计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(11):266−270 [doi: 10.15888/j.cnki.csa.006072]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 ICMP 扩展的链路状态检测方法①

周　明, 郦苏丹, 王　宏

(国防科技大学 计算机学院学院, 长沙 410073)

摘　要: 传统的 ICMP 在主机存活检测、端口扫描及网络拓扑发现等网络信息获取方面已经有了广泛应用. 但检测信息单一, 方法不灵活, 网络局限性大等问题依然突出. 本文就基于 ICMP 协议提出了一种携带链路接口信息的连通性检测方法. 主要是在原始 ICMP 协议的基础上, 增加一个可变长度的链路状态字段, 用于存储接口设备标识和带宽负载. 论文重点解决如何利用 ICMP 回显应答报文携带传输这些接口信息给源端以及中间节点的接收处理. 通过此方法能有效的帮助我们了解整个网络拓扑和带宽延迟, 填补了传统的连通性检测方法缺少网络链路状态信息这一空白.

关键词: ICMP; 链路状态; 接口信息; 网络拓扑; 连通性检测

引用格式: 周明,郦苏丹,王宏.基于 ICMP 扩展的链路状态检测方法.计算机系统应用,2017,26(11):266–270. http://www.c-s-a.org.cn/1003-3254/6072.html

Link State Detection Protocol Based on ICMP Extension

ZHOU Ming, LI Su-Dan, WANG Hong

(Department of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: The conventional ICMP has been widely used in network information acquisition such as host survivability detection, port scanning and network topology discovery. But the problems like less detection information, the inflexible method, and limitations in network, and other issues are still prominent. In this paper, we propose a connectivity detection method based on ICMP protocol to carry link information. Mainly based on the original ICMP protocol, we add a variable-length link-state field for storage of interface device identification and bandwidth load. This paper focuses on how to use the ICMP Echo Reply to carry these interface information to the source and intermediate nodes of the receiving process. This method can effectively help us understand the entire network topology and bandwidth delay, filling the gap that the traditional connectivity detection method lacks network link state information.

Key words: ICMP; link information; interface information; network topology; connectivity detection

链路状态协议主要用于获取网络拓扑的链路状态信息, 其功能包括信息获取、性能监测和流量控制等[1]. 目前的网络链路状态协议主要有: Cisco 发现协议 (CDP)、以太网 OAM 技术 (EFMOAM)、网际控制报文协议 (ICMP). 这些协议基本为我们链路状态检测提供了标准和方法, 但就为网络维护提供更深层次的信息推断网络拓扑结构和了解链路状态来说, 还不能满足我们的要求.

CDP 即 Cisco 发现协议, 是一种数据链路层的、与设备和协议无关的链路状态协议. 它能极大的帮助我们了解网络状态、排除故障, 是理解网络拓扑最好的方法之一[2]. 通过 CDF 不同命令操作, 可以为网络管理员获取相关设备或直连到交换机的如设备标识、地址列表、接口类型、端口标识等网络信息. 具有网络

① 收稿时间: 2017-02-19; 修改时间: 2017-03-23; 采用时间: 2017-03-27

开销低, 灵活性强, 发现速度快, 获取信息细等特点. 但由于 CDP 仅运行在 cisco 设备, 而现实中却是各种设备混合的复杂型网络. 单单使用 CDF 协议来发现网络拓扑, 将会造成网络的不完整, 从而导致整个网络拓扑的价值大打折扣, 实用性不强.

以太网 OAM 作为以太网络发展的建立的管理维护机制, 用以提高以太网的管理和维护能力, 保证网络的稳定运行[3], 其功能包括: 链路性能监测、故障侦测和告警、环路测试等. 也可以监测如带宽通量、帧丢失率、帧时延、帧时延变化以及最大速率等不同的性能参数[4]. 其扩展性强, 实用面广, 但随着网络节点增加, 用户网络接口的成指数型增长, 造成管理成本和难度的提高, 性能监视也将遇到极大的扩展性和性能瓶颈的问题.

ICMP 是 TCP/IP 协议中一种具有特殊用途的报文机制, 用于主机或路由器发送差错和控制报文的协议[5]. 其在网络中的主要用于: 主机探测、路由维护、路由选择、流量控制和交换状态信息. Ping 程序就是利用 ICMP 协议, 向目标主机发送 ICMP 的请求报文, 用以对主机存活性进行探测. 并根据返回的信息, 得到通往目的地的部分 IP 地址, 和依据发送和收到的时间差推断概要的网络连通情况. 但是目前 ICMP 协议只是提供简单的连通性探测, 而不支持链路状态检测[6], 因此利用传统的 ICMP 协议, 我们无法了解传输路径上的链路状态情况. 尽管如此, 因为 ICMP 协议非常灵活和快捷, 所以现目前的绝大多数设备和主机依然可以支持 ICMP 协议.

