

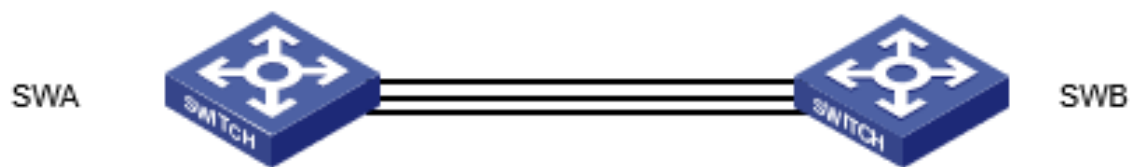
链路聚合

引入

- STP的作用?
- STP的问题?
- 链路聚合：允许将多个物理端口聚合在一起，形成一条逻辑链路。

链路聚合简介

- 产生
- 链路聚合是将多条物理链路聚合在一起，形成一条逻辑链路；
- 优点
 - 根据需要灵活的增加网络设备之间的带宽供给；
 - 增强网络设备之间连接的可靠性；
 - 节约成本。



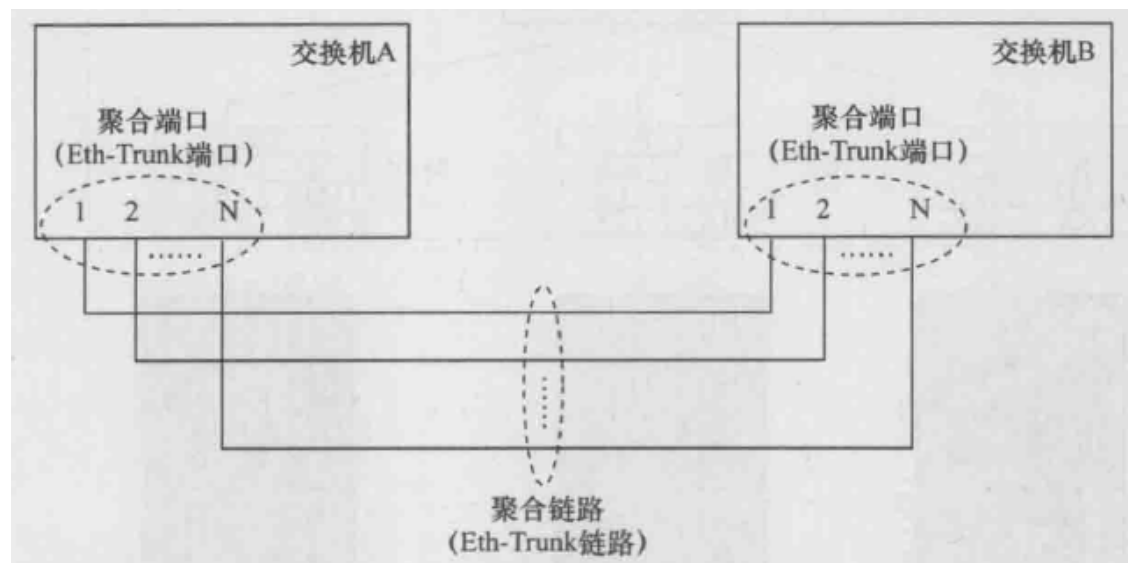
接口规范

- (1) 发送/接收速率为10Mbit/s的以太网端口长被称为标准以太网端口，或标准以太口、10兆以太网端口、10兆以太口、10M以太网端口、10M以太口，或10M口；
- (2) 发送/接收速率为100Mbit/s的以太网端口长被称为快速以太网端口，或快速以太口、100兆以太网端口、100兆以太口、100M以太网端口、100M以太口，或FE端口、FE口（FE：Fast Ethernet简称）；
- (3) 发送/接收速率为1000Mbit/s的以太网端口长被称为千兆以太网端口，或千兆以太口、千兆以太口、千兆口、吉比特端口、吉比特口或GE端口、GE口（FE：Fast Ethernet简称）；
- (4) 发送/接收速率为10Gbit/s的以太网端口常被称为万兆以太网端口，或万兆以太口、万兆口、10GE端口或10GE口；
- (5) 发送/接收速率为100Gbit/s的以太网端口常被称为100GE端口或100GE口。

如果一条链路两端的端口是GE口，那么该链路就是一条GE链路；若链路两端的端口是FE端口，则这条链路是一条FE链路，等等。

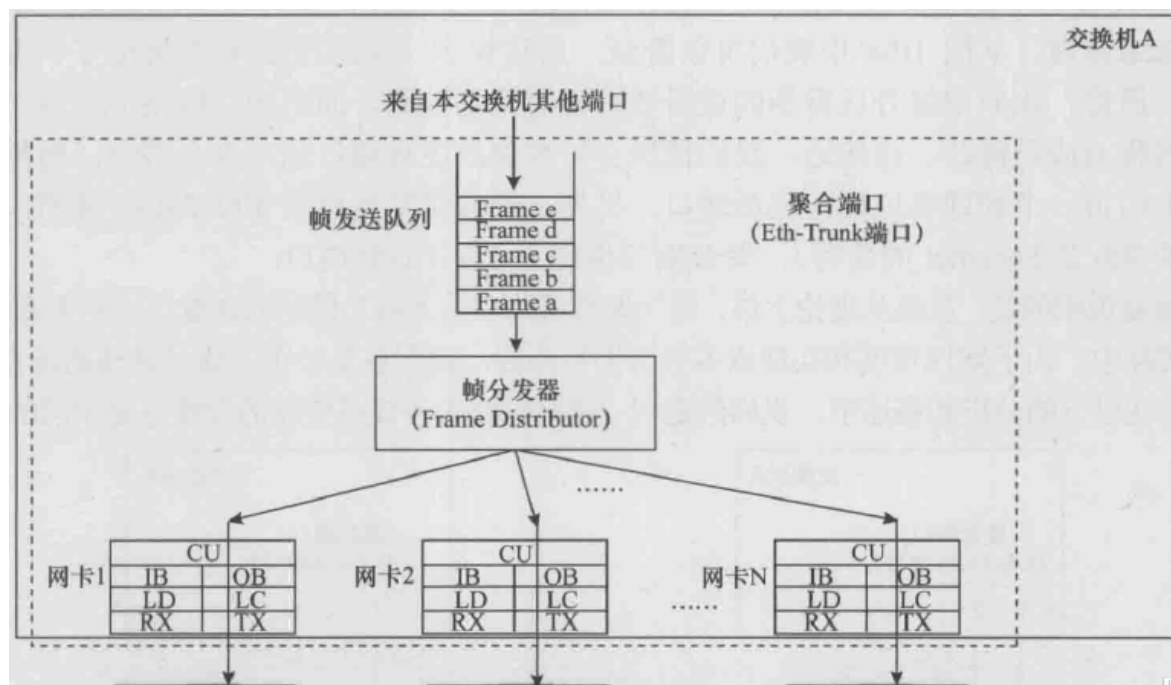
链路聚合的原理

- 把聚合后得到的逻辑链路称为聚合链路，把聚合链路中的每一条物理链路称为成员链路。
- 把聚合后得到的逻辑端口称为聚合端口，把聚合端口中的每一个物理端口称为成员端口。
- 聚合链路也被称为Eth-Trunk链路，聚合端口也被称为Eth-Trunk端口。



