HCRSE113-HA&网管

BFD 的检测机制：

BFD 的检测机制是两个系统建立 BFD 会话，并沿它们之间的路径周期性发送 BFD 控制报文，如果一方在既定的时间内没有收到 BFD 控制报文，则认为路径上发生了故障，BFD 控制报文是 UDP 报文，端口号 3784。
BFD 控制报文采用 UDP 封装，目的端口号为 3784，源端口号在 49152 到 65535 的范围内。

BFD 提供异步检测模式。在这种模式下，系统之间相互周期性（默认为 1s ）地发送 BFD 控制报文，如果某个系统连续 3 个报文都没有接收到，就认为此 BFD 会话的状态是 Down。

表1 BFD参数缺省值

参数	缺省值
全局BFD功能	未使能
发送间隔	1000毫秒
接收间隔	1000毫秒
本地检测倍数	3
等待恢复时间	0分钟
会话延迟Up时间	0秒钟
BFD报文优先级	7

BFD 会话状态:

Down：会话处于 Down 状态或刚刚创建。
Init：已经能够与对端系统通信，本端希望使会话进入 Up 状态。
Up：会话已经建立成功。
AdminDown：会话处于管理性 Down 状态。

bfd vrrp
shutdown

常见命令

检测 IP 链路分为单跳检测和多跳检测
bfd 命令用来在系统视图下全局使能 BFD 功能，并进入 BFD 全局视图。
bfd bind peer-ip 命令用来创建 BFD 会话绑定，并生成 BFD 会话。
discriminator 命令用来配置当前 BFD 会话的本地标识符和远端标识符。
commit 命令用来提交 BFD 会话配置。

根据对端设备是否支持 BFD，分为 2 种情形：
1、对端支持 BFD 时，创建两端协商 BFD 参数、两端都上送 MPU 才可建立的 BFD 会话；
2、对端不支持 BFD 时，创建 BFD 单臂回声。

与接口状态联动
bfd 命令用来在系统视图下全局使能 BFD 功能，并进入 BFD 全局视图。
bfd bind peer-ip default-ip 命令用来创建检测链路物理状态的 BFD 会话绑定。
discriminator 命令用来配置当前 BFD 会话的本地标识符和远端标识符。
process-interface-status 命令用来配置当前 BFD 会话与其绑定的接口进行状态联动。

NSF　NSR

NSF（Non-Stopping Forwarding，不间断转发）
当由于某种原因系统发生故障时，在主备倒换过程中，转发平面（业务）不中断。
系统恢复后，设备能够重新建立邻居关系，从邻居处获取路由信息并重建路由表。

NSR（ Non-Stopping Routing，不间断路由）：通过协议备份机制，实现主备倒换时控制平面（路由）和转发平面（业务）均不中断。在设备发生倒换的过程中，路由处理不中断，因为：邻居和拓扑信息不丢失

在主备倒换端，系统支持 NSR 和 GR 两种不同的高可靠性保护，他们是互斥的。即，对于一个特定协议，系统倒换后，只能采用 NSR 或 GR 两种处理方式的一种。

SNMP
SNMPv1、SNMPv2c 和 SNMPv3 工作原理基本一致。
SNMP 基本操作（SNMPv1 操作）
get-request：从代理进程处提取一个或多个参数值。
get-next-request：从代理进程处提取紧跟当前参数值的下一个参数值。
set-request：设置代理进程的一个或多个参数值。
response：返回的一个或多个参数值。这个操作是由代理进程发出的，它是对前面 3 种操作的响应。
trap：代理进程主动发出的报文，通知管理进程有某些事件发生。

SNMPv2c 新增操作
getbulk-request：实现了 NMS 对被管理设备的信息群查询。基于 GetNext 实现，相当于连续执行多次 GetNext 操作。在

NMS 上可以设置被管理设备在一次 GetBulk 报文交互时，执行 GetNext 操作的次数。
Inform-request：被管理设备向 NMS 主动发送告警。与 Trap 告警不同的是，被管理设备发送 Inform 告警后，需要 NMS 回复 InformResponse 来进行确认。

