

实验：Eth-Trunk

HCIP 分解实验 - 链路聚合

臧家林制作



链路聚合 实验 1：Eth-Trunk 链路聚合

在没有使用 Eth-Trunk 前，百兆以太网的双绞线在两个互连的网络设备间的带宽仅为 100Mbit/s。若想达到更高的数据传输速率，则需要更换传输媒介，使用千兆光纤或升级成为千兆以太网。这样的解决方案成本较高。如果采用 Eth-Trunk 技术把多个接口捆绑在一起，则可以以较低的成本满足提高接口带宽的需求。例如，把 3 个 100Mbit/s 的全双工接口捆绑在一起，就可以达到 300Mbit/s 的最大带宽。

Eth-Trunk 是一种捆绑技术，它将多个物理接口捆绑成一个逻辑接口，这个逻辑接口就称为 Eth-Trunk 接口，捆绑在一起的每个物理接口称为成员接口。Eth-Trunk 只能由以太网链路构成。

手动负载均衡 (manual load-balance) 模式和 LACP (Link Aggregation Control Protocol) 模式

Trunk 的优势在于：

增加带宽：Eth-Trunk 接口的带宽是各成员接口带宽的总和。

提高可靠性：当某个成员链路出现故障时，流量会自动的切换到其他可用的链路上，从而提供整个 Eth-Trunk 链路的可靠性。

负载分担：在一个 Eth-Trunk 接口内，通过对各成员链路配置不同的权重，可以实现流量负载分担。

Eth-Trunk 技术也存在以下局限性：

必须是点对点连接，使用 Eth-Trunk 接口连接的两台设备之间不允许存在中间设备。

必须是对称连接，Eth-Trunk 接口连接的两台设备上的接口需要成对加入。

负载分担方式分为逐流负载分担和逐包负载分担。

逐流负载分担指根据报文 MAC 地址或 IP 地址区别数据流，使属同一数据流的报文从同一个的成员链路上通过。

逐流负载分担能保证包的顺序，但不能保证带宽利用率。

逐包负载分担是指不区分数据流，而是以报文为单位，将流量分担到不同的成员链路上进行传输。

逐包负载分担能保证带宽利用率，但不能保证包的顺序。

Eth-Trunk 在逻辑上把多条物理链路捆绑等同于一条逻辑链路，对上层数据透明传输。所有 Eth-Trunk 中物理接口的参数必须一致，Eth-Trunk 链路两端要求一致的物理参数有：Eth-Trunk 链路两端相连的物理接口类型、物理接口数量、物理接口的速率、物理接口的双工方式以及物理接口的流控方式。