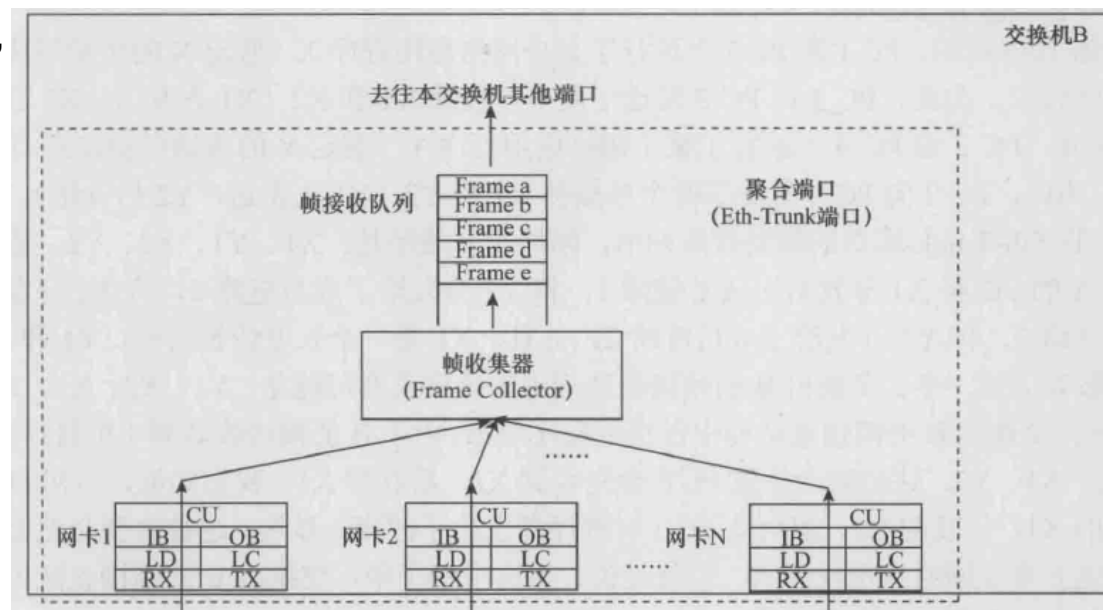
聚合端口数据发送

- 来自交换机A的其他端口的帧进入到Eth-Trunk端口的帧发送队列；
- Eth-Trunk端口的帧分发器（FD）将这些帧按照某种算法依次发给成员端口。FD的分发顺序是：先将Frame A发给某个成员端口，再将Frame b发给某个成员端口，依次类推；
- 每个成员端口按照常规方法将来自FC的帧发送到自己的物理链路上。
- 假若FD能够非常均匀的将帧分发给不同的成员端口，则Eth-Trunk端口的带宽就等于各成员端口的带宽之和，而Eth-Trunk链路的带宽就等于各成员链路的带宽之和。
- 但是，FD对帧的分发并不是非常均匀的，所以Eth-Trunk链路实际上能够提供的最大的链路带宽一般会小于各成员链路带宽的总和。



聚合端口数据接收

- 每个成员端口会按照常规方式接收来自物理链路上的帧，将接收到的帧送到Eth-Trunk端口的帧收集器（FC）；
- 当帧“完全进入”（指：帧的末尾也进入到FC中）到FC中后，FC会将它送至Eth-Trunk端口的帧接收队列中；
- 最先完全进入FC的帧是Frame a,其次是Frame b,等等类推。
- Eth-Trunk端口的帧接收队列中的帧会被依次送往交换机B 的其他端口。



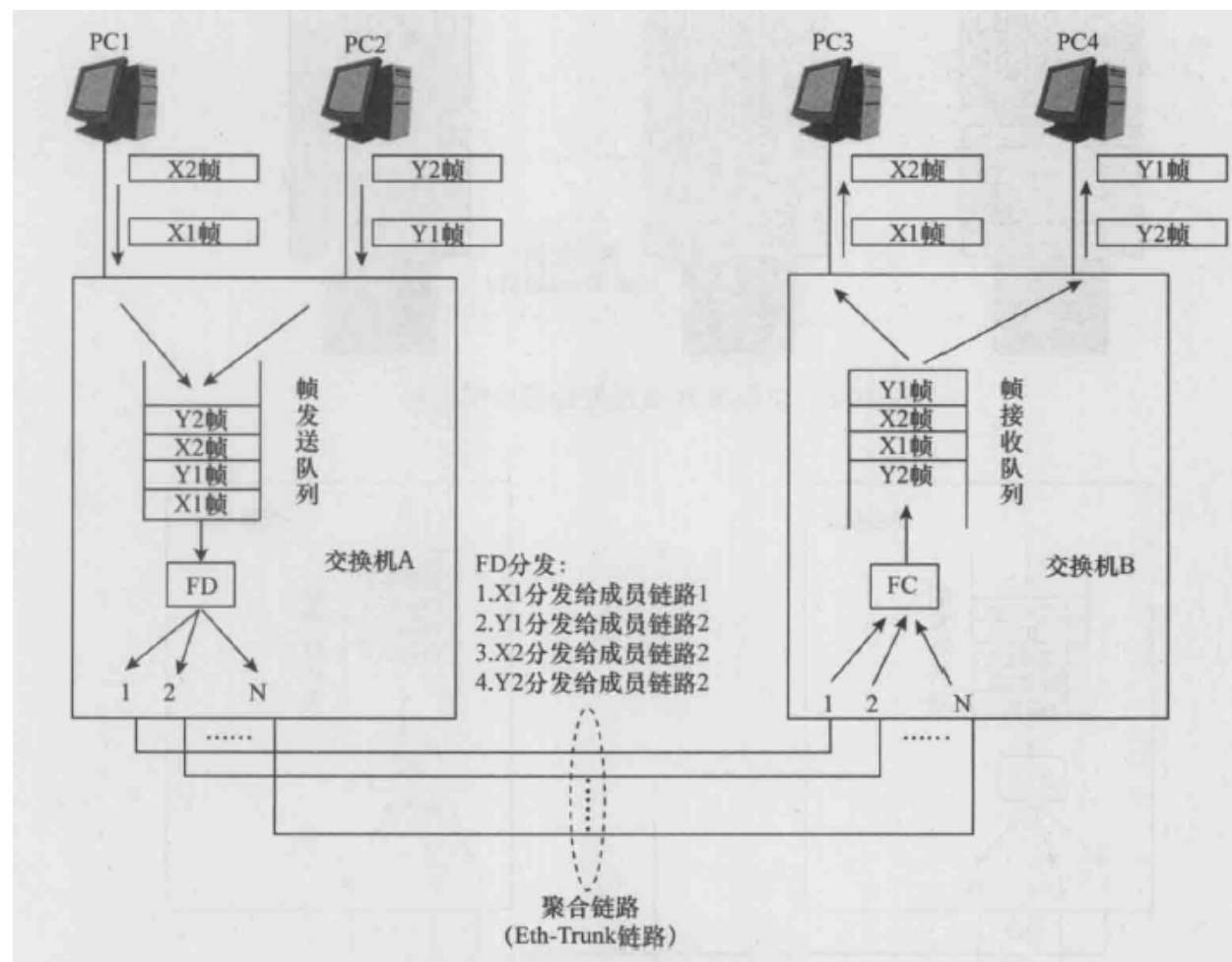
原理总结

- 链路聚合的基本原理就是“流量分担”原理：多条成员链路共同分担了聚合链路的总流量。
- 如果聚合链路中的某一条成员链路发生故障而中断了，那么聚合链路的总流量会继续由其他的成员链路来分担（本该由故障链路所分担的流量将会被FD转移到其他的成员链路）。

