

# 基于混合 P2P 的端到端网络测量系统<sup>①</sup>

赵文俊, 黎文伟

(湖南大学 信息科学与工程学院, 长沙 410082)

**摘要:** 基于 P2P 的网络测量系统以降低单点瓶颈和部署开销为目标, 其中结构化 P2P 一般假设节点具有相同的能力, 不适合大规模部署; 非结构化 P2P 通信负载过大, 可扩展性差. 因此, 本文结合多种 P2P 模式的优点, 引入分层思想, 对节点按能力分簇, 选取性能佳的作为簇首组成上层高速转发网, 由簇首负责维护和管理簇内普通节点, 同时描述了系统的初始化构建过程、各个功能模块、自适应维护机制等, 并实现了一个系统原型. 实验结果表明, 该系统实现了节点的自动化部署, 同时提高了系统的可扩展性, 降低了测量开销.

**关键词:** 网络测量; P2P; DHT; 心跳检测; 簇首

## Network Measurement System Based on Hybrid P2P

ZHAO Wen-Jun, LI Wen-Wei

(Department of Information Science and Engineering, Hunan University, Changsha 410082, China)

**Abstract:** The goal of network measurement system based on P2P is to reduce the single-point bottleneck and deployment costs. Generally, assumed to have the same capacity among the nodes, the structured P2P is unsuitable for large-scale deployment. The unstructured P2P communication is overload and its expandability is poor. Therefore, combing with the advantages of various P2P modes, the thesis brings in layered thought, clustering of the nodes in accordance with ability, which is the heads of a cluster selected from the excellent ones form the upper high-speed transmitting net, and take control of the maintenance and management of the common nodes in the clusters. The thesis also describes the initialized establishment process of the system, function modules and self-adaption maintenance mechanism and so on. Ultimately, system prototype is taken shape. The experiment suggests that not only is the automatic deployment of the node realized, but also the expandability of the system is improved, which reduces the measurement costs.

**Key words:** network measurement; P2P; DHT; heartbeat detection; regional center

伴随互联网的普及与发展, 以网络互连为依托的新型应用应运而生, 诸如视频、文件分享与传输、协同计算<sup>[1]</sup>等多种应用对端到端的网络性能提出了更高的要求<sup>[2]</sup>. 有效的测量是了解和认识网络的基本方法<sup>[3]</sup>, 可以获取复杂网络与业务的运行状态, 了解网络与业务行为特征并建立理论模型, 是提高网络服务的关键和基础<sup>[4]</sup>. 端到端的网络测量通常以主动测量的方式, 可以对节点间的网络性能有全方面的了解, 能突破互连网的异构性以及不同网络服务提供商(ISP)对

数据资源的访问权限.

P2P 是一种完全去中心化的自组织结构, 每一个节点完全对等, 在主动提供服务的同时, 也是服务的受众者. P2P 最早用于音乐分享, 相比传统的 C/S 架构, 它不依赖唯一的全局服务器, 所以基于 P2P 模式的应用可以在维持高并发的同时, 大大降低单点瓶颈从而提高系统的稳定性与可用性. P2P 的拓扑结构先后经历过集中式、分布式半结构化、分布式结构化, 并被广泛用于即时通讯、文件分享与下载、在线视频点播、

<sup>①</sup> 基金项目: 国家自然科学基金(61173168); 中央高校基本科研业务费项目

收稿时间: 2016-03-06; 收到修改稿时间: 2016-04-27 [doi:10.15888/j.cnki.csa.005481]

