

一种基于位置/标识分离的站点多宿主路径失效恢复机制

涂睿 苏金树 陈锋

(国防科技大学计算机学院 长沙 410073)

摘要 多宿主是克服路径失效,提高站点服务可靠性的有效途径之一。然而囿于当前 TCP/IP 体系结构的限制,多宿主还没有得到很好的普及。其中一个重要原因就是 IP 地址语义过载,IP 地址既代表了网络节点的拓扑位置又是节点的标识。在 LISA(Locator Identifier Split Architecture)命名与寻址体系结构下,提出了基于位置与标识分离的站点多宿主方案,并以此为基础提出了路径失效恢复机制:LISA-Recovery。模拟实验验证了该机制能够有效地检测路径失效和性能下降,并进行快速的路径切换,从而保障了上层应用服务的正常运行。通过理论计算,分析了 LISA-Recovery 所带来的开销很小,显示其具有较好的实用性。

关键词 路径失效,多宿主,位置/标识分离,路径切换

中图法分类号 TP393 **文献标识码** A

Site Multihoming Path Failure Recovery Mechanism Based on Locator/Identifier Split

TU Rui SU Jin-shu CHEN Feng

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

Abstract The multihoming is one of the effective ways to defeat path failure and increase the reliability of site network service. However, limited by the TCP/IP naming and addressing architecture, the multihoming has not been well deployed. One of the most important reasons is the problem of semantic overloading of IP address. Current IP address has dual semantic functions, which indicates both the network node's routing locator and its endpoint identifier. We proposed a sit multihoming path failure recovery mechanism(LISA-Recovery) based on the LISA(Locator Identifier Split Architecture) architecture. The results of simulation test prove that LISA-Recovery can effectively detect the path failure and performance reduction, and make a rapid path switch. It will greatly strengthen the reliability of application service. We also gave a theory analysis of the bandwidth overhead brought by LISA-Recovery. It shows that the bandwidth overhead is very small and our approach is feasible.

Keywords Path failure, Multihoming, Locator/ID Split, Path switch

1 引言

路径失效会导致上层网络应用的中断和服务质量下降,可能原因包括:链路中断、网络设备故障、拒绝服务攻击、突发流量导致的报文丢失或延迟增大等。路径失效的程度包括彻底失效和性能下降。彻底失效是比较严重的路径失效,导致大部分应用不能正常运行;性能下降则是指路径的性能有所降低,但还是在可以容忍的范围之内,具体表现在吞吐率降低、延迟增大和丢包率上升等方面。

目前基于路由协议的路由体系结构在路径失效的处理上是低效的,例如 BGP 协议通常需要几分钟才能从路径失效中恢复路由^[1],造成上层应用服务的中断。此外,路由协议主要考虑彻底失效的情况,很少考虑与上层应用性能息息相关的路径质量指标(延迟、报文丢失率等)的性能下降问题。对上层应用而言,路由协议选择的路径可能不是最佳路径。

多宿主(Multihoming)是减少路径失效对上层应用服务影响的重要途径之一。通过维护多条备选路径,在主路径失效时自动切换到备选路径,从而确保上层应用不中断。然而,由于当前 TCP/IP 网络体系结构的限制,所宿主机制还难以实现和部署。其中的一个重要原因就是 IP 地址过载问题(IP Overload)。在当前的 TCP/IP 协议中,IP 地址在语义上具有双重含义,既代表了网络节点的拓扑位置又是节点的标识。IP 地址过载不利于支持移动性,影响了互联网核心路由的扩展性,带来了一系列的安全性问题^[2]。此外,IP 地址过载还限制了多宿主、流量工程等新技术的发展^[3]。

为了解决 IP 地址过载问题,国际 IAB 组织提出引入两个名字空间来分别表示节点的位置和标识,即所谓的“Locator/Identifier Split”^[4]。引入“Locator/Identifier Split”之后,边缘网络的机构和用户可以利用 PI(Provider Independent)地址方便地实现多宿主、流量工程和任播,避免使用 PA(Pro-

到稿日期:2008-11-19 返修日期:2009-02-09 本文受国家 863 项目(2008AA01A325),国家重大基础研究发展计划(973)课题(2009CB320503)资助。