SNMPv3

SNMPv1 和 SNMPv2c 的安全性较弱。SNMPv3 增加了身份验证和加密处理。
NMS 向 Agent 发送不带安全参数的 Get 请求报文，同时向 Agent 获取安全参数（SNMP 实体引擎的相关信息、用户名、认证参数、加密参数等）。
Agent 响应 NMS 的请求，并向 NMS 反馈请求的参数。
NMS 再次向 Agent 发送带安全参数（NMS 通过配置的算法计算出的用于身份认证的认证参数和用于报文加密的加密参数）的 Get 请求报文。
Agent 首先对消息进行认证，S 然后解密报文信息；对响应消息进行加密，并向 NMS 反馈。

前言

- 网络仅仅满足数据连通还不够，还需要考虑业务的不间断性，这就需要网络具备高可靠性和高可用性，使网络能够在发生故障时不影响业务的运行，同时能够快速定位故障解决问题。针对上述需求，本章节将介绍网络高可靠性技术及网管协议。

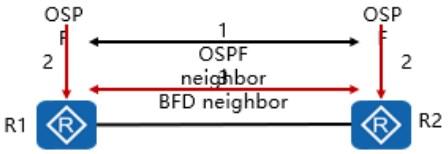
HA概述

- HA（High availability，高可用性）
  - 定义：指一个产品或系统具有很高的可用性
  - 高可用性特点
    - 不能频繁出现故障
    - 出现故障后能快速恢复
  - 高可用性衡量标准
    - MTBF，一个组件或设备的无故障运行平均时间
    - MTTR，一个组件或设备从故障到恢复正常所需的平均时间

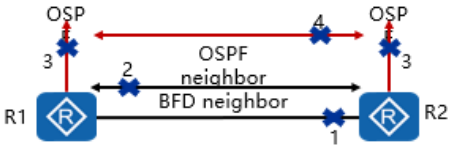
$$\text{系统可用性} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

BFD - 工作流程

- BFD（Bidirectional Forwarding Detection，双向转发检测）基本概念
  - 用于快速检测、监控网络中链路或者IP 路由的转发连通状况。
- BFD会话建立



- BFD故障检测

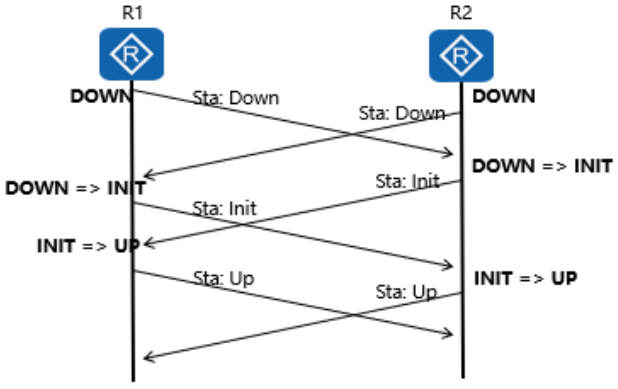


- BFD 基本概念
- BFD 在两台网络设备上建立会话，用来检测网络设备间的双向转发路径，为上层应用服务。BFD 本身并没有邻居发现机制，而是靠被服务的上层应用通知其邻居信息以建立会话。会话建立后会周期性地快速发送 BFD 报文，如果在检测时间内没有收到 BFD 报文则认为该双向转发路径发生了故障，通知被服务的上层应用进行相应的处理。

- BFD 控制报文采用 UDP 封装，目的端口号为 3784，源端口号在 49152 到 65535 的范围内。
- BFD 会话建立
- OSPF 通过自己的 Hello 机制发现邻居并建立连接。
- OSPF 在建立了新的邻居关系后，将邻居信息（包括目的地址和源地址等）通告给 BFD。
- BFD 根据收到的邻居信息建立会话。
- 会话建立以后，BFD 开始检测链路故障，并做出快速反应。
- BFD 故障检测
- 被检测链路出现故障。
- BFD 快速检测到链路故障，BFD 会话状态变为 Down。
- BFD 通知本地 OSPF 进程 BFD 邻居不可达。
- 本地 OSPF 进程中断 OSPF 邻居关系。