分布式计算,如 eMule、迅雷、PPlive 等。

近年来对网络测量系统已经有了很多方面的研究。早期的 Surveyor<sup>[5]</sup>、GIMS<sup>[6]</sup>等采用了分布式测量、集中式控制的组织形式,即测量站分布在网络中执行测量任务,由一个集中的控制中心与用户交互,同时控制中心对测量节点进行管理,负责任务分发与回收结果数据。但集中式控制本质还是 C/S 架构,可扩展性差,部署开销过大且存在单点瓶颈,不适用于大规模测量。在如何吸收大量普通端主机进行大规模协作计算上,对数据量庞大的太空无线电信号进行分析的 SETI@home 项目<sup>[7]</sup>是一个成功的范例,受该项目的启示,研究者开始研究吸收普通端主机为测量站的测量结构组织形式。DipZoom<sup>[8]</sup>采用 P2P 形式吸纳端主机,但仍需要集中服务器供用户与测量提供者进行协商;pMeasure 项目提出了基于 Pastry<sup>[9]</sup>的结构化 P2P 式测量结构。

但是,非结构化 P2P 模式通讯开销过大,查找与定位效率低,而结构化 P2P 模式一般都假定节点具有相同的能力,仅适用于规模较小的系统,且基于分布式哈希表(Distributed Hash Table)的自适应算法比较复杂;其次,目前基于 P2P 模式的研究均采用无层次结构,端主机频繁的加入、退出造成的波动容易影响系统的稳定性。此外,由于假定节点具有相同的能力,没有有效利用和平衡节点之间的差异性,所以难以进行测量任务的合理调度,进而影响了系统的可扩展性。

针对上述问题,本文引入分层思想,在考虑节点间差异性的前提下,充分结合几种 P2P 模式的优点,实现测量节点自动化部署,在保证高效查询、简单易用的同时,提高系统的可扩展性,并完成了一个基于混合 P2P 的网络测量系统原型。

## 1 P2P模式概述

P2P 是一种完全去中心化的自组织结构,节点之间完全对等,在提供服务的同时,也是服务的受众者。P2P 最早用于音乐分享,它的组织形式先后经历过集中式、半结构化、结构化。

最早的集中式拓扑使用一台中心服务器来共享资源,为查询信息的节点提供资源匹配服务,简单易用,查找效率高,但是基于中心服务器的集中式架构与传统的 C/S 架构类似,存在单点故障。

分布式半结构化 P2P 取消了中心服务器,采用泛

洪(Flooding)式搜索算法,每个节点生成一个查询后,将该请求发送给所有的邻居节点,如果邻居节点中包含所查询的内容,则查询完成,否则需要在自己的邻居节点之间继续转发这个请求,直到完成查询或者查询结束。该结构能够去掉任何节点,因此能够适应网络的动态变化,具有较好的容错能力,但是该算法效率低下,且通讯开销大。

结构化 P2P 基于 DHT,通过哈希算法将存储对象映射到一个连续的散列表空间,散列表空间由多个块组成,每个散列表块由一个节点管理。存储对象通过某种哈希函数(如 SHA-1)得到一个散列值,散列值以及存储对象的主机信息被存储在管理该散列值所在散列表块的节点上,相比半结构化 P2P 的泛洪查询算法具有较好的查询效率,且没有过高的通信开销<sup>[10]</sup>。然而 P2P 网络是一种动态且随时变化的网络,任何时候都有新节点加入网络,同时也有很多节点离开网络,动态维护开销大,且未考虑到节点间的性能差异,不适用大规模网络。

## 2 系统架构设计

端到端的网络测量需要一个有效的节点自动化部署和管理方案,为了完成该目标,我们必须解决一些关键问题。首先,我们设计的程序应该是独立于系统平台的,可以采用一门支持跨平台的语言来编写系统,比如 Python、Java;其次,借鉴 P2P 的思想,一个节点在请求服务的同时也为其它节点提供服务,但由于潜在节点个数庞大,我们需要一个有效的机制来管理各个节点的信息,包括引导节点初始化、实时管理/统计/维护节点的常用信息。此外,我们需要一个自适应维护机制,尽可能地减小节点频繁加入/退出对整个系统的影响。最后,在实现功能的基础上,要考虑节点之间性能、稳定性的差异,尽可能地分担单一节点的负荷从而降低单点瓶颈发生的概率,同时采取一些优化措施来提高数据存储、数据查询以及任务调度的性能。