涂睿(1980-),男,博士生,主要研究方向为计算机网络等,E-mail:trbomb@vip.163.com;苏金树(1962-),男,教授,博士生导师,CCF 高级会员,主要研究方向为计算机网络、信息安全等;陈锋(1979-),男,博士生,主要研究方向为信息安全。

vider Allocated)地址所引发的地址重分配(Renumbering)所产生的代价,例如许多基于IP地址的访问控制策略、配置都需要改变等。

“Locator/Identifier Split”的命名和寻址方式为多宿主机制的实现和部署提供了现实基础。目前的多宿主机制主要有两种:站点多宿主和主机多宿主。站点多宿主支持较大的主机数目、复杂的路由网络,可以满足大型商业机构的需要,但端用户无法根据自己的需要实施流量工程和网络配置;主机多宿主针对单个主机,端用户可以对主机和备选网络进行配置,通过边缘网络接入其偏好的核心网路。我们的方案通过引入“Locator/Identifier Split”的思想,用于支持站点多宿主。在已完成的命名和寻址体系结构:LISA(Locator Identifier Split Architecture)框架下,为边缘网络提供多宿主功能,实现站点多宿主。以此为基础,通过路径失效的检测、恢复机制(LISA-Recovery),保障站点内服务的可靠性。

本文第2节介绍目前在路径失效的快速检测和恢复方面的相关研究情况;第3节简要介绍我们之前提出的位置与标识分离的命名和寻址体系结构:LISA;第4节详细论述本文提出的在LISA体系结构下的站点多宿主路径失效恢复机制:LISA-Recovery;第5节通过模拟的方式对LISA-Recovery进行功能验证和性能分析;最后对本文的工作进行了总结,并展望了下一步工作。

2 相关工作

为了提高应用服务的可靠性,研究人员一方面试图加快路由协议的收敛,一方面也试图通过层叠网、多宿主等方式来实现路径失效的快速检测和恢复。

RON^[5]是MIT实验室的David Andersen等人于2001年提出的基于层叠网的路由基础设施,它可以使分布式的网络应用检测到路径的失效和周期性性能降低现象,并能够把恢复时间限制在秒级范围内。利用RON的选路机制,可以为用户提供更有弹性和容错能力的服务业务。RON是一种在现有的互联网网络层上的应用层层叠网,RON节点可以监控它们之间路径的质量,并根据这些信息决定是直接利用Internet转发IP报文还是通过其它RON节点转发IP报文。

David Andersen对分布在Internet上的RON进行两组对比测量,实验结果证明了RON体系结构的优点。实验在Internet上实现了有12个节点和132条路径的RON,测试进行了64小时,总共经历了32次严重的路径失效。测试结果表明RON路由机制可以检测失效、发现路径和恢复路由,整个过程所用的平均时间不到20s。

SCTP(Stream Control Transmission Protocol)^[6]提供了基于主机的多宿主机制。SCTP与传统传输协议(TCP,UDP等)的最大不同在于提供了终端主机的多宿主和多流功能。为了支持多宿主,SCTP为每个端节点提供了与之通信的对端的一组传输地址(如:一个SCTP端口与多个IP地址的组合)。SCTP引入了联合(Association)的概念,两台主机之间进行通信需要首先建立联合,联合可以使用每台主机上的多个接口进行协作。多宿主主机包含多个网络接口,同时连接多条路径,多条路径可以合并到一个联合中。SCTP使用内嵌的Heartbeat机制来监测联合中的所有路径。当检测到一条路径失效时,协议就会通过另外一条路径来发送数据,并对

上层应用保持透明。

SCTP的缺陷在于需要修改主机协议栈以适应传输层协议的变化,联合基于一组静态地址,不适合动态的移动场景,不支持流量工程和多播。

SHIM6^[7]提供了基于主机的多宿主机制。SHIM6没有引入新的地址空间,Locator和Identifier均使用IPv6地址空间。SHIM6为多宿主的端主机提供了多个IPv6地址构成Locator集合,并从中挑出一个作为ULID(Up Layer ID)。Locator是端主机接口用于路由的IPv6地址,ULID是上层会话(TCP/UDP)使用的介于网络层和传输层之间,在传输连接中使用的固定标识。上层应用使用原有的IPv6地址作为ULID,并由主机的SHIM6层根据链路和负载情况从Locator集合中确定特定的Locator,以选择提供者 and 路径。由于ULID使用已有地址空间且保持固定,不需要改变上层Socket API,当源或目的位置改变时,也不会影响上层通信。以此为基础,SHIM6提供了链路故障检测及恢复机制^[8]。SHIM6转发平面使用REAP协议探测链路故障,重新选择源和目的Locator以及ISP来规避故障链路。

SHIM6采用了基于主机的方式,可以根据链路状态动态选择Locator,并实现了对多播的支持。SHIM6的缺点在于机制过于复杂,需要更新大量的主机,仅支持IPv6。

LISA-Recovery借鉴了SCTP和SHIM6的“Locator/Identifier Split”思想,不同之处在于提供基于网络的站点多宿主故障恢复机制。我们还借鉴了RON的路径失效和周期性性能检测机制,对站点的多条路径质量进行监测。