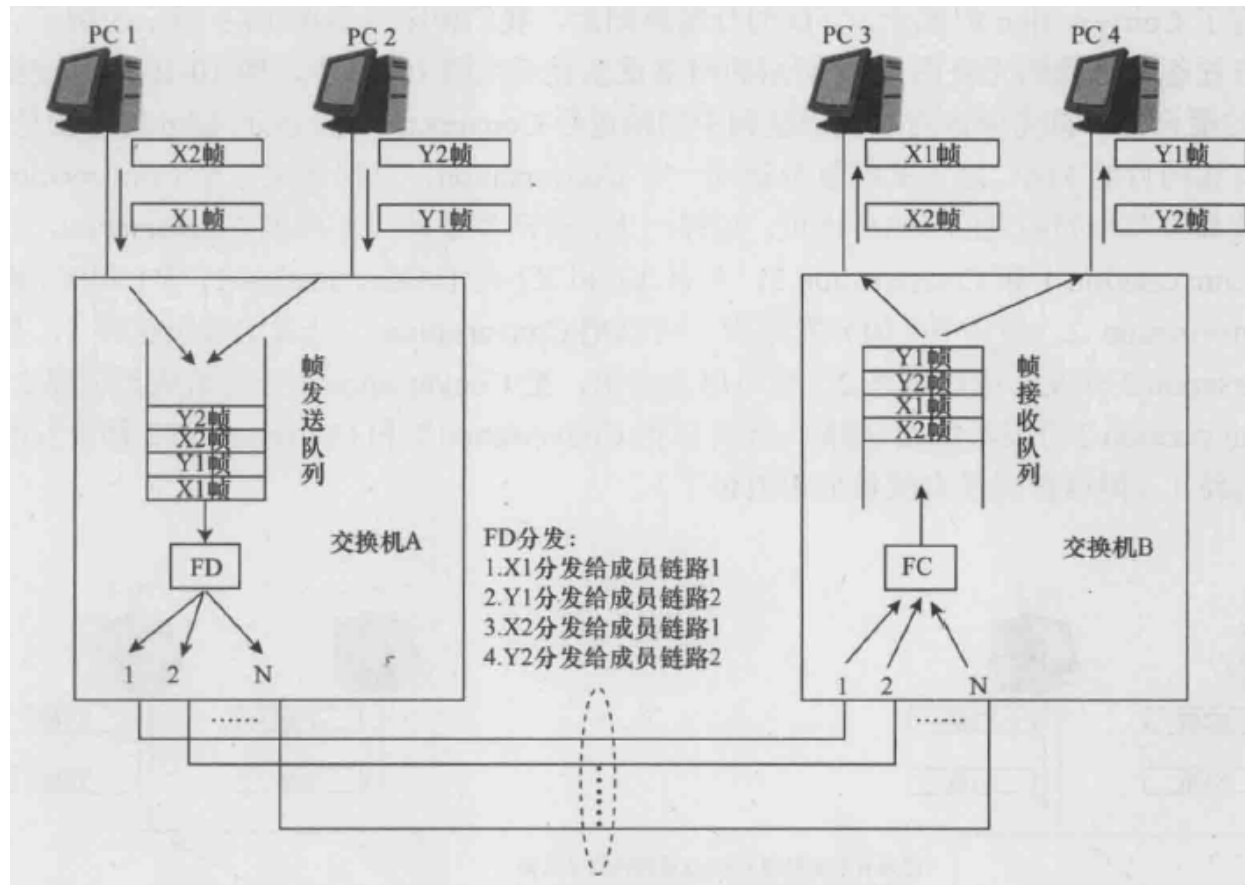
乱序问题

- 所谓的乱序现象就是指：交换机B的帧接收队列中帧的排序不同于其在交换机A的帧发送队列中的顺序。
- 乱序现象分为两种：“有害”乱序；“无害”乱序。

乱序问题



乱序问题



Conversation概念

- 聚合链路在工作过程中，由于帧的长度有所不同，则帧的传输时间就会有所不同，有长有短的时间，同时不同的帧所经过的成员链路也可能不同，所有在一般情况下总是会出现乱序现象。虽然无法避免乱序现象的发生，但是可以尽可能的避免有害乱序现象的发生。
- 避免有害乱序现象的关键是：聚合端口的FD是怎样将帧分发给不同的成员端口的。