在充分比较各种 P2P 模式的优缺点后,我们结合集中式、结构化 P2P、非结构化 P2P 的优点,设计了一种基于混合 P2P 模式的架构,引入分层思想对节点实现分簇,考虑到节点性能之间的差异,我们选择较优的节点作为簇首,其它则为普通节点,由簇首管理本簇的普通节点,簇首之间为分布式结构化,簇内为非结构化。

基于该设计, 我们实现了一个端到端网络测量系统的原型, 它分为上下两层结构, 共由三个部分组成: 注册服务器(RS, Register Server)、簇首节点(RC, Regional Center)、普通节点(EP, End Point), 见图 1.

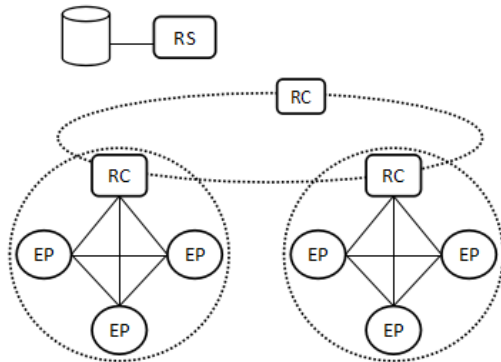


图 1 系统结构

注册服务器(RS)负责引导节点的首次加入与注册、管理簇首信息、收集测量结果、数据处理与分析等功能。由于注册服务器只负责引导节点的初次加入, 并收集测量结果对外提供可视化查询, 所以不会成为系统的瓶颈。

簇首(RC)需要维护本簇的普通节点(EP)和部分其它簇首(RC), 既要响应上层 DHT 网络中其它簇首节点的查询请求和簇内的任务调度, 还要负责动态维护和选择备选簇首。簇内的测量任务的结果由簇首合并压缩后, 定期发送给注册服务器。

普通节点(EP)在加入系统后, 可以向所属簇首发起任务请求, 同时接收来自簇首的任务调度。此外, 通过发送心跳包辅助簇首动态维护本簇的节点信息。

### 2.1 系统构建方法

我们采用基于 IP 和地理位置的分簇方法对节点进行分簇, 尽量减小跨域造成的延时, 考虑到节点间性能的差异, 我们从中挑选性能较好的节点作为簇首, 其它节点各自加入对应的簇。于是我们形成了上层为各个簇首组成的 DHT 网络, 簇首之间为结构化网络, 基于 Kademia 协议修改实现, 既保证消息的高速转发, 也提高了查询速度。下层为簇首节点维护的簇内结构, 簇内为非对称结构, 各个簇内实现区域自治, 由簇首负责管理本簇的节点信息, 从而在保证查询效率的同时, 尽可能降低节点频繁加入/退出对整个系统的影响。此外, 我们使用心跳检测机制来动态维护节点的状态, 在保证功能的同时尽可能减轻簇首节点的负荷, 从而

降低单点瓶颈, 提高系统的可扩展性。

### 2.2 初始化构建过程

节点的状态分为 EP、RC, 如图 2 所示, 在系统的初始化阶段, 所有节点的状态都是 EP, 当一个新节点想要加入系统时, 它需要通过 RS 的引导完成注册认证, 选择切换为 RC 或者以 EP 加入对应的簇内结构, 详细过程如下:

① EP 向 RS 发起注册申请。

② RS 收到请求后, 判断该节点所属的 RC 是否存在——若不存在, 则将该节点的信息加入到所维护的 RC LIST 中, 并返回若干数量其它 RC, 反之则返回所属 RC 的信息。