3 LISA 简介

LISA采用了基于网络的“Locator/Identifier Split”命名和寻址体系结构,如图1所示。将网络划分为核心网络和边缘网络,核心网络使用Locator名字空间,边缘网络使用Identifier名字空间。主机可以继续使用原有的地址空间(如IPv4/v6地址),但仅仅作为Identifier存在,不再具有全局路由能力。主机上层会话基于固定的Identifier,Identifier所对应的Locator可以发生变化。LISA保证了“Locator/ID Split”对端主机的透明,端主机协议栈及上层应用就可以继续使用原有的地址格式,从而避免了大规模主机更新的困境。

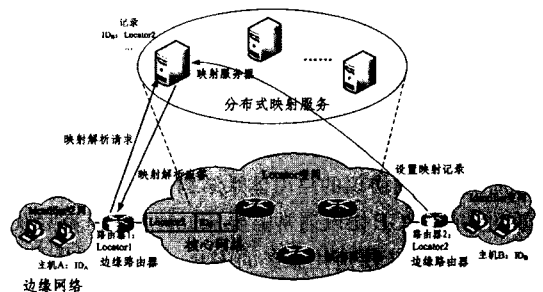


图1 基于网络的位置与标识分离的命名和寻址体系结构(LISA)

网络设备(边缘路由器)通过查询分布式映射服务系统,完成Identifier空间和Locator空间之间的转换。当主机移动(Locator变化)时,边缘路由器还可以快速更新分布式映射服务系统中对应的映射记录。

LISA采用了“映射+封装”的实现方式。边缘路由器收到主机发送的报文后,根据报文包含的Identifier信息,从分

布式映射服务系统查询获得匹配的 Locator 记录。随后将一个包含 Locator 的新报文头附加在当前的报文上,通过 在路由器之间建立的 GRE(Generic Routing Encapsulation)^[9] 隧道进行转发。在新报文中,内部报头的源和目的地址是 Identifier(IP 地址),外部报头的源和目的地址是 Locator。当一个经过封装后的报文到达目的地的边缘路由器时,该路由器将解封该报文,并根据 Identifier 将其发送到目的地。

4 LISA-Recovery

LISA-Recovery 以站点多宿主为基础,提供了路径状态和性能监控功能。一方面,上层通信会话可以在多个传输路径间选择和切换,提高了网络的容错的能力;另一方面, LISA-Recovery 在路径的选择上还综合考虑了 2 个与上层应用性能息息相关的 路径质量指标:延迟和报文丢失率,从而可以根据上层应用的需要灵活地选择最佳传输路径。

4.1 总体结构

LISA 为实现站点多宿主提供了基础。如图 2 所示,当站点边缘路由器连接多个 ISP 时,就获得了多个 Locator。边缘路由器可以在分布式映射服务系统中为本站点内主机创建多宿主映射记录,即一个站点主机 Identifier 对应多个核心网络 Locator。在多宿主映射记录中,一个 Locator 作为首选 Locator,其余的作为备选。这样站点就拥有了跨越不同 ISP 的多条转发路径。由于 Locator 为不同 ISP 分配,不同 Locator 对应的不同路径跨越不同的 ISP,存在较少的路径重叠,从而保证了多条路径的异质性,为路径切换提供了基础。

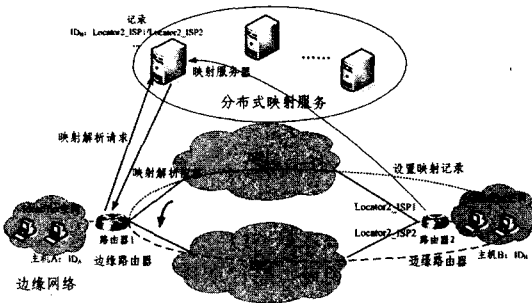


图 2 站点多宿主的路径失效恢复(LISA-Recovery)

LISA-Recovery 利用多宿主站点的多条转发路径来实现路径失效的快速恢复。当边缘路由器通过查询映射记录,发现目的主机的 Identifier 对应多个 Locator,就会在使用首选 Locator 转发报文的同时,对每个 Locator 对应的转发路径的状态和质量进行监控。边缘路由器 1 通过查询主机 B 的 Identifier: ID_B,得知主机 B 位于多宿主站点,将主机 B 对应的 2 个 Locator(Locator2_ISP1, Locator2_ISP2)加入到探测成员列表,并开始发送探测消息对分别跨越 ISP1 和 ISP2 的这两条路径进行监控。