针对当前 ICMP 对网络链接信息提供不足的缺点, 本文提出了一种基于 ICMP 扩展的链路状态检测协议. 它主要用于源端获取目的端的设备标识、带宽负载等链路信息, 便于管理人员了解、掌握和维护整个网络拓扑, 具有很强扩展性和超高实用度, 同时也方便终端用户的使用[7]. 本论文主要就协议的设计和实现进行理论说明, 论文先对研究的背景进行介绍, 之后对协议的扩展提出设想, 最后对实现的过程和步骤进行详细的说明并指出存在的不足以及对以后工作的设想.

1 ICMP 协议扩展设计

1.1 ICMP 协议原理

ICMP 全称 Internet Control Message Protocol, 中文名为因特网控制报文协议. 它工作在 OSI 的网络层, 向数据通讯中的源主机报告错误. 因为网络本身是不可靠的, 在网络传输过程中, 可能会发生许多突发事件并导致数据传输失败[8]. 而网络层的 IP 协议又是一个无连接的协议, 它不会处理网络层传输中的故障, 所以在 IP/TCP 协议上增加了 ICMP 协议. 它使用 IP 协议进行信息传递, 向数据包中的源端节点提供发生在网络层的错误信息反馈.

ICMP 报文差错和请求处理基本涵盖了日常所能见到的各种情况, 其消息种类多达 15 种共 38 类, 由首部的类型 (type) 和代码 (code) 字段组合构成. 其主要报文有用于差错报告的差错报文、用于管理的控制报文以及请求/应答报文. 因功能的不同, 在数据报的结构上也有所不同, 图 1 是 ICMP 报文的一般格式.

<table>
<tr><td colspan="3">IP</td></tr>
<tr><td colspan="3">……</td></tr>
<tr><td>类型 (type)</td><td>代码 (code)</td><td>检验和</td></tr>
<tr><td colspan="2">标识符 (id)</td><td>序列号</td></tr>
<tr><td colspan="3">数据选项（根据消息类型, 是一个可变长度的数据）</td></tr>
</table>

图 1 ICMP 报文格式

类型字段用来指定 ICMP 的消息类型, 代码字段进一步定义了请求或错误消息的的具体划分. 而数据选项根据传递消息类型的不同存放的数据也有所不同. 在差错报文中, 数据部分存放引发差错的报文首部和数据报前 8 个字节. 在请求/应答报文中, 标示符 (id) 用来区分发给不同主机的 ICMP 请求消息, 而序列号则用来区分发给同一主机的不同 ICMP 请求报文[9]. ICMP 协议规定, ICMP 报文的头部是一个定长的空间, 而数据选项部分却是一个可变长度的数据区域, 这样带来了协议扩展的可能, 也为我们下一步的设计提供了依据.

1.2 扩展设计

传统的 ICMP 请求/应答报文是一种查询报文, 方法和原理都较简单. 发送方初始化 ICMP 报文的 ID 和序列号, 并在数据域中任意的加入一些数据, 发给接收方, ICMP 首部代码 code 字段设为 0. 接收方仅仅将收到的请求报文类型变为应答类型, 原文复制数据信息返回发送方.

我们提出方法中, 将在 ICMP 报头后面增加一个可变长度的链路状态信息字段, 用于填充应答报文回

送或返回链路节点故障时经过的每个节点的链路状态信息. 其主要结构如图 2 所示.

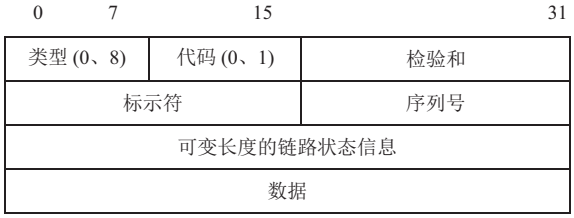

图 2 请求/应答报文结构

报文主要字段取值和定义如下:

报文类型: 8=回显请求消息; 0=回显应答消息.

代码: 为 0 时, 网络设备回显应答时不需要添加回显所在接口的链路状态信息; 为 1 时, 网络设备进行回显应答处理时需要添加回显所在接口的链路状态信息.

链路状态信息的定义格式:

typ	len	val

Typ 存储数据的类型, 占一个字节; 类型 0 表示终了 TLV, 其 len 字段和 val 字段均未使用, 仅是表明 TLV 最末端节点; 类型为 1, 表示存储的数据 TLV 为设备标识; 2 为端口标识; 3 为接口带宽, 以 kbps 为单位; 4 为接口负载, 表示端口的带宽占用率.