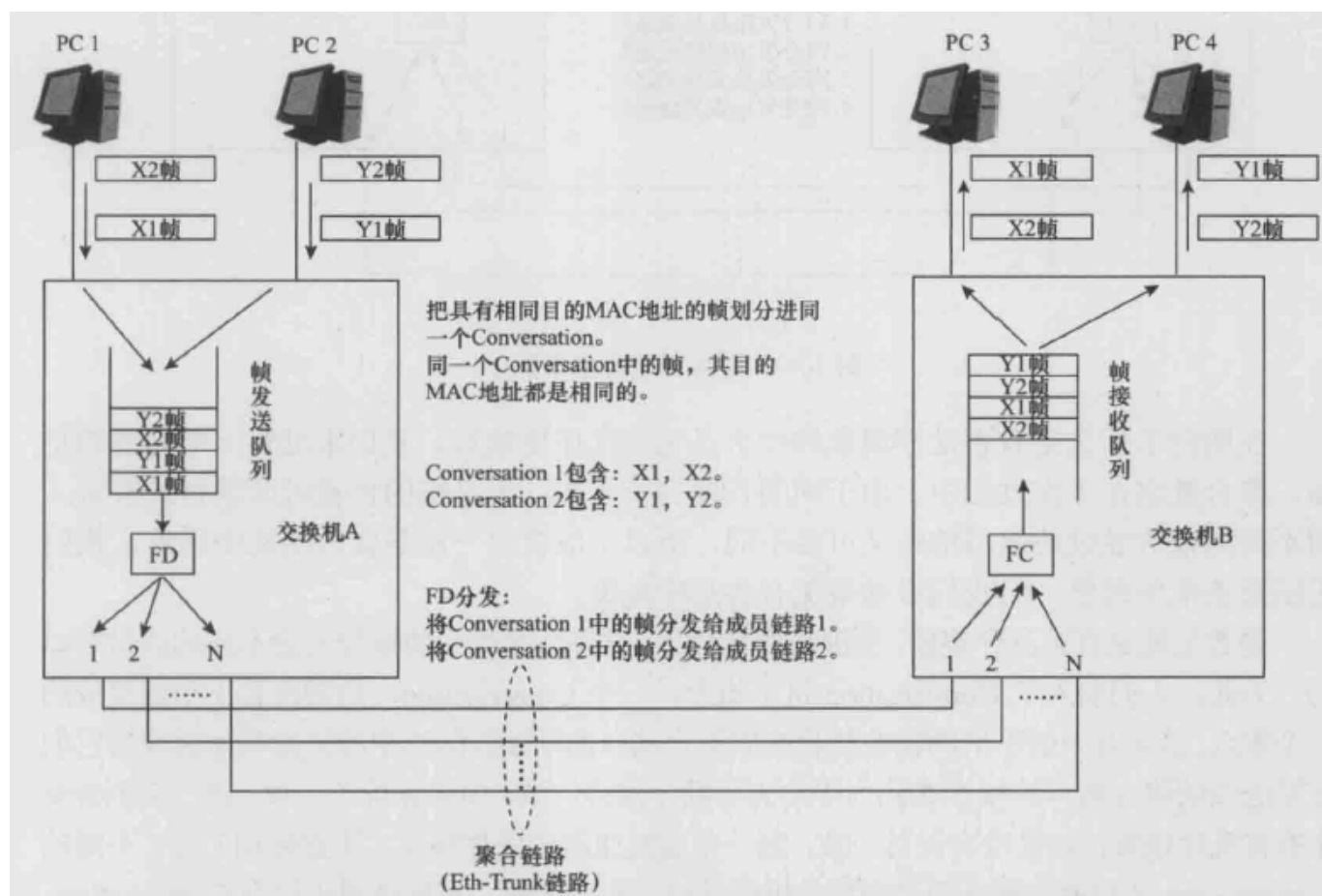
Conversation概念

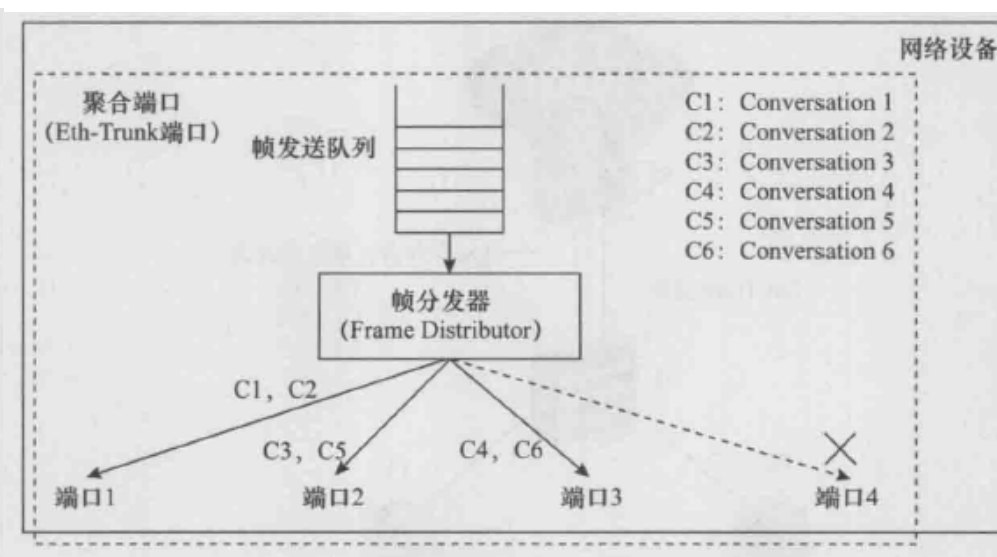
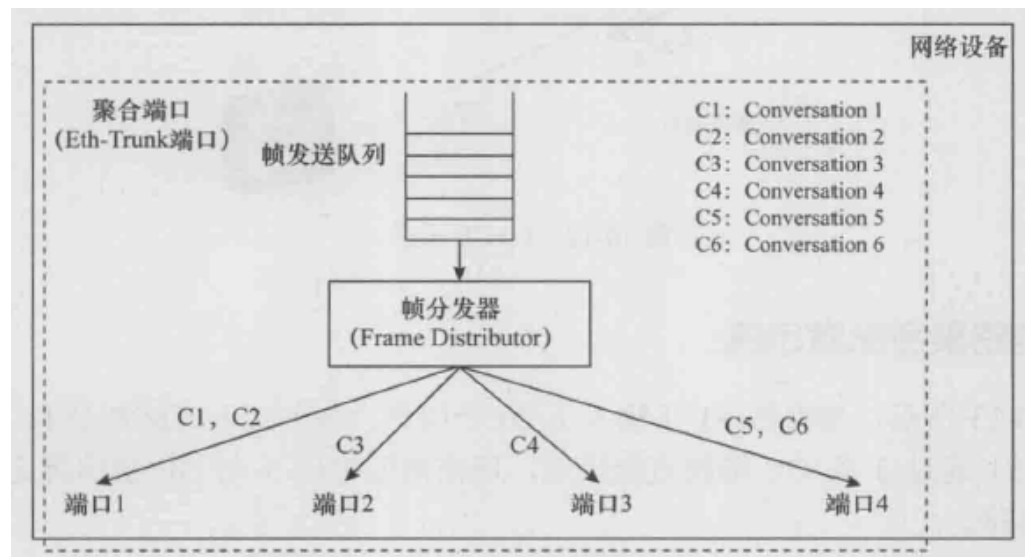
- 一个Conversation是指由若干个帧组成的一个集合，该集合中的不同的帧在接收端的聚合端口的帧的接收队列中的先后顺序必须与它们在发送端的聚合端口的帧的发送队列中的先后顺序保持一致。如果保持了一致，则一定不会发生有害乱序现象；若没有保持一致，则一定会发生有害乱序现象。
- 强调:不同的Conversation之间的交集必须是一个空集，即同一个帧，不能既属于这个Conversation，又属于另外一个Conversation；一个帧不能不属于任何Conversation。

聚合端口需要遵从的分发原则

- 同一个Conversation中的帧，必须被分发给同一条成员端口（可避免有害乱序现象）；
- 不同Conversation中的帧，可以被分发给同一个成员链路，也可以被分发给不同的成员链路（可实现流量分担）；
- 同一个不会乱序？
- 不同的存在乱序？

- 具有相同目的MAC地址的帧分进同一个Conversation，而且要保证同一个Conversation中的帧都具有相同的目的MAC地址



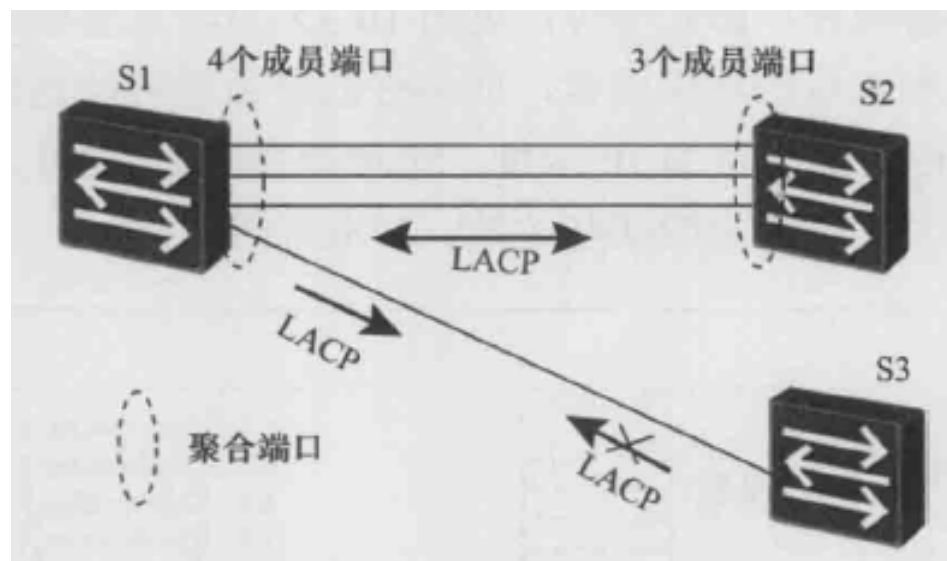


LACP协议

- Link Aggregation Control Protocol
- LACPPUD进行交互
- IEEE 802.3ad （包含： LACP和Market Protocol两个协议）