当默认的首选 Locator 所对应的路径出现通信故障时, LISA-Recovery 会将当前转发路径自动切换到备用的 Locator 所对应的转发路径上。假定 Locator2_ISP1 是目的主机 B 的首选 Locator。正常情况下,主机 A 和 B 之间的报文通过 ISP1 转发。若边缘路由器 1 通过探测报文发现经由 ISP1 的转发路径出现故障或性能下降,而经由 ISP2 的转发路径保持正常,就使用 Locator2_ISP2 作为主机 B 的 Locator 来重新封装报文,所有报文通过跨越 ISP2 的路径进行转发。

4.2 LISA 边缘路由器

4.2.1 基本结构和探测机制

在 LISA 体系结构中,核心路由器的组成和功能与目前的网络核心路由器相同,但边缘路由器则有了很大变化。LISA 边缘路由器是 LISA-Recovery 的核心,它通过发送探测报文监控它和多宿主站点之间的路径质量,进行路径选择,并完成报文转发。

如图 3 所示, LISA 边缘路由器除了提供基本的报文转发功能外,还增加了映射服务查询/设置、探测和探测对象管理等模块。其中,映射服务查询/设置模块用于同分布式映射服务系统交互,以获得或更新映射记录;探测模块用于监控目的端多宿主站点的多条路径的质量;探测对象管理模块用于维护探测对象(Locator)成员列表,保存多宿主站点多条路径的详细性能信息(延迟、报文丢失率等);转发模块根据查询得到的映射信息和探测得到的路径质量信息,决定报文的转发路径。

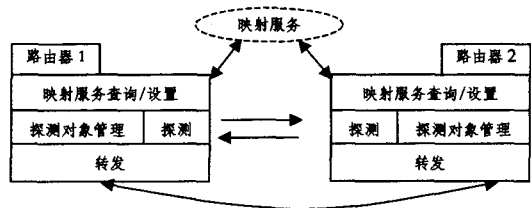


图 3 LISA 边缘路由器系统组成

为了控制探测成员的规模,减少边缘路由器的探测开销, LISA-Recovery 采用数据驱动(Data Driven)的方式来启动探测机制,只对活跃的多宿主站点的 Locator 进行探测。若 LISA 边缘路由器在转发报文时,通过查询 Identifier 得知目的端为一多宿主站点,即把该多宿主站点的多个 Locator 加入探测列表中。每个被探测的 Locator 都有生存期。若某个被探测的 Locator 在生存期满后,没有发生针对该 Locator 所对应的 Identifier 的查询,则将该 Locator 从探测成员列表中清除,停止对该 Locator 所对应路径的探测监控。

4.2.2 探测机制优化

随着多宿主的日益普及,通信双发都需要探测路径状态和质量,并进行路径选择。LISA-Recovery 通过测量探测报文的 RTT 时间来判断路径状态和性能,若通信双发均为多宿主站点(拥有多个 Locator),则双方都需要发送探测报文来测量 RTT 时间,为此引入了 RTT 协作探测机制。

RTT 协作探测过程如图 4 所示。LISA 路由器对多宿主站点的某个 Locator 进行探测,初始探测报文携带了探测报文 ID。对端收到初始探测报文后,随即发送响应报文(包含初始探测报文的 ID_i 和响应报文的 ID_r)。LISA 路由器收到响应报文,通过对比探测报文的 ID,并根据事先记录的发送时刻,即可算出探测报文的 RTT(8 秒)。LISA 路由器也会将收到的响应报文看作探测报文,向多宿主站点发送响应报文,这样多宿主站点也可以计算出 RTT(9 秒)。具有 LISA-Recovery 功能的两端就可以在 3 个报文之内分别计算出 RTT,这种探测方式非常适合通信双方均为具有 LISA-Recovery 功能的多宿主站点的情况。一方面可以减少重复的探测报文,若探测双方均独立探测,则探测到一次 RTT 至少需要发送 4 个报文;另一方面可以避免探测双方进行时间同步,探测双方

均根据本地时间计算 RTT。

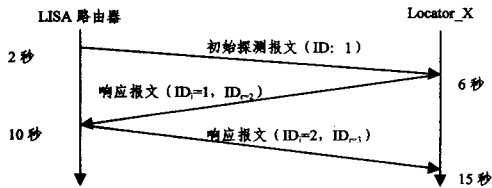


图 4 RTT 主动协同探测机制

为了进一步减少重复探测,需要对 RTT 协作探测的不同探测性质进行分析,为此,引入如下定义:

定义 1(主动探测) 依据设定的探测频率对探测列表中的 Locator 主动发起的探测。

定义 2(被动探测) 被动响应其它 LISA 路由器的主动探测而发起的探测。

定义 3(初始探测报文) LISA 路由器主动探测时发出的第一个报文,仅携带自身 ID。

定义 4(响应报文) 对初始探测报文和其他响应报文做出响应的报文,不但携带自身 ID,还携带被响应报文的 ID。

假定 LISA 路由器 A, B 为多宿主站点的边缘路由器,2 个站点之间存在通信流量, A, B 相互进行 LISA-Recovery 探测。 A, B 进行主动探测时间间隔分别为: $\Delta T_A, \Delta T_B$,则在一段时间 T ,则 A, B 更新 RTT 的次数为: $count = \frac{T}{\Delta T_A} + \frac{T}{\Delta T_B}$ 。可见实际的更新次数远远超过了 A, B 各自需要的次数。

为了减少探测机制对 LISA 路由器资源和链路带宽的消耗,应当减少不必要的主动探测。为此,每个 LISA 路由器需要维护两个 Locator 探测列表:主动探测列表、被动探测列表。当 LISA 路由器收到对端的主动探测后,就把对端的 Locator 加入到被动探测列表中。在进行主动探测之前,先要查询被动探测列表,若探测的 Locator 在其中,则不进行主动探测,也不把该 Locator 加入到主动探测列表中。这样就使得在一对多宿主站点 LISA 路由器之间,只保持一次的主动/被动探测,从而减少了资源开销。改进后的 LISA-Recovery 探测算法伪码描述如图 5 所示。

```

Algorithm LISA-Recovery_Response(packet)
IF packet.type='initial probe packet' Then
    Add to Passive_Probe_List(packet, locator)
    Send response packet(packet, ID, ID_r)
    Record send time(ID_r)
ELSE
    Record receive time(packet, ID)
    Send response packet(packet, ID, ID_r)
    Record send time(ID_r)
ENDIF
Algorithm LISA-Recovery_Probe()
FOR num=1 to Active_Probe_List.count
IF Active_Probe_List[num] is in the Passive_Probe_List THEN
    Delete form Active_Probe_List[num]
ELSE
    Send probe packet(ID_s)
    Record send time(ID_s)
END IF

```

END FOR

图 5 LISA-Recovery 探测算法

4.3 路径选择

LISA-Recovery 提供了快速的路径失效检测和恢复,以及满足特定应用对路径性能要求的路径选择机制。LISA 路由器探测并维护目的地多宿主站点的多条路径状态信息,判断首选路径是否活跃,同时评估潜在路径的质量,根据路径状态和应用需求选择最适合的路径。LISA-Recovery 默认维护两种路径质量指标:延迟、报文丢失率。上层应用也可以根据需要自定义路径质量指标参数。

LISA-Recovery 定义了用于探测转发路径状态和质量的探测报文。探测报文基于 UDP,根据预先设定的探测频率,探测报文沿着每条路径发送到多宿主对端。对端收到后立即返回相应的确认消息。这种机制被用来精确测量多宿主路径的 RTT,实时监控路径的可用性和丢包情况。探测报文的发送频率可以根据用户的需求灵活改变。

源端 LISA 边缘路由器通过某条路径(对应不同的 Locator),向目的端多宿主站点的 LISA 边缘路由器发送探测报文后,发送端会使用定时器等待对方的接收确认,如果收到对端的确认表示探测报文已经完成传输,如果是发生定时器超时,则意味着消息丢失或者发生了较大的拥塞延迟。当某条路径上连续发生了多次传递消息超时,达到了设定的 Path_Max_Retrans 次数,则这条路径被标记为不可用,所有发往目的端多宿主站点的报文被切换到备选的路径上传输。

LISA-Recovery 通过测量探测报文的 RTT 来确定某条路径上的报文延迟情况,每次探测得到一个新的延迟数据,即根据式(1)计算一个当前的路径平均延迟。通过比较多条路径的平均延迟,并根据上层应用对延迟的要求,LISA 边缘路由器选择最佳的路径转发报文。

$$Latency_{average} = (1-\alpha)Latency_{average} + \alpha(Latency_{new}) \quad (1)$$

LISA-Recovery 通过对特定时间间隔,经由某条路径发送和丢弃的探测报文进行计数,计算该路径的丢包率,并根据上层应用对丢包率的限制进行路径选择。

5 性能评估

首先通过 NS-2 模拟实验,检验当出现路径失效时,LISA-Recovery 的快速检测和恢复能力。其次,检验出现路径性能下降(丢包率、延迟增大),LISA-Recovery 重新进行路径选择,并改善上层应用延迟、丢包率的能力。最后,还通过理论计算,分析了网络额外带宽开销的情况。