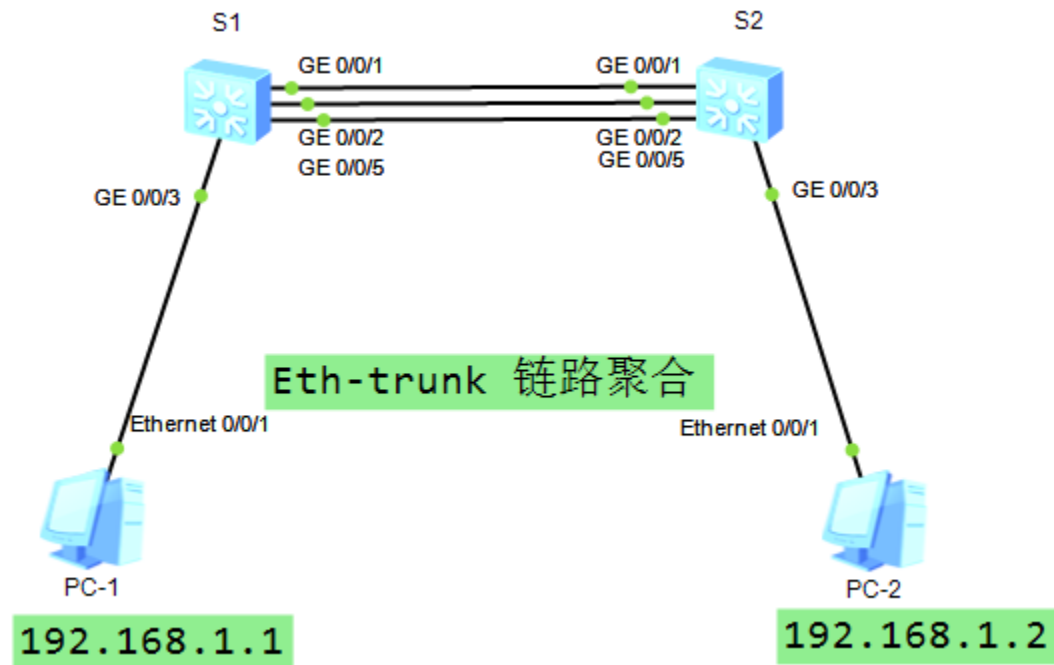
注意：

每个 eth-trunk 下最多可以支持 8 个成员端口;
成员接口下不能配置任何业务和静态 MAC 地址;
成员接口加入 Eth-trunk 时,必须为缺省的 hybrid 类型接口;
Eth-trunk 不可以嵌套,也就是说成员接口不能是 eth-trunk;
一个以太网接口只能加入到一个 eth-trunk;
一个 eth-trunk 中的成员接口必须相同类型的,比如 FE 口和 GE 口是不属于同一个聚合组的.
可以将不同接口板上的接口加入同一 eth-trunk;
如果本地设备使用了 Eth-Trunk , 与成员接口直连的对端接口也必须捆绑为 Eth-Trunk 接口
当成员接口速率不一致时,实际使用中速率小的成员接口会出现拥塞,导致丢包;
当成员接口加入 eth-trunk 后,学习 mac 地址时是按照 eth-trunk 来学习的,而不是按照成员接口来学的;

系统 lacp 优先级和接口 lacp 优先级:

用于选择主动端和被动端,系统 lacp 优先级越低越好,默认是 32768;如果优先级一样则选择 mac 小的;

接口 lacp 优先级用于确定活动链路,默认也是 32768,优先值越低,lacp 优先级越高.



Eth-Trunk 工作模式可以分为两种：

手动负载均衡（manual load-balance）模式和 LACP（Link Aggregation Control Protocol）模式

手工负载分担模式：需要手动创建链路聚合组，并配置多个接口加入到所创建的 Eth-Trunk 中

静态 LACP 模式：该模式通过 LACP 协议协商 Eth-Trunk 参数后自主选择活动接口。

手工负载分担模式

在 SW1 和 SW2 上配置链路聚合，创建 Eth-Trunk 1 接口，并指定为手工负载分担模式。

将 g0/0/1 和 g0/0/2 加入到 Eth-Trunk 1 接口

```
SW1 :  
undo ter mo  
sys  
sysname SW1  
user-interface console 0  
idle-timeout 0 0  
int Eth-Trunk 1  
mode manual load-balance  
int g0/0/1  
eth-trunk 1  
int g0/0/2  
eth-trunk 1  
q
```

```
SW2 :  
undo ter mo  
sys  
sysname SW2  
user-interface console 0  
idle-timeout 0 0  
int Eth-Trunk 1  
mode manual load-balance  
int g0/0/1  
eth-trunk 1  
int g0/0/2  
eth-trunk 1  
q
```

或者在 eth-trunk 组下，加成员
int eth-trunk 1
trunkport g0/0/5

配置完成后，查看 Eth-Trunk 1 接口状态，可以看到工作

模式为 normal（手工负载分担方式），g0/0/1 和 g0/0/2 接口已经添加到 Eth-Trunk 1 中，并且处于 UP 状态

<SW1>display eth-trunk 1

```
[SW1]dis eth-trunk 1
```

Eth-Trunk1's state information is:

WorkingMode: **NORMAL** Hash arithmetic: According to SIP-XOR-DIP

Least Active-linknumber: 1 Max Bandwidth-affected-linknumber: 8

Operate status: up Number Of Up Port In Trunk: 2

```
-----  
PortName           Status    Weight  
GigabitEthernet0/0/1    Up        1  
GigabitEthernet0/0/2    Up        1
```

可以看到目前该接口的总带宽，是 g0/0/1 和 g0/0/2 接口带宽之和

<SW1>display interface Eth-Trunk 1

```
[SW1]display int Eth-Trunk 1
Eth-Trunk1 current state : UP
Line protocol current state : UP
Description:
Switch Port, PVID :    1, Hash arithmetic : According to SIP-XOR-DIP,1
  2G, Current BW: 2G, The Maximum Frame Length is 9216
IP Sending Frames' Format is PKTFMT_ETHNT_2, Hardware address is 4c1f.
Current system time: 2018-05-04 21:31:47-08:00
    Input bandwidth utilization :    0%
    Output bandwidth utilization :    0%
```

PortName	Status	Weight
GigabitEthernet0/0/1	UP	1
GigabitEthernet0/0/2	UP	1

查看 SW1 接口的生成树状态，可以看到 SW1 的两个接口被捆绑成为一个 Eth-Trunk 接口，并且该接口处于转发状态。

```
<SW1>display stp brief
```

```
[SW1]dis stp brief
```

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/3	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/5	DESI	FORWARDING	NONE
0	Eth-Trunk1	DESI	FORWARDING	NONE

当链路故障发生时，链路可以进行切换，并且只要物理链路有一条是正常的，Eth-Trunk 接口就不会断开，仍然可以保证数据的转发。

交换机关闭生成树协议

SW1:

stp disable

SW2:

stp disable

PC>ping 192.168.1.2 -t

```
From 192.168.1.2: bytes=32 seq=10 ttl=128 time=47 ms
From 192.168.1.2: bytes=32 seq=11 ttl=128 time=109 ms
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
From 192.168.1.2: bytes=32 seq=28 ttl=128 time=31 ms
From 192.168.1.2: bytes=32 seq=29 ttl=128 time=79 ms
```

=====

配置 Eth-Trunk 实现链路聚合（静态 LACP 模式）

现网络管理员为公司再部署一条链路作为备份链路，并采用静态 LACP 模式配置 Eth-Trunk 实现两条链同时转发，一条链路备份，当其中一条转发链路出现问题时，备份链路可立即进行数据转发。

需要将先前已经加入到 Eth-Trunk 接口下的物理接口先删除，之后在 SW1 SW2 上的 Eth-Trunk 接口下，将工作模式改为

静态 LACP 模式

SW1 :

```
int g0/0/1
undo eth-trunk
int g0/0/2
undo eth-trunk
q
int Eth-Trunk 1
mode lacp-static
int g0/0/1
eth-trunk 1
int g0/0/2
eth-trunk 1
int g0/0/5
eth-trunk 1
q
```

SW2 :

```
int g0/0/1
undo eth-trunk
int g0/0/2
undo eth-trunk
q
int Eth-Trunk 1
mode lacp-static
int g0/0/1
eth-trunk 1
int g0/0/2
eth-trunk 1
int g0/0/5
eth-trunk 1
q
```

配置完成后，查看 SW1 的 Eth-Trunk 1 的接口状态，可以看到 3 个接口默认都处于活动状态 (Selected)

<SW1>display eth-trunk 1

```
[SW1]dis eth-trunk 1
```

Eth-Trunk1's state information is:

Local:

LAG ID: 1 WorkingMode: **STATIC**
Preempt Delay: Disabled Hash arithmetic: According to SIP-XOR-DIP
System Priority: 32768 System ID: 4c1f-cc8d-3d21
Least Active-linknumber: 1 Max Active-linknumber: 8
Operate status: up Number Of Up Port In Trunk: 3

```
-----  
ActorPortName        Status   PortType PortPri PortNo PortKey PortState Weight  
GigabitEthernet0/0/1   Selected   1GE       32768    2       305       10111100   1  
GigabitEthernet0/0/2   Selected   1GE       32768    3       305       10111100   1  
GigabitEthernet0/0/5   Selected   1GE       32768    6       305       10111100   1
```