Len 记录存储数据的长度, 其长度值由数据存储的内存空间决定, 占一个字节.

Val 存储具体的数据内容, 按照字节的总长度, 以 4 字节对齐.

1.3 处理流程

在 ICMP 请求/应答报文中, 我们定义了代码为 1 时, 表明我们需要目标或转发的中间节点接口的状态信息. 当连通检测请求报文发往目标主机时, 因中间路由只做转发处理, 并不会理会 ICMP 报文内容[10]. 目标主机 ICMP 协议模块接收到该报文, 需对此报文进行处理. 按照正常的处理流程进入回显应答部分, 在该部分中增加一个判断. 如果 ICMP 代码字段为 1, ICMP 协议处理模块将本设备的接口信息填充到回显报文的链路状态信息字段, 重新计算 ICMP 的数据报文的检验和, 将报文交由 IP 层发送处理. 同样的, 回显报文在发回源主机时, 每经过一个路由节点, 路由器检查 ICMP 类型和代码, 如果为 ICMP_ECHOREPLY 和 1, 该路由也需要将其接口链路信息添加到状态信息字段里. 最后, 源主机接到回送回来的应答信息报文后, 逐条将回送回来的链路信息提取出来, 根据这些链路信息, 就能很好的了解整个转发网路的队列长度、接口带宽、带宽利用率等链路信息了. 其示意图如图 3.

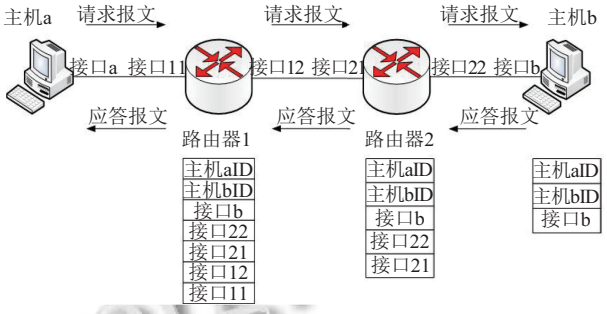

图 3 链路状态信息添加过程

主机 a 向目标主机 b 发送了一个代码为 1 的 ICMP 请求报文, 并等待主机 b 的回应. 主机 b 接收了 a 的请求报文, 先生成一个应答报文并将自己的接口信息添加到 TLV 字段里发回主机 a. 经过路由 2 时, 路由检查报文类型, 若是 ICMP 的应答报文且 code 字段为 1, 同样的将添加一个带有自己接口信息的 TLV 到原报文后面, 将此报文转发到下一跳路由 1. 重复过程, 直到到达源主机 a, 主机 a 检查报文是发给自己, 所以不用再添加自己的信息, 而是从接收到得报文中提取自己所需的信息, 完成整个过程.

2 协议实现

ICMP 报文作为 IP 的数据部分封装在 IP 报文中传输, 因此我们在 ICMP 报文的头部后面增加一个可变长度的 TLV 用于存储我们所需的接口链路状态信息[11]. 为新的 TLV 申请内存空间, 初始化 TLV 类型为 0. 目的主机应答 ICMP 报文时, 将源地址和目的地址对换, 查找路由表信息, 找出对应的设备接口 rt_dev, 调用函数 dev_ifconf() 获取设备的接口信息[4]. 将接口信息内容存储在 val 字段, 并依次设置 TLV 类型, 计算 val 在存储空间所占大小, 将其长度记录到 TLV 的 len 字段, 改变 tlvlen 长度为 len+16. 并将整段 tlv 内容添加到原数据报文 sk_buff 结构中数据缓存区域, 存储长度 skb→len 加上添加的数据长 tlvlen, 重新计算校验和. 当回送目标主机接收到携带有链路状态信息的报文时, 根据报文的标识和序列号确认是否为自己发送的链路状态请求报文, 找到 ICMP 报文的数据部分, 此时 skb→data 指向 ICMP 报头尾部, 这正好是最近存储

的一条 TLV 开始部分, 根据每段 TLV 中 len 字段值, 读取缓存区存放的 len+16 长度的 TLV 链路信息, skb→data 指向下一条 TLV 链路信息头. 以此类推, 直到取出所有携带的接口链路信息.