5.1 模拟场景设计

模拟实验采用了 NS-2(2.33 版本)。我们借鉴 SCTP patch 模拟主机多宿主场景的思想,设计了 LISA 站点多宿主场景。在 NS-2 中,由于单个节点(node)不具备多宿主特性,需要构建一个基于多个实物节点的虚拟多宿主节点。在 LISA 站点多宿主场景中,站点对外拥有 2 条传输路径,边缘路由器即为虚拟多宿主节点。如图 6 所示,虚拟多宿主节点由 3 个节点构成:1 个 master 节点和 2 个 slave 节点。每个 slave 节点连接 1 条路径,并传输实际流量。master 节点决定哪个 slave 节点用于转发报文,master 节点与 slave 节点之间通过单向链路连接,该链路不传输实际流量,仅用于 master 节点对 slave 节点的动态控制。LISA-Recovery Agent 用于实现

LISA-Recovery 的功能, LISA-Recovery Agent 被连接在每个 LISA 节点上。

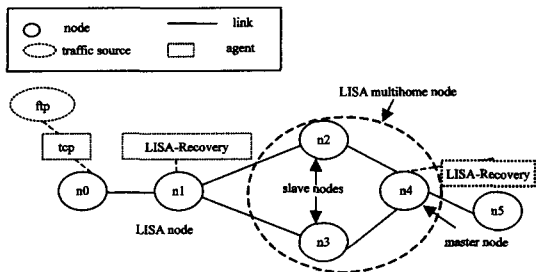


图 6 LISA 站点多宿主场景

5.2 机制验证

5.2.1 路径失效检测

我们设定 n1-n2 是报文转发主路径, n1-n3 是备用路径。实验模拟了 FTP 服务在具有 LISA-Recovery 机制的多宿主站点上运行的情况。在实验过程中, 通过切断 n1-n2 的链路来模拟主路径失效的情况。

实验模拟了在关闭和开启 LISA-Recovery 两种情况下, 出现路径失效时, FTP 服务吞吐率的变化情况, 如图 7 所示。实验开始后不久, n1-n2 中断; 经过一段时间的探测, LISA-Recovery 检测到 n1-n2 路径失效, 选择 n1-n3 进行报文转发。在没有 LISA-Recovery 机制的条件下, 当主路径失效后, FTP 在整个实验过程中完全失效; 在开启 LISA-Recovery 的情况下, FTP 服务经过短暂的中断后, 逐渐恢复正常。由此可见, LISA-Recovery 可以很好地检测路径失效, 并快速恢复报文传输, 从而提高了上层应用服务的可靠性。

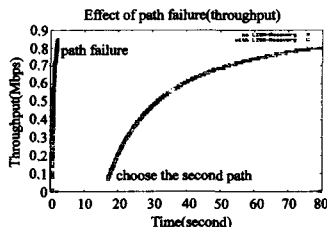


图 7 路径失效影响(吞吐量)

5.2.2 路径性能检测

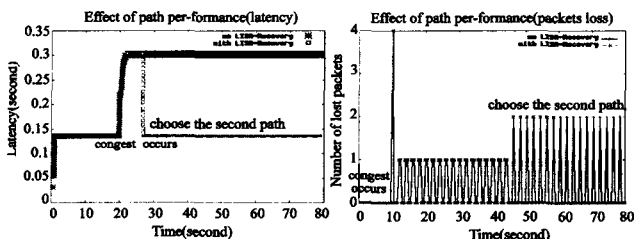


图 8 网络拥塞影响(延迟)

图 9 网络拥塞影响(丢包)

实验还模拟了网络拥塞造成主路径性能下降(延迟增大, 丢包)时, LISA-Recovery 的检测和恢复机制, 如图 8 和图 9 所示。在 n1-n2 路径加入稳定的额外流量(CBR), 使得 n1-n2 传输的实际流量超过其带宽, 从而使得已有的 FTP 服务出现丢包, 报文延迟加大。额外流量进入 n1-n2 后, n1-n2 出现了拥塞, 导致 FTP 报文的延迟迅速增大; 经过一段时间的探测, LISA-Recovery 检测到 n1-n2 的报文延迟情况超出了预先设定的阈值, 于是选择 n1-n3 进行报文转发。在没有 LISA-Recovery 机制的条件下, 当主路径出现拥塞后, FTP 在整个实

验过程中都处于高延迟的状态; 在开启 LISA-Recovery 的情况下, FTP 服务经过短暂的高延迟后, 迅速恢复到正常延迟。

5.3 探测性能及开销分析

LISA-Recovery 通过发送探测报文来探测路径的状态和质量, 探测报文的探测频率直接影响路径失效和性能检测的效率和开销。设定每个 LISA 路由器的探测报文探测间隔时间为 Probe_Interval 秒。为了防止多个 LISA 路由器探测出现自同步, 导致探测报文风暴, LISA 路由器在固定的探测间隔时间基础上增加一个随机变化值: 0 到 $1/4 \text{Probe_Interval}$ 秒。因此, 可以得出 2 次探测之间的平均间隔时间为 $9/8 \text{Probe_Interval}$ 秒。