③ 节点获得响应后, 解析 RS 的通知——若自己是 RC, 则切换状态并开启相应的功能, 反之则向所属的 RC 发起加入申请。

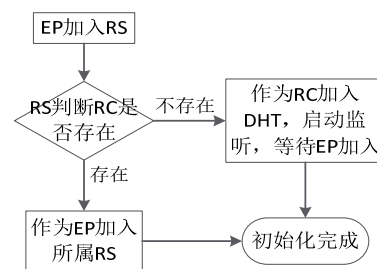


图 2 初始化流程图

以此类推, 随着加入节点的数量增加, RC 加入到上层 DHT 网络中, EP 加入下层对应的簇内结构, 完成平台的初始化构建过程。

### 2.3 任务调度

系统内分为上下两层结构, 上层是由各个簇首组成的 DHT 网络, 下层是各簇内部结构, 簇首维护部分其它簇首, 用于在上层网络中转发查找目标节点。此外, 簇首负责本簇的任务调度, 比如接受簇内某个节点发起的查询。所以对于一个查询任务, 往往分为两个阶段:

阶段 1: 查找簇首。

首先识别目标节点所属的簇, 在簇首列表中查询目标簇首是否存在, 若存在进入下一阶段, 否则在上层 DHT 网络中查找目标簇首。

阶段 2: 向簇首查找目标节点。

在获得目标簇首的信息后, 向它请求若干目标节点, 获取响应节点后, 返回给任务的发起者, 由发起节点与目标节点发起端到端的网络测量。详细过程如

图3所示.

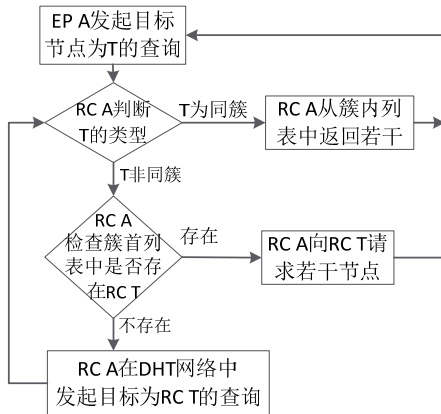


图3 查询流程图

① EP A 向所属 RC A 发起任务, 由 RC 对查询目标 EP T 进行分类:

若 EP T 属于本簇, 则从可用节点列表中随机返回一个, 反之则查询可用簇首列表: 如果 T 所属的 RC T 存在, 则向它发起节点请求, 反之, 则在上层 DHT 网络中, 向簇首列表中的其它簇首发起目标为 RC T 的簇首请求. 在获得 RC T 的响应后, 向 RC T 请求 EP T.

② RC A 获得 RC T 返回的 EP T 后, 返回结果给 EP A.

③ EP A 获得响应结果后, 向目标节点 EP A 发起端到端的网络测量.

此外, 为了提高高并发场景下任务调度的性能, 我们引入生产者/消费者模式, 使用基于消息队列的无阻塞异步方式来取代传统的同步模式, 既降低了内部模块间的耦合度, 也加快了响应速度.

### 2.4 自适应维护机制

在我们的两层结构中, 簇首节点需要维护本簇内所有普通节点的信息以便高效的任务调度, 簇内普通节点维护簇首信息. 由于节点可能频繁地加入/退出, 我们需要簇首节点能够动态维护簇内的可用节点信息, 同时动态维护一个备选簇首, 从而避免簇首异常退出对系统的影响.

#### 2.4.1 心跳检测机制

心跳在现实生活中通常用于判断一个人的生命体征, 于是诞生了心跳检测并被广泛应用于分布式系统中, 用于检测目标节点是否在线. 通过向目标节点发送一个指定格式的消息, 根据是否接收对应的响应来判断目标节点的状态.