链路聚合实现模式

- 手工负载分担模式
- LACP模式



链路聚合配置步骤

(1) 第1步：创建eth-trunk端口

在思科网络技术中是创建port-channel组

(2) 第2步：【可选】配置链路聚合模式

mode lacp 或mode manual load-balance

默认为：mode manual load-balance

(3) 第3步：将物理端口加入eth-trunk口

trunkport g0/0/1 to g0/0/3

(4) 对eth-trunk口配置为trunk口，并放行流量。

链路聚合的验证

```
[S1]display eth-trunk 1 verbose
```

WorkingMode:NORMAL

//normal表示是当前是手工负载分担模式

如果显示lacp则表示工作模式为lacp模式

eth-trunk1口的转发量是所有成员端口转发量的总和。

补充：思科交换机作链路聚合

- 1) 二层链路捆绑
- ===创建端口通道组，组的编号为1===
- `sw1(config)#int port-channel 1`
- ===将端口通道组的接口配置为trunk口
- `sw1(config-if)#switchport mode trunk`
- ===指定端口范围，将f0/1-3端口配置为trunk口，并加入端口通道组1种===
- `SW1(config)#int range f0/1-3`
- `sw1(config-if-range)#switchport mode trunk`
- `sw1(config-if-range)#channel-group 1 mode on`

补充：思科交换机作链路聚合

- 1) 二层链路捆绑
- ===查看链路捆绑状态信息===

```
sw1#show etherchannel
```

```
Channel-group listing:
```

```
-----
```

```
Group: 1
```

```
-----
```

```
Group state = L2
```

```
Ports: 3 Maxports = 8
```

```
Port-channels: 1 Max Port-channels = 1
```

```
Protocol: -
```

补充：思科交换机作链路聚合

- 2) 三层链路捆绑
- ===在三层交换机上启用三层功能===
- SW1(config)#ip routing
- ===创建端口通道组，组的编号为10，对端口通道组启用三层功能，并配置IP地址，最后启用接口再保存退出===
- SW1(config)#int port-channel 10
- SW1(config-if)#no switchport
- SW1(config-if)#ip address 12.1.1.1 255.255.255.252
- SW1(config-if)#no shut
- SW1(config-if)#exit

补充：思科交换机作链路聚合

- 2) 三层链路捆绑
- ===将f0/1-3号接口配置为三层端口，并加入端口通道组10===
- SW1(config)#int range f0/1-3
- SW1(config-if-range)#no switchport
- SW1(config-if-range)#channel-group 10 mode on
- ===将连接PC1的交换机接口f0/4配置为三层端口，并配置IP地址，该IP地址作为PC1的网关===
- SW1(config)#int f0/4
- SW1(config-if)#no switchport
- SW1(config-if)#ip add 10.1.1.254 255.255.255.0
- SW1(config-if)#no shut

补充：思科交换机作链路聚合

- 2) 三层链路捆绑
- ===在交换机SW1上查看三层链路捆绑的工作状态===

```
SW1#show etherchannel
                        Channel-group listing:
                        -----

Group: 10
-----
Group state = L3
Ports: 3 Maxports = 8
Port-channels: 1 Max Port-channels = 1
Protocol:      -
SW1#|
```

补充：思科交换机作链路聚合

- 2) 三层链路捆绑
- ===SW1上查看端口通道组的接口带宽===

```
SW1#show int port-channel 10
Port-channel10 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 0060.4735.8173 (bia
0060.4735.8173)
  Internet address is 12.1.1.1/30
  MTU 1500 bytes, BW 400000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 400Mb/s
  input flow-control is off, output flow-control is off
  Members in this channel: Fa0/1 ,Fa0/2 ,Fa0/3 ,
```

补充：思科交换机作链路聚合

- 1) 三层链路捆绑
- ===在交换机SW1上配置RIP路由协议，实现全为互通===

```
SW1(config)#router rip
SW1(config-router)#no auto-summary
SW1(config-router)#version 2
SW1(config-router)#network 10.0.0.0
SW1(config-router)#network 12.0.0.0
SW1(config-router)#end
```

补充：思科交换机作链路聚合

- 1) 三层链路捆绑
- ===SW2上配置略===
- ===通过PC1 ping PC2, 检验网络连通性===

```
C:\>ping 192.168.1.2
```

```
Pinging 192.168.1.2 with 32 bytes of data:
```

```
Reply from 192.168.1.2: bytes=32 time=1ms TTL=126
```

```
Reply from 192.168.1.2: bytes=32 time<1ms TTL=126
```

```
Reply from 192.168.1.2: bytes=32 time<1ms TTL=126
```