LISA-Recovery 设定了探测报文的超时时间: Probe_Timeout, 若一次探测报文在 Probe_Timeout 之内没有回应, 即认为报文丢失。当检测一个探测报文丢失, 将立即发送下一个探测报文, 最多发送 2 次。如果这 3 次探测报文均丢失, 即认为该路径失效。因此, 路径失效的最短检测时间为 $3 \times \text{Probe_Timeout}$, 最长检测时间为:

$$\text{Detect_time}_{\max} = \text{Probe_Interval} + \frac{1}{4} \text{Probe_Interval} + 3 \times \text{Probe_Timeout} \quad (2)$$

平均检测时间为:

$$\text{Detect_time}_{\text{average}} = \frac{1}{2} \times \frac{9}{8} \text{Probe_Interval} + 3 \times \text{Probe_Timeout} \quad (3)$$

在系统的实现中, 设定 Probe_Interval: 5 秒, Probe_Timeout: 3 秒, 则带入式(2)得到对路径失效的平均检测时间为 11.8125 秒。路径性能的检测除了与探测机制相关, 还与应用服务的性能要求相关。在不同的性能要求下, 检测的时效性也存在不同。设定探测报文的大小为 Packet_Size, 监测路径数目为 n, 则消耗的额外带宽为:

$$\text{Band_Width} = \frac{2 \times \text{Packet_Size}}{\frac{9}{8} \text{Probe_Interval}} n \quad (4)$$

在系统的实现中, 设定 Packet_Size: 64 字节, Probe_Interval: 5 秒, n: 2, 则带入式(4)得到额外的带宽消耗为 0.364 kbps, 仅占模拟实验设定链路带宽(1M)的 0.036%。由此可见, 探测报文对链路带宽的消耗是很小的。

结束语 多宿主是应对路径失效, 提高站点可靠性的有效途径之一。然而囿于当前 TCP/IP 体系结构的限制, 多宿主还没有很好地普及。位置与标识分离的思想天然支持多宿主。在 LISA 命名与寻址体系结构下, 我们提出了基于位置与标识分离的站点多宿主方案, 并以此为基础提出了路径失效恢复机制: LISA-Recovery。模拟实验验证了 LISA-Recovery 能够有效地检测路径失效和性能下降, 并进行快速的路径切换, 从而保障了上层应用服务的正常运行。通过理论计算, 分析了 LISA-Recovery 所带来的开销很小, 具有较好的实用性。下一步, 将进行小范围的部署运行, 以检验其在真实环境下的性能。

参考文献

- [1] Labovitz C A A, Bose A, Jahanian F. Delayed Internet Routing Convergence[C]// SIGCOMM'00. Stockholm, Sweden, 2000: 175-187

[2] Tu Rui, Su Jinshu, Meng Zhaowei, et al. UCEN: User Centric Enterprise Network[C]// IEEE ICACT 2008. Phoenix Park, Korea, 2008; 66-71

[3] Meyer D. The Locator Identity Separation Protocol(LISP)[J]. The Internet Protocol Journal, 2008, 11(1): 23-36

[4] Meyer D, Fall K. Report from the IAB Workshop on Routing and Addressing. Internet Draft, 2006

[5] Andersen D, Kaashoek F, Morris R. Resilient Overlay Networks [C] // 18th ACM Symp. on Operating Systems Principles (SOSP). Banff, Canada, 2001; 131-145

[6] Ong L J Y. An Introduction to the Stream Control Transmission Protocol(SCTP). RFC 3286, 2002

[7] Nordmark E M B. Shim6: Level 3 Multihoming Shim Protocol for IPv6. Internet Draft, 2008

[8] S 'ebastien Barr 'e O B. Improved Path Exploration in shim 6 - based Multihoming[C]//SIGCOMM 2007 Workshop "IPv6 and the Future of the Internet". 2007; 61-68

[9] Farinacci D, Hanks T L S, Mayer D, et al. Generic Routing Encapsulation(GRE). RFC 2784, 2000

(上接第 48 页)

意义与(a)相同,星号点线表示用锚节点自身提高定位精度的方法的结果。

但是实际中理想的信号传播环境是不存在的,信号传播会受到外界环境的干扰。干扰有时可能很强,在通信半径内的节点可能会收不到发射节点发出的信号。这时就会出现图 5 的情况,图中在不规则的实线圆环内的节点会根据收到的锚节点的广播信号确定自己的相对位置在虚线的规则圆环内。

这种情况对定位精度影响较大,特别是当锚节点确定的圆环宽度较小、圆环较多时,对定位精度影响会很大。这时,用锚节点自身进行精确化的方法会得到更大定位误差。而我们的方法有很好的稳定性,可以有效地避免传播环境的干扰对定位的影响。在 4.1.1 节中的仿真环境下,增加信号传播的干扰因素,再次进行仿真,得到的结果很好地证明了这一点。仿真结果如图 6 所示。