将 SW1 的系统优先级从默认的 32768 改为 100，使其成为主动端口（值越低优先级越高），并按照主动端口的接口来选择活动接口。两端设备选出主动端后，两端都会以主动端的接口优先级来选择活动端口。两端选择了一致的活动接口，活动链路组便可以建立起来，设置这些活动链路以负载分担的方式转发数据。

SW1 系统优先级为 100，活动接口上限阈值为 2，设置 g0/0/1、g0/0/5 接口优先级为 100

SW1 :

```
lacp priority 100
```

```
int eth-trunk 1
max active-linknumber 2
int g0/0/1
lacp priority 100
int g0/0/5
lacp priority 100
q
```

配置完成后，可以看到 g0/0/5 没有被选择，因为默认是不抢占的

优先级低，但没有开启抢占，所以没有被选择

```
[SW1]dis eth-trunk 1
```

Eth-Trunk1's state information is:

Local:

```
LAG ID: 1                      WorkingMode: STATIC
Preempt Delay: Disabled        Hash arithmetic: According to SIP-XOR-DIP
System Priority: 100           System ID: 4c1f-cc8d-3d21
Least Active-linknumber: 1    Max Active-linknumber: 2
Operate status: up           Number Of Up Port In Trunk: 2
```

ActorPortName	Status	PortType	PortPri	PortNo	PortKey	PortState	Wei
GigabitEthernet0/0/1	Selected	1GE	100	2	305	10111100	1
GigabitEthernet0/0/2	Selected	1GE	32768	3	305	10111100	1
GigabitEthernet0/0/5	Unselect	1GE	100	6	305	10100000	1

=====

改变抢占方式

在 lacp 模式下，默认是不抢占的，

开启抢占后，默认的抢占延时为 30s

```
int eth-Trunk 1
lacp preempt enable
lacp preempt delay 10
```

配置完成后，查看一下 <SW1>display eth-trunk 1

```
[SW1]dis eth-trunk 1
Eth-Trunk1's state information is:
Local:
LAG ID: 1                      WorkingMode: STATIC
Preempt Delay Time: 10         Hash arithmetic: According to SIP-XOR-DIP
System Priority: 100           System ID: 4c1f-cc8d-3d21
Least Active-linknumber: 1     Max Active-linknumber: 2
Operate status: up            Number Of Up Port In Trunk: 2

-----
ActorPortName      Status  PortType PortPri PortNo PortKey PortState Weight
GigabitEthernet0/0/1 Selected 1GE      100    2      305     10111100 1
GigabitEthernet0/0/2 Unselect 1GE      32768  3      305     10100000 1
GigabitEthernet0/0/5 Selected 1GE      100    6      305     10111100 1

=====
```

改变负载分担模式

默认是基于源目 IP 地址进行负载分担，可以根据需要进行改变

```
[SW1]dis eth-trunk 1
Eth-Trunk1's state information is:
Local:
LAG ID: 1                      WorkingMode: STATIC
Preempt Delay Time: 10         Hash arithmetic: According to SIP-XOR-DIP
System Priority: 100           System ID: 4c1f-cc8d-3d21
Least Active-linknumber: 1     Max Active-linknumber: 2
Operate status: up             Number Of Up Port In Trunk: 2
```

```
SW1 :
int eth-trunk 1
load-balance dst-ip
```

=====

配置 Eth-Trunk 接口的散列依据

缺省情况下，当 Eth-Trunk 接口根据 IP 进行散列

```
load-balance packet
```

说明：

基于 IP 的散列算法能保证包顺序，但不能保证带宽利用率。
基于包的散列算法能保证带宽利用率，但不能保证包的顺序。

配置 Eth-Trunk 成员接口的负载分担权重

```
int g0/0/1
distribute-weight 2
```

缺省情况下，成员接口的负载分担权重为 1