BFD - 会话管理

- BFD会话管理
  - BFD会话包含Down、Init、Up和AdminDown
  - BFD状态机的建立和拆除都采用三次握手机制



- BFD 会话状态
- Down：会话处于 Down 状态或刚刚创建。
- Init：已经能够与对端系统通信，本端希望使会话进入 Up 状态。

- Up：会话已经建立成功。
- AdminDown：会话处于管理性 Down 状态。
- BFD 会话迁移过程
- R1 和 R2 各自启动 BFD 状态机，初始状态为 Down，发送状态为 Down 的 BFD 报文。
- R2 收到状态为 Down 的 BFD 报文后，状态切换至 Init，并发送状态为 Init 的 BFD 报文。
- R2 本地 BFD 状态为 Init 后，不再处理接收到的状态为 Down 的报文。
- R1 的 BFD 状态变化同 R2。
- R2 收到状态为 Init 的 BFD 报文后，本地状态切换至 Up。
- R1 的 BFD 状态变化同 R2。

BFD - 应用场景

- BFD应用场景
  - 检测IP链路
  - 与接口状态联动
  - 与静态路由联动
  - 与OSPF联动
  - 与ISIS联动
  - 与BGP联动
  - 与MPLS LSP联动
  - 与VRRP联动
  - 与PIM联动

- 常见命令
- 检测 IP 链路分为单跳检测和多跳检测
- bfd 命令用来在系统视图下全局使能 BFD 功能，并进入

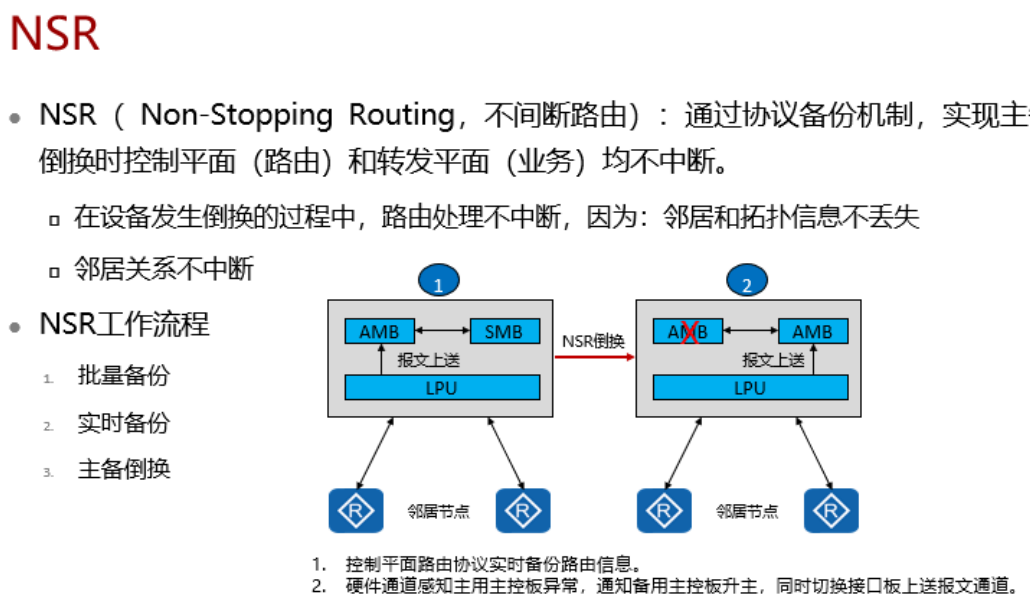
BFD 全局视图。

- bfd bind peer-ip 命令用来创建 BFD 会话绑定，并生成 BFD 会话。
- discriminator 命令用来配置当前 BFD 会话的本地标识符和远端标识符。
- commit 命令用来提交 BFD 会话配置。
- 根据对端设备是否支持 BFD，分为 2 种情形：1、对端支持 BFD 时，创建两端协商 BFD 参数、两端都上送 MPU 才可建立的 BFD 会话；2、对端不支持 BFD 时，创建 BFD 单臂回声。
- 与接口状态联动
- bfd 命令用来在系统视图下全局使能 BFD 功能，并进入 BFD 全局视图。
- bfd bind peer-ip default-ip 命令用来创建检测链路物理状态的 BFD 会话绑定。
- discriminator 命令用来配置当前 BFD 会话的本地标识符和远端标识符。
- process-interface-status 命令用来配置当前 BFD 会话与其绑定的接口进行状态联动。
- 与接口状态联动与路由联动的配置十分相近，这里不进行描述。