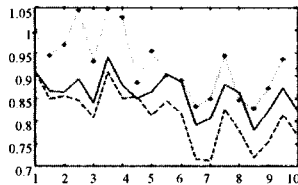
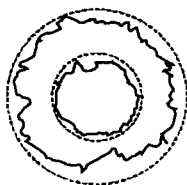


图 5 有干扰条件下的同心圆模型 图 6 有干扰环境下的仿真结果

4.2 能量分析

传感器节点以半径为 r 广播消耗的能量为 $E = cr^k$, $2 \leq k \leq 4$ 。其中 c 为系数,与广播数据量等参数有关。因此,若锚节点的初始发射半径为 R ,则对于某一锚节点精确化次数为 n 的未知节点在精确化过程中消耗的总能量为

$$E = \sum_{j=1}^n c \cdot \left(\frac{R}{2^j}\right)^k \leq c \cdot \left(\sum_{j=1}^n \frac{R}{2^j}\right)^k \leq c \cdot \left(\sum_{j=1}^{\infty} \frac{R}{2^j}\right)^k = c \cdot R^k$$

设每个锚节点初始时确定 3 个宽度为 R 的圆环,则每个锚节点确定这 3 个圆环消耗的能量为 $E = c \cdot R^k + c \cdot (2R)^k + c \cdot (3R)^k \geq 14cR^k$ 。如果由锚节点自身进行精确化,将每个圆环的宽度变为原来的 $1/3$,确定 9 个宽度为 $\frac{R}{3}$ 的圆环,则每个

锚节点确定这 9 个圆环消耗的能量为 $E' = \sum_{j=1}^9 c \cdot \left(j \cdot \frac{R}{3}\right)^k \geq 30cR^k$,比初始消耗的能量多了 $16cR^k$ 。如果确定的圆环的宽度变得更小,如圆环的宽度变为原来的 $1/6$,则要消耗更多的能量。这样锚节点在定位结束后就耗尽能量而失效。如果锚节点在网络中所占比例比较大的话,则锚节点的失效会影响

整个网络的性能。

结束语 节点位置区域估计方法是一类重要的无线传感器网络定位非测距定位方法。同心圆定位方法是新近提出的具有代表性的一种区域定位方法。本文提出一种改进的同心圆定位方法,在原方法的基础上,利用相邻圆环内的未知节点间的通信减小包含未知节点的圆环的宽度,从而更精确地估计出了未知节点的位置。该方法只比原方法增加很少的能耗,同时在理想环境和有干扰环境下其定位精度都明显高于原方法。

参 考 文 献

[1] Vivekanandan V, Wong V W S. Concentric anchor beacon localization algorithm for wireless sensor networks[J]. IEEE Transactions on Vehicular Technology, 2007, 56(5): 2733-2744

[2] Bulusu B, Heidemann J, Estrin D. Density adaptive algorithm for beacon placement in wireless sensor networks[C]// IEEE ICDCS'01. Phoenix, AZ, April 2001

[3] He T, Huang C, Blum B M, et al. Range-free localization scheme for large scale sensor networks[C]// Proc. 9th Annual Int'l Conf. on Mobile Computing and Networking. San Diego, CA, 2003; 81-95

[4] Savvides A, Han C-C, Srivastava M B. Dynamic fine-grained localization in adhoc networks of sensors[C]// Proceedings of the 7th Annual Conference on Mobile Computing and Networking. Rome, Italy, July 2001; 166-179

[5] Nagpal R. Organizing a global coordinate system from local information on an amorphous computer[R]. AI Memo 1666. MIT AI Laboratory, August 1999

[6] Yang Sungwon, Yi Jiyoung, Cha Hojung. HCRL: A Hop-Count-Ratio based Localization in Wireless Sensor Networks[J]. Sensor, Mesh and Ad Hoc Communications and Networks, 2007, 18(21): 31-40

[7] Wong Sau Yee, Lim Joo Ghee, Rao S V, et al. Density-aware Hop-Count Localization (DHL) in Wireless Sensor Networks with Variable Density[C]// Wireless Communications and Networking Conference. Volume 3. March 2005; 1848-1853

[8] Stojmenovic I. Handbook of Sensor Networks—Algorithms and Architectures[M]. a john wiley & sons, inc. , publication

[9] 孙利民,李建中,陈渝,等. 无线传感器网络[M]. 北京:清华大学出版社

[10] 陈文萍. 无线传感器网络节点的协同定位策略[J]. 计算机科学, 2007, 34(9): 97-101