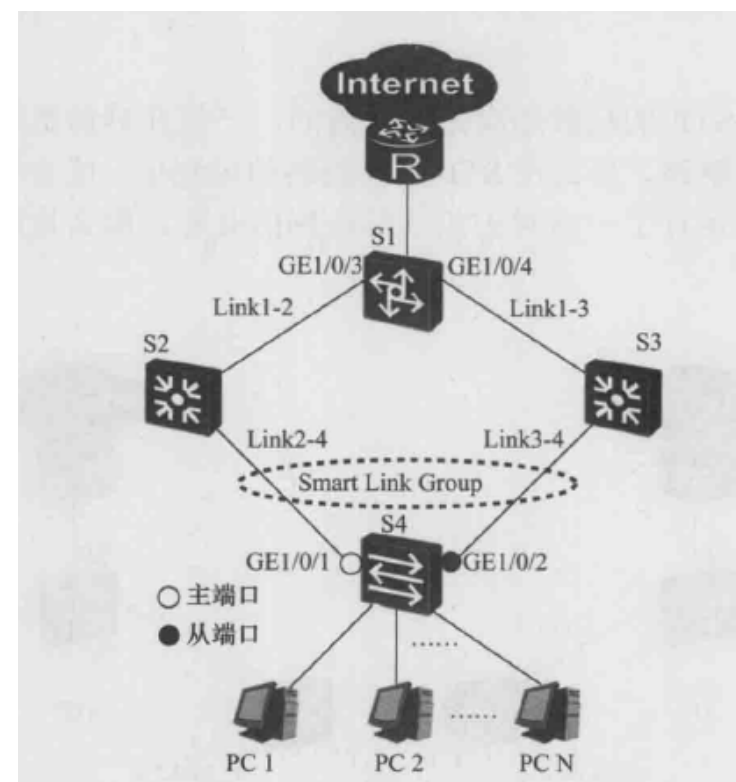
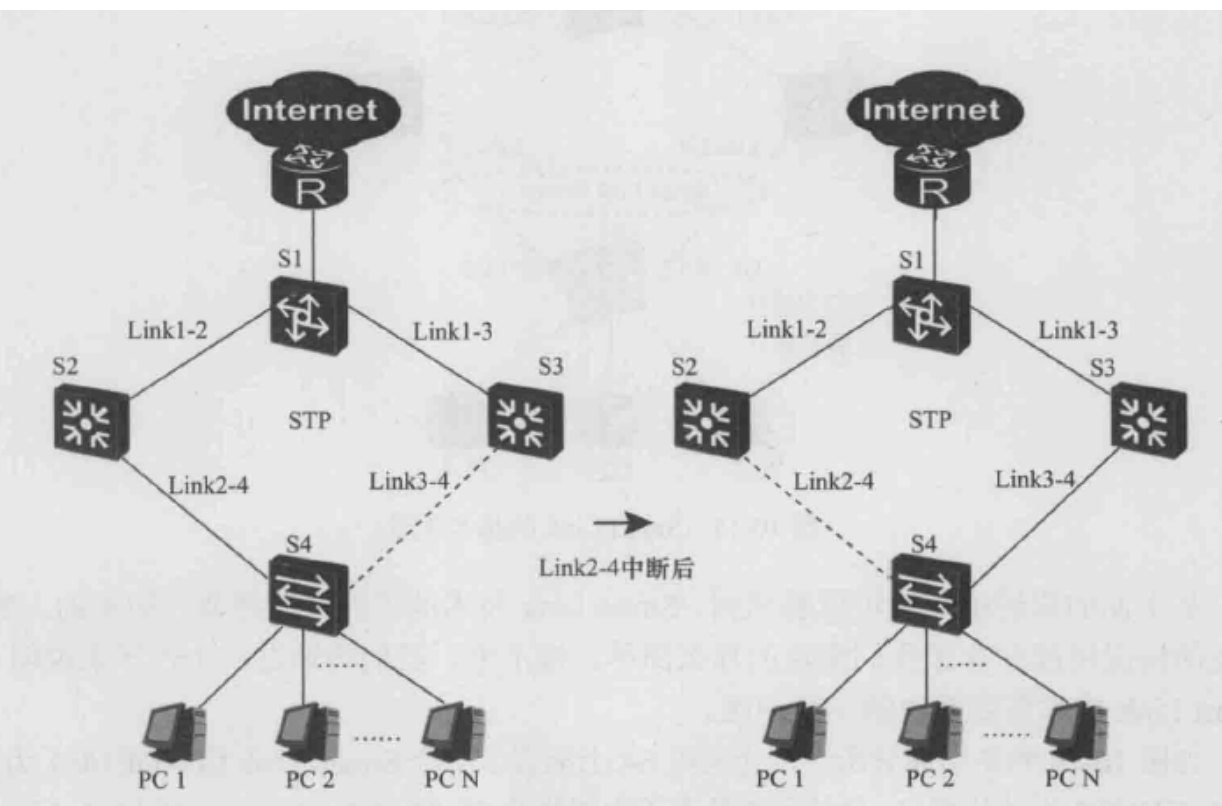
```
Reply from 192.168.1.2: bytes=32 time<1ms TTL=126
```

Smart Link

- 在一定的场景下替代STP协议，并能实现快速（毫秒级）的链路切换。
- 该技术长用于双上行组网环境中。
- 一个Smart Link组由两个端口组成，其中一个为主端口，另一个为从端口。在正常情况下，只有主端口处于转发（Active）状态,而从端口被阻塞，处于待命（Inactive）状态。当主端口发生故障时，Smart Link组会自动将主端口阻塞，并立即将从端口的状态从待命状态切换到转发状态。

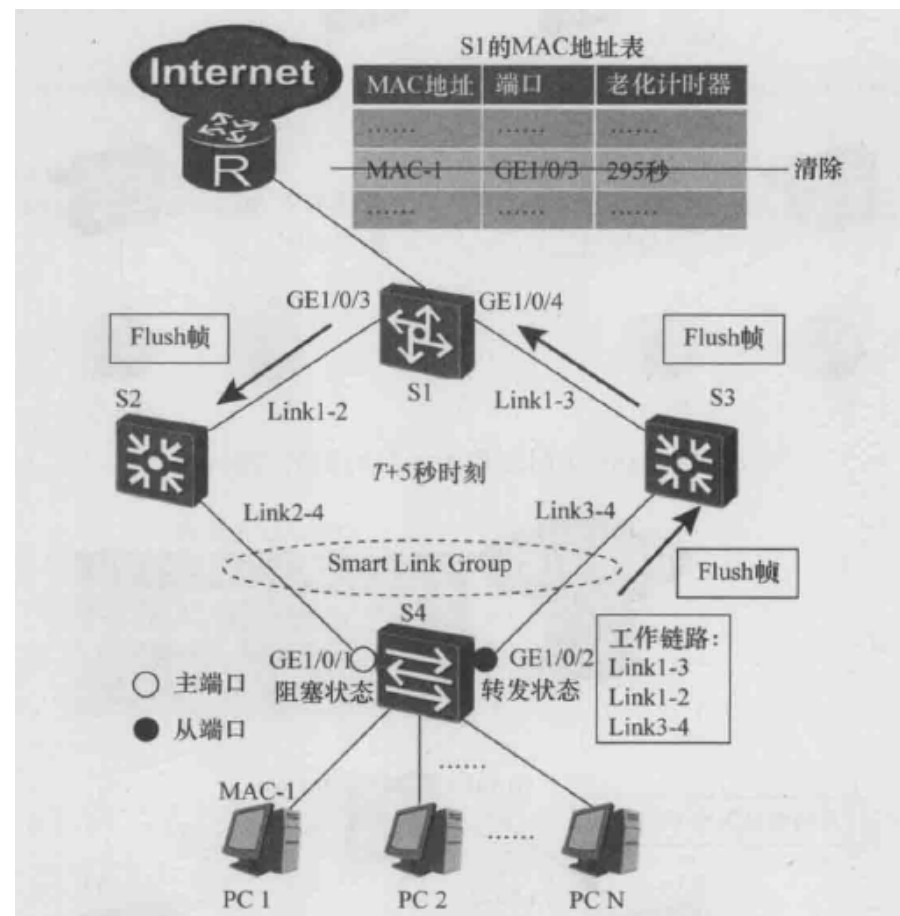
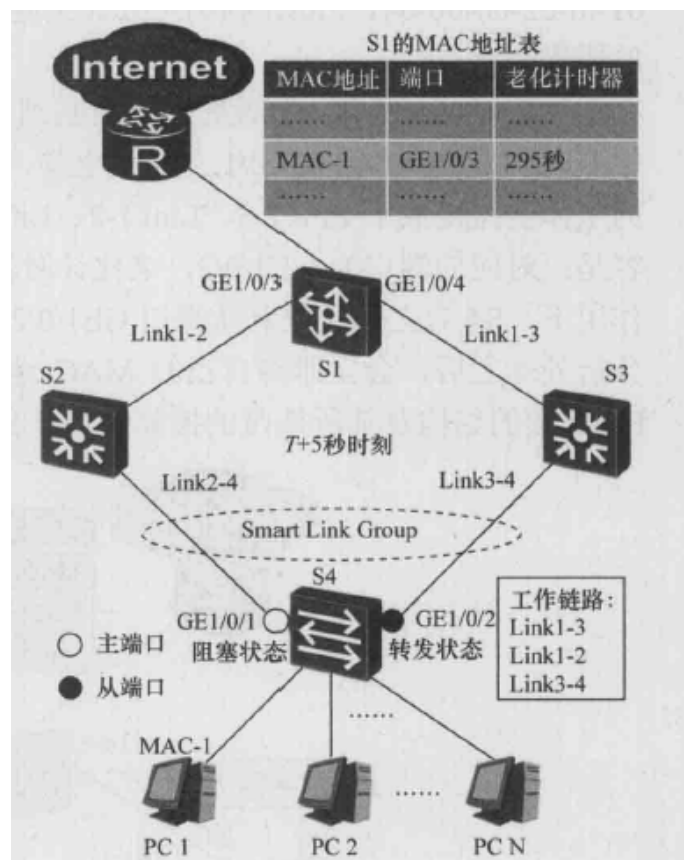
Smart Link