NSF

- NSF（Non-Stopping Forwarding，不间断转发）
  - 当由于某种原因系统发生故障时，在主备倒换过程中，转发平面（业务）不中断。
  - 系统恢复后，设备能够重新建立邻居关系，从邻居处获取路由信息并重建路由表。
- NSF需要满足条件
  - 硬件要求，系统双主控RP冗余配置。
  - 软件要求，需要主备板卡同步。
  - 协议要求，需要网络协议支持GR。

- 通常情况下，路由器故障后，其路由协议层面的邻居会检测到它们之间的邻居关系 Down 掉，然后过段时间再次 Up，这个过程被称之为邻居关系震荡。这种邻居关系的震荡将最终导致路由震荡的出现，使得重启路由器在一段时间内出现路由黑洞或者导致邻居将数据业务从重启路由器处旁路，从而导致网络的可靠性大大降低。不间断转发技术的目标就是为了解决上述路由震荡的问题，为此，需要满足以下要求：
- 硬件要求：系统双主控 RP 冗余配置，即一块做主用主控板，一块做备用主控板，主用主控硬件要求：系统双主控 RP 冗余配置，即一块做主用主控板，一块做备用主控板，主用主控板重启，备用主控板成为新的主板；分布式结构，数据转发和控制分离，有专门的线卡（接口板）用于数据转发。
- 系统软件要求：主板正常运行的过程中，会把配置信息、接口状态信息备份到备用板；主备倒换的时候，接口板不需要重启，接口保持 Up，接口板转发表不撤销。
- 协议要求：要求各相关网络协议，如路由协议 OSPF、IS-IS、BGP，其他协议如 LDP 做扩展，具备优雅重启（GR）能力。

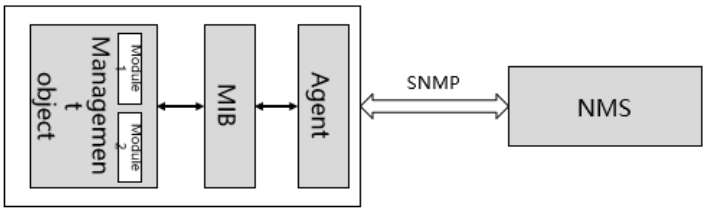


- NSR 优点

- 不依赖也不影响对端设备，没有互通问题。
- 路由的收敛速度要比 NSF 快。
- NSR 工作流程
- 批量备份：NSR 功能使能后，备板复位重启时，主用主控板上的业务进程会收到备用主控板上线的消息。业务进程开始进行内部数据的批量备份。
- 批量数据备份完毕后，系统进入冗余保护状态。进入该状态后，如果主控板出现故障，备板升主后就可以利用之前从主板备份过来的数据进行升主，恢复业务。
- 如果业务批量备份尚未结束时，主控板故障，备板升主后可能会因为业务数据不全而导致无法升主，因此这个种状态下无法完成 NSR 倒换，设备会整机重启，恢复故障前状态。
- 完成批量备份后，系统进入实时备份阶段，在该阶段当邻居状态或路由信息发生时，主用主控板会实时将变化信息备份到备用主控板。
- 完成批量备份，进入冗余保护状态的系统，当主用主控板发生软件或硬件故障后，备用主控板会从底层应该感知到主用主控板的故障，并自行升主。升主后业务进程会使用之前从主用主控板备份来的数据进行工作。同时也会向接口板平滑倒换期间变化的信息。实现真正的路由不中断，转发不中断。
- 在主备倒换端，系统支持 NSR 和 GR 两种不同的高可靠性保护，他们是互斥的。即，对于一个特定协议，系统倒换后，只能采用 NSR 或 GR 两种处理方式的一种。

SNMP - 基本概念

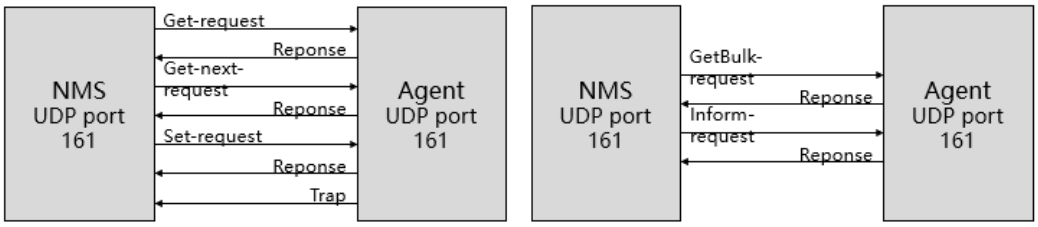
- SNMP（Simple Network Management Protocol，简单网络管理协议）基本概念
  - SNMP提供了一种通过运行网络管理软件的中心计算机（即网络管理工作站）来管理设备的方法。
- SNMP版本
  - SNMPv1、SNMPv2c和SNMPv3
- SNMP模型