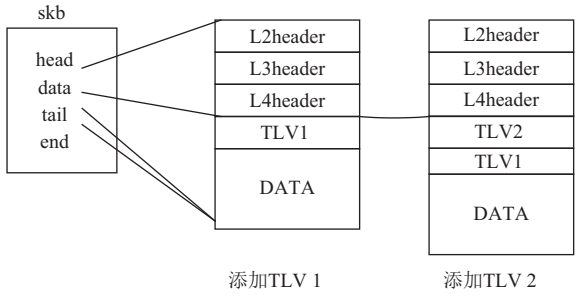

图 4 数据区 TLV 添加过程

图 4 显示了如何在一个报文中增加 TLV 数据内容. 根据当初的设想, 我们在 icmp_bxm 结构中增加一个 TLV 字段, 其定义为:

```
int tlvlen;
struct icmp_tlv {
        __u8        typ;
        __u8        len;
        unsigned char val;
} tlv;
```

其设计的伪代码思想为:

(1) ICMP 请求报文处理

1) 接收并检验报文

2) If 报文类型为 ECHO

复制原 skb 头并生成一个应答报文

3) If ICMP 代码为 1

状态添加处理

①查找发往目的地址的路由表

②根据路由表获取本机连接的接口信息

③为新的 TLV 申请内存空间

④复制接口信息到 val 字段里, 改变 len

⑤TLV 块添加到 icmp 头尾部

⑥计算 icmp 头部校验和

4) 计算校验和

5) 发送报文

(2) 中间路由接收 ICMP 回显报文

1) skb→head 指针指向 icmp 头

2) If icmp 代码 code 为 1

3) 调用状态添加处理将自己接口信息添加到 icmp 头尾部即上一个 TLV 头部

4) 计算校验和并转发该报文

(3) 源地址收到 ICMP 回显报文

1) 将 skb→data 指针指向 icmp 头尾部, 剥离报文头部, 得到 TLV 数据

2) or 每条 TLV 数据

If TLV 类型不为 0

复制从 skb→data 指针开始, 长度为 tlvlen 的报文数据

skb→data 指针加 tlvlen 长度的偏移

将复制数据返回给用户

Else

结束退出

此次设计仅是在目的主机可达的基础上通过回显应答的报文携带回路的接口链路信息, 但是在实际运用过程中, 假设路径设备故障, 致使网路不通, 目标主机无法接收到请求信息, 自然就不能返回我们想要的路径接口信息了. 再有, 因为需要转发网络中路由器添加接口的信息, 势必会造成路由器开销的增加, 影响路由器处理性能, 也带来核心路由信息泄露的安全风险. 当然这些都是我们后期需要考虑的问题.

3 结语

ICMP 作为 IP 协议为报告差错以及进行控制而专门衍生出来的同层协议, 其重要性不言而喻. 随着网路规模的壮大, 网路复杂性和问题的多样性也日见突显, 但 TCP/IP 协议栈的处理方式却几乎没有变化. 本文基于 ICMP 的原理扩展设计的携带有链路状态信息的连通性检测方法在一定程度上来说, 满足了目前对链路状态探知的需求, 也能更好的帮助我们了解整个网络拓扑的连通情况. 下一步, 我们将考虑在路由设备中进行 ICMP 协议的扩展, 兼容传统 ICMP 协议的基本功能, 并可实现到目的地址的链路状态检测, 从而帮助我们更详细和准确的掌握整个网络状态或精确定位网络故障.

参考文献

1 万民. 基于 ICMP 的网络状态监测研究. 科技信息, 2009, (31): 865, 904.

2 Peng T, Leckie C, Ramamohanarao K. Survey of network-based defense mechanisms countering the DoS and DDoS

problems. ACM Computing Surveys, 2007, 39(1): 3. [doi: 10.1145/1216370]
3 ITU-T Recommendation Y.1730. Requirements for OAM functions in ethernet-based networks and Ethernet services. 2004.
4 薛东. 基于 ICMP 的网络信息获取技术研究[硕士学位论文]. 西安: 西安电子科技大学, 2002.
5 樊东东, 莫澜. Linux 内核源码分析——TCP/IP 实现. 北京: 机械工业出版社, 2011.
6 胡延平, 王连杰, 刘武. 基于 ICMP 的网络性能分析. 计算机工程与设计, 2003, 24(4): 30–32.
7 Postel J. RFC 792 Internet control message protocol. IETF, 1981.
8 Karn P, Simpson W. RFC 2521 ICMP security failures messages. Internet Engineering Task Force, 1999.
9 Kent S, Atkinson R. RFC 2401 Security architecture for the internet protocol. Internet Engineering Task Force, 1998.
10 Wang HN, Jin C, Shin KG. Defense against spoofed IP traffic using hop-count filtering. IEEE/ACM Trans. on Networking, 2007, 15(1): 40–53. [doi: 10.1109/TNET.2006.890133]
11 Barbhuiya Fa, Roopa S, Ratti R, *et al*. An active detection mechanism for detecting ICMP based attacks. Proc. of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. Liverpool, Britain. 2012. 51–58.