- 华为私有,与STP互斥



Smart Link

- 丢帧问题
- Smart Link定义了一种被称为Flush帧的协议帧，这种帧的目的MAC地址为组播MAC地址01-0f-e2-00-00-04。Flush帧的主要作用是通知相关的交换机即时清除MAC地址表中的错误表项。
- 为了控制Flush帧的传播范围，Smart Link会专门为Flush帧定义一个VLAN，称为控制VLAN。Flush帧在被发送之前必须要带上控制VLAN的Tag。若某台设备需要接收并处理Flush帧，就必须先对该设备进行相应的配置，使它能够接收和识别处理带有控制VLAN Tag的帧。若没有该设置，则其在接收到带有控制VLAN Tag的帧后，会直接将其丢弃。



Smart Link

- **回切功能**
- 在正常的情况下，Smart Link的主端口处于Active状态，从端口处于Inactive状态。当主端口Down掉（主链路中断）后，主端口的状态会切换成Inactive状态，从端口的状态会被切换成Active。但是，当主端口重新恢复正常后（UP后），Smart Link并不会自动将主端口的状态回切到Active，同时也不会将从端口的状态回切到Inactive。如果需要将主端口的状态回切到Active，将从端口回切到Inactive状态，需要配置Smart Link的回切功能。
- 在配置Smart Link回切功能时，还需要配置“回切时间”的参数，其缺省值为60s。即当主端口重新Up后，还需要等待一段时间（回切时间）之后才开始进行回切操作。原因是：主端口重新接通后，其工作状态可能并不是很稳定，甚至会出现闪通和闪断的现象，因此回切操作不宜立刻进行。

Smart Link配置步骤

(1) 在下行设备上，对上联接口禁用STP

```
[S1]int g0/0/1
```

```
[S1-GigabitEthernet0/0/1]stp disable
```

(2) 在下行设备上创建smart link组，将所有上联接口加入smart link组，并指定端口角色；

```
[S1]smart-link group 1
```

```
[S1-smlk-group1]port g0/0/1 master/slave
```

(3) 在下行设备的smart-link组启用flush帧发送功能

```
[S1-smlk-group1]flush send control-vlan 10  
password simple 123
```

(4) 在上行设备的下联接口配置smart-link flush帧接收功能

```
[S2-GigabitEthernet0/0/1]smart-link flush receive  
control-vlan 10 password simple 123
```

(5) 设定回切时间

```
[S1-smlk-group1]restore enable
```

```
[S1-smlk-group1]timer wtr 30
```

(6) 验证smart link组

```
<S1>display smart-link group 1
```

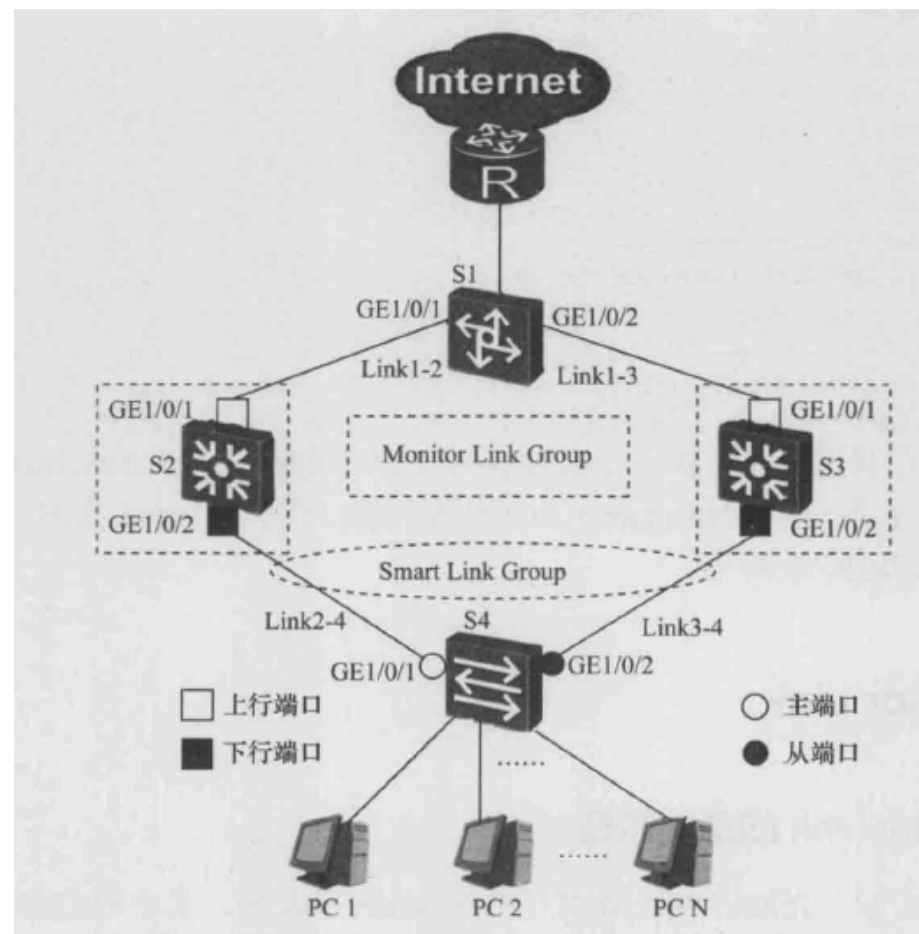
//注意观察主、从端口和回切时间等。

Monitor Link

- 该协议的主要作用是在一定的场景下配合Smart Link的使用，已至更好的避免丢帧情况的发生
- 华为私有
- 一个Monitor Link组由一个上行端口和若干个下行端口组成，如果上行端口因为某些原因而不能正常工作，则其所有的下行端口都必须立刻被Down掉。因为下行端口与上行端口之间存在一种联动机制，下行端口的工作状态需要与上行端口的工作状态保持一致。