- SNMP 模型
- 网络管理站 NMS（Network Management Station）：NMS 通常是一个独立的设备，运行网络管理应用程序。网络管理应用程序至少能够提供一个人机交互界面，网络管理员通过它完成绝大多数网络管理工作。
- SNMP 代理器（Agent）：Agent 是驻留在被管理设备的一个软件模块，主要负责接收和处理来自 NMS 的请求报文，并形成响应报文，返回给 NMS；在一些紧急情况下，它会主动发送 Trap 报文，通知 NMS。
- SNMP 协议：SNMP 协议属于 TCP/IP 网络的应用层协议，用于在 NMS 和被管理设备间交互管理信息。
- 管理信息库 MIB（Management Information Base）：MIB 是一个被管理对象的集合，是 NMS 同 Agent 进行沟通的桥梁，可以使网管软件和设备进行标准对接。每一个 Agent 都维护这样一个 MIB 库，NMS 可以对 MIB 库中对象的值进行读取或设置。
- Management object 指被管理对象。每一个设备可能包含多个被管理对象，被管理对象可以是设备中的某个硬件（如一块接口板），也可以是某些硬件，软件（如路由选择协议）及其的配置参数的集合。

SNMP - SNMPv1/SNMPv2c工作原理

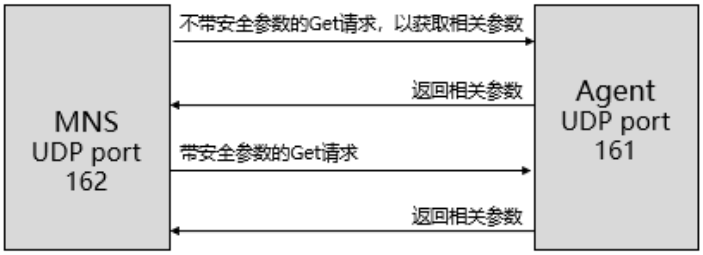
- SNMPv1、SNMPv2c和SNMPv3工作原理基本一致。
- SNMP基本操作（SNMPv1操作）
- SNMPv2c新增操作



- 基本操作：
- get-request：从代理进程处提取一个或多个参数值。
- get-next-request：从代理进程处提取紧跟当前参数值的下一个参数值。
- set-request：设置代理进程的一个或多个参数值。
- response：返回的一个或多个参数值。这个操作是由代理进程发出的，它是对前面 3 种操作的响应。
- trap：代理进程主动发出的报文，通知管理进程有某些事件发生。
- SNMPv2c 新增操作类
- getbulk-request：实现了 NMS 对被管理设备的信息群查询。基于 GetNext 实现，相当于连续执行多次 GetNext 操作。在 NMS 上可以设置被管理设备在一次 GetBulk 报文交互时，执行 GetNext 操作的次数。
- Inform-request：被管理设备向 NMS 主动发送告警。与 Trap 告警不同的是，被管理设备发送 Inform 告警后，需要 NMS 回复 InformResponse 来进行确认。
- SNMPv1 和 SNMPv2c 的安全性较弱。

