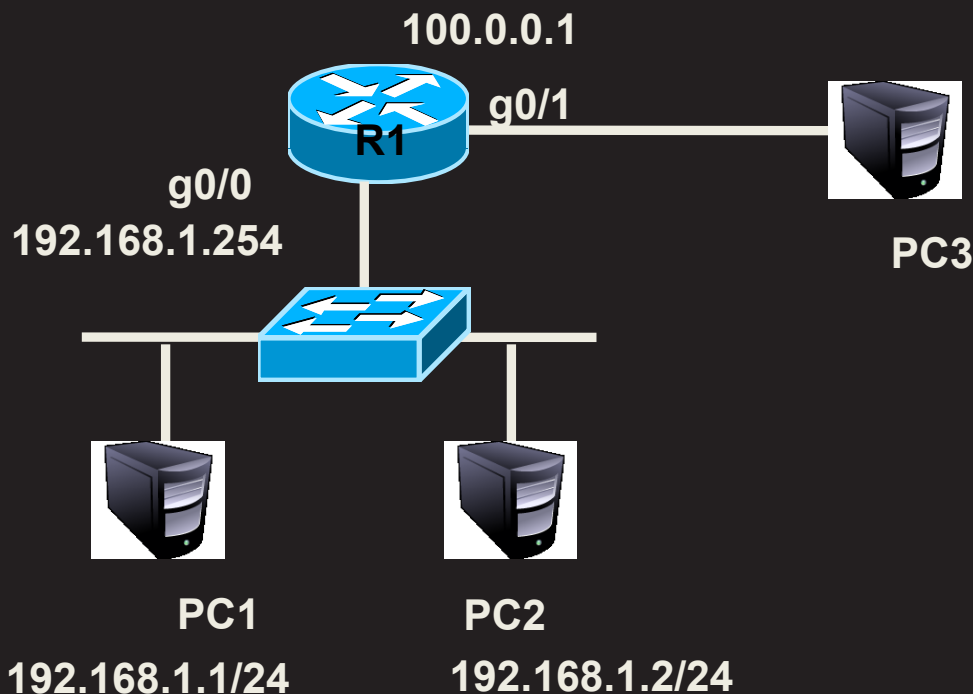
**Router(config)#ip nat inside source list 1 interface g 0/1 overload**

- 在内部和外部端口上启用NAT，以及配置默认路由
  - 与静态NAT配置相同



## 案例4：PAT配置

在R1配置PAT端口多路复用使企业内网192.168.1.0/24复用g0/1端口的IP，实现外部网络的访问。



# 跟踪NAT

- debug ip nat命令跟踪NAT操作

R1#debug ip nat

IP NAT debugging is on

```
*Mar 1 00:03:56.875: NAT: s=192.168.4.2->145.52.23.2, d=1.1.1.1 52225]
*Mar 1 00:03:57.667: NAT*: s=192.168.4.2->145.52.23.2, d=1.1.1.1 [52481]
*Mar 1 00:03:57.811: NAT*: s=1.1.1.1, d=145.52.23.2->192.168.4.2 [52481]
```

s = 192.168.4.2表示源地址是192.168.4.2

d = 1.1.1.1表示目的地址是1.1.1.1

192.168.4.2->145.52.23.2表示将地址192.168.4.2转换为145.52.23.2