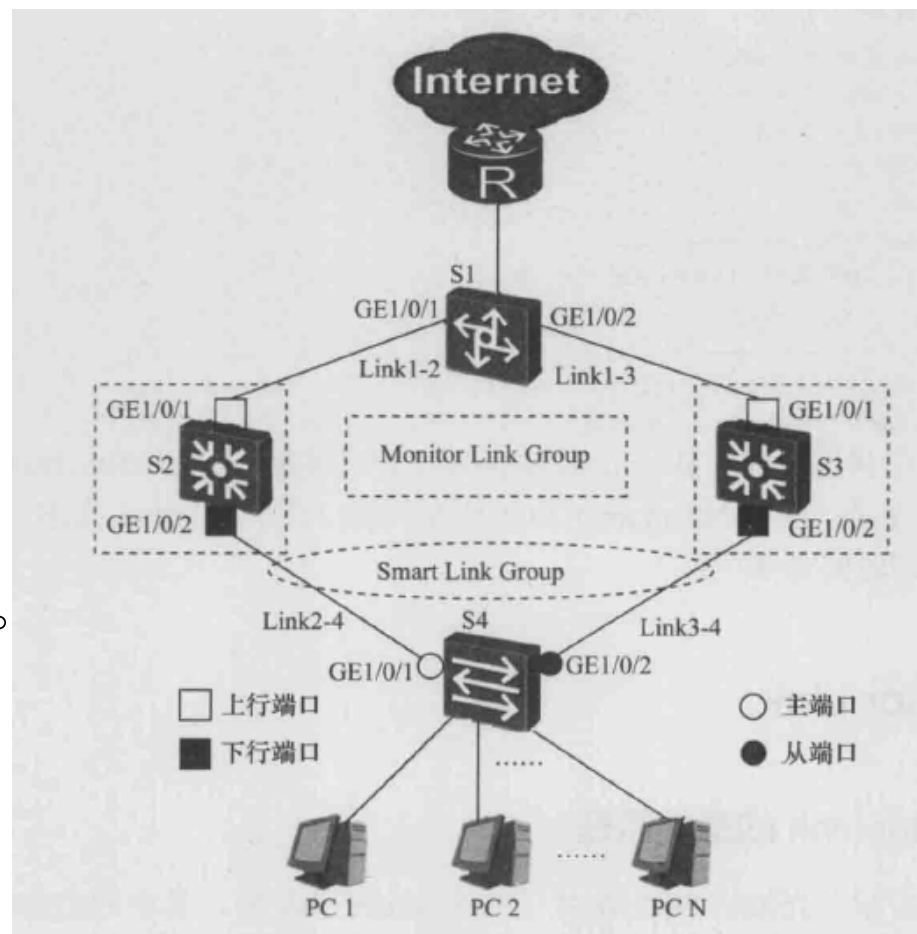
Monitor Link

- 在S2上配置一个Monitor Link组, 而这个Monitor Link组包含了两个端口, 即GE1/0/1端口——上行端口, GE1/0/2端口——下行端口



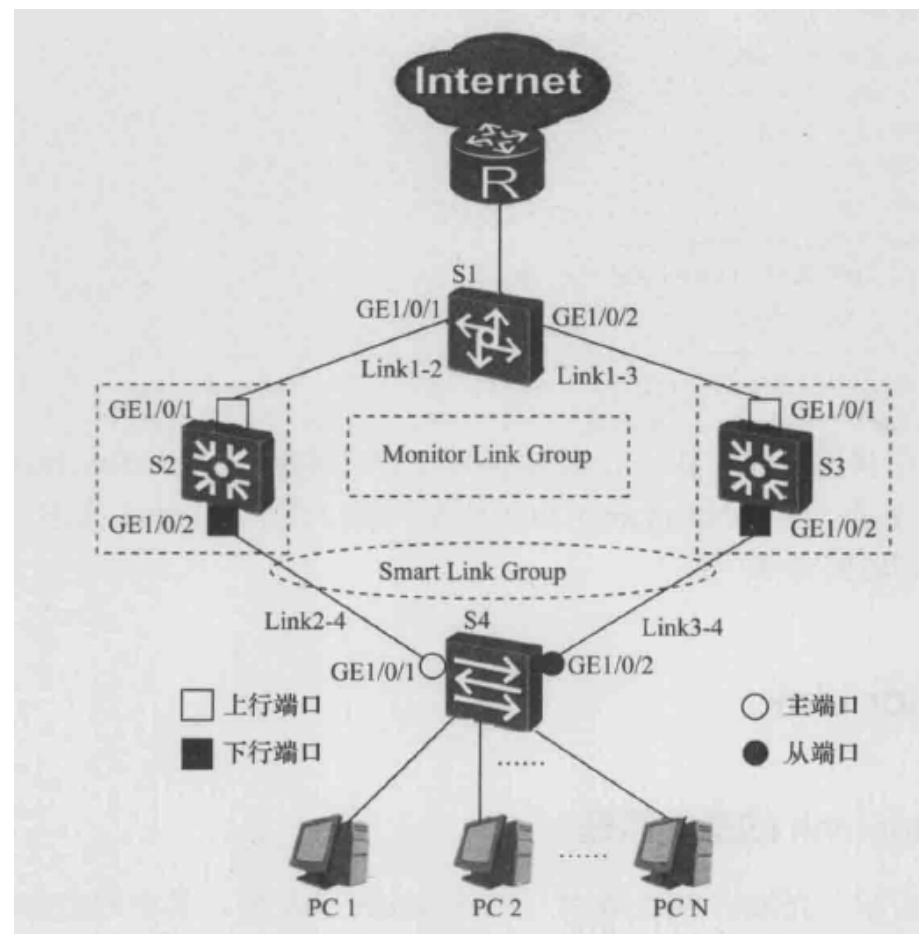
Monitor Link

- 正常情况下，处于工作状态的链路有 Link1-3、Link1-2、Link2-4。如果 S2 的 GE1/0/1 端口发生了故障，那么在 Monitor Link 协议的作用下，S2 的 GE1/0/2 端口就会被立即 Down 掉，从而，S4 的 GE1/0/1 端口也无法正常工作。那么，S4 的 Smart Link 就会立即进行切换操作，将其从端口 GE1/0/2 的状态从 Inactive 切换到 Active。



Monitor Link

- 还可以进一步的进行网络可靠性的增强，就是也在S3上配置一个Monitor Link组，使得S3的GE1/0/2端口可以与GE1/0/1端口实现联动。



Monitor Link

- 回切问题
- 一个Monitor Link组的上行端口不能正常工作时，其所有的下行端口都会被Down掉。但是，如果上行端口恢复了正常，则其下行端口也会自动重新Up。我们也可以为Monitor Link的回切功能配置一个适合的回切时间。

Monitor Link的配置步骤

- (1) 创建Monitor Link组，并填写相应的上行端口和下行端口

```
[SwitchB] monitor-link group 1  
[SwitchB-mtlk-group1] smart-link group 1 uplink  
[SwitchB-mtlk-group1] port gigabitethernet 2/0/1 downlink 1
```

- (2) 配置Monitor Link组的回切时间

```
[SwitchB-mtlk-group1] timer recover-time 10
```