SNMP - SNMPv3工作原理

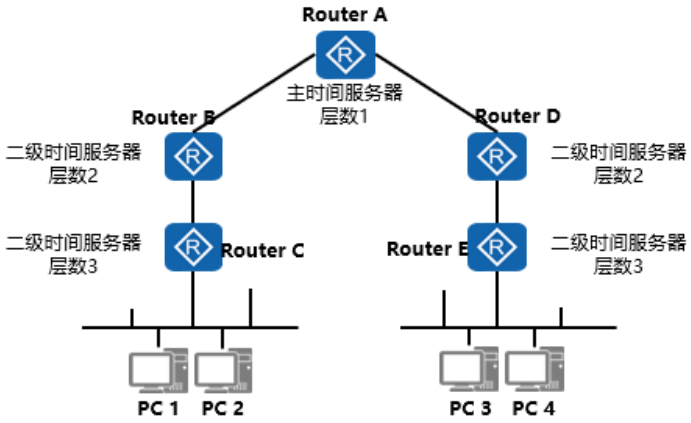
- SNMPv3增加了身份验证和加密处理。



- SNMPv3 的实现原理与 SNMPv1 和 SNMPv2c 基本一致。
- SNMPv3 工作原理
- NMS 向 Agent 发送不带安全参数的 Get 请求报文，同时向 Agent 获取安全参数（SNMP 实体引擎的相关信息、用户名、认证参数、加密参数等）。
- Agent 响应 NMS 的请求，并向 NMS 反馈请求的参数。
- NMS 再次向 Agent 发送带安全参数（NMS 通过配置的算法计算出的用于身份认证的认证参数和用于报文加密的加密参数）的 Get 请求报文。
- Agent 首先对消息进行认证，然后解密报文信息；对响应消息进行加密，并向 NMS 反馈。

NTP - 网络结构

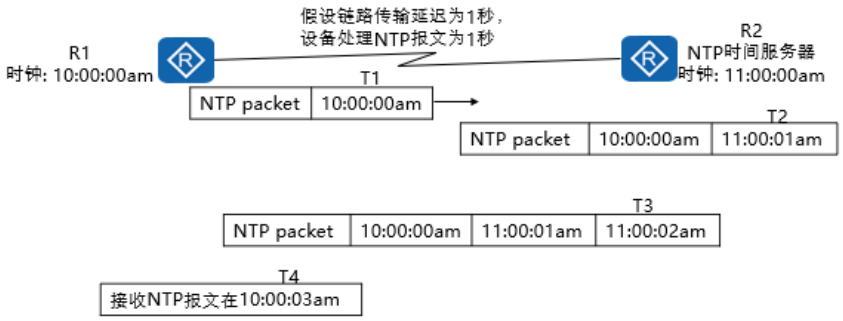
- NTP网络结构
  - 同步子网
  - 主时间服务器
  - 二级时间服务器
  - 层数（stratum）



- 在 NTP 的网络结构中，主要存在如下概念：
- 同步子网：由主时间服务器、二级时间服务器、PC 客户端和它们之间互连的传输路径组成同步子网。
- 主时间服务器：通过线缆或无线电直接同步到标准参考时钟，标准参考时钟通常是 Radio Clock 或卫星定位系统等。
- 二级时间服务器：通过网络中的主时间服务器或者其他二级服务器取得同步。二级时间服务器通过 NTP 将时间信息传送到局域网内部的其它主机。
- 层数（stratum）：层数是对时钟同步情况的一个分级标准，代表了一个时钟的精确度，取值范围 1～16，数值越小，精确度越高。1 表示时钟精确度最高，16 表示未同步。
- 在正常情况下，同步子网中的主时间服务器和二级时间服务器呈现出一种分层主从结构。在这种分层结构中，主时间服务器位于根部，二级时间服务器向叶子节点靠近，层数递增，准确性递减，降低的程度取决于网络路径和本地时钟的稳定性。

NTP - 工作原理

- NTP
  - 用于在一系列分布式时间服务器与客户端之间同步时钟。
  - NTP报文通过UDP传输，端口号是123。



- NTP 的同步流程
- R1 发送一个 NTP 报文给 R2，该报文中带有它离开 R1 时的时间戳 10:00:00a.m.（T1）。
- 此 NTP 报文到达 R2 时，R2 加上到达时间戳 11:00:01a.m.（T2）。
- 此 NTP 报文离开 R2 时，R2 再加上离开时间戳 11:00:02a.m.（T3）。
- R1 接收到该响应报文时，加上新的时间戳 10:00:03a.m.（T4）。至此，RouterA 获得了足够信息来计算以下两个重要参数：
- NTP 报文来回一个周期的时延：Delay=（T4 - T1）-（T3 – T2）。
- R1 相对 R2 的时间差：Offset=（（T2- T1）+（T3 – T4））/ 2。
- R1 根据计算得到 Delay 为 2 秒，Offset 为 1 小时。R1 根据这些信息来设定自己的时钟，实现与 R2 的时钟同步。

思考题

1. NSF与NSR共同使用，更好地保障业务连续运行。（ ）

   A. T

   B. F

- 参考答案：
- B。

本章总结

- 高可靠性技术
  - BFD
  - NSF
  - NSR
- 网络管理技术
  - SNMP
  - NTP