在本系统中, 簇首需要动态维护本簇的可用节点列表, 如果采用主动式, 需要簇首主动发送心跳包给所需要维护的对象, 简单易用, 但随着簇内节点数量的增加, 双向往返群发心跳包的开销会线性增长, 从而加剧了簇首的负荷. 如果采用被动式, 由普通节点承担定时发送心跳包的功能, 而簇首只需要单向接收这些包也能实现预期功能. 相比主动式心跳机制, 伴随主动权的转移, 由双向发送接收改为单向收包后, 簇首在心跳功能上的网络开销降低了一半. 此外, 由于主动式必须向列表中的所有节点发包, 而被动式则只会接收到在线节点的报文, 这进一步降低了收包的数量, 从而减轻簇首的负荷, 降低了单点故障的概率.

#### 2.4.2 动态簇首维护机制

除了第一次由注册服务器任命, 在随后的时间里, 簇首的动态维护工作在各簇内进行落实, 从而减轻注册服务器的功能负荷.

由于簇首维护本簇内所有可用节点信息, 所以首先由簇首选取一个备选簇首, 然后由簇首和备选簇首通过心跳包来互相感知彼此的状态. 当备选簇首检测到簇首退出后, 首先向注册服务器发出簇首变更通知, 并从接收到的响应中提取必要信息完成簇首列表初始化, 然后向本簇 EP 发出变更通知, 节点获得通知后, 更新自己的簇首信息, 交替工作完成.

#### 2.4.3 可用节点维护机制

簇首需要动态维护本簇普通节点和若干其它簇首的状态. 通过被动式心跳检测机制, 簇首可以根据收到的普通节点发送的心跳包来更新对应的状态, 及时删除已下线节点. 对于其它簇首节点, 簇首使用注册服务器返回的一部分其它簇首信息完成对簇首列表的初始化后, 需要定时向其它簇首发送簇首查询请求, 从而实现簇首列表的动态更新, 包括获得新的簇首信息、更新被替换的簇首信息.

#### 2.5 数据持久化与数据查询

为了避免重启后造成内存中的数据丢失, 我们需要把任务请求、测量结果、有效节点信息等数据持久化存储. 相比文件操作, 使用 SQL 是一个好选择, 所以我们使用数据库来实现持久层. 此外, 考虑到我们的原型系统是一个多读少写的的应用, 所以在数据持久层之上加了一个数据访问层用来提供缓存数据, 既可以大大减小数据库的压力, 也能提高系统的查询速度, 如图4所示.

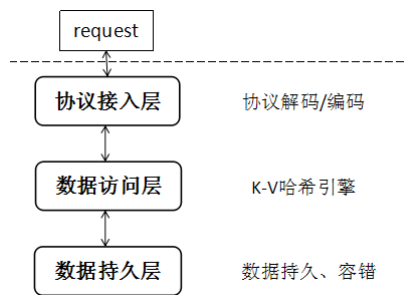


图 4 数据访问流程

对于读操作,我们首先在缓存中检索,如果命中则直接返回,反之则转向数据持久层,同时更新缓存中对应的数据。

对于写操作,我们先更新缓存中的数据,再将更新作用于数据库。

此外,我们使用日志组件来记录所有的用户行为,既保证数据的安全性,也提高了系统的可维护性。

### 3 系统实现

根据上一节描述的系统设计,使用 Python 2.7 实现了一个端到端网络测量系统原型,且支持跨平台,用户只需一键点击,即可自动完成初始化,并在注册服务器的引导下加入到系统,支持自定义测量任务,操作简单,易于扩展且具有较好的性能。

#### 3.1 注册服务器

注册服务器主要包括:认证、簇首管理、数据处理与分析、通讯等功能模块,见图 5。

簇首管理模块维护一个全局列表,包括系统内所有簇首的状态与信息(如果存在),该模块根据节点的 IP 归属地对节点进行分类,由于目前 IP 识别的粒度为城市级别,所以按照不同的城市划分为不同的簇。当某个簇的簇首发生变更时,该模块自动更新对应簇首的状态和信息。为了加快节点信息的查询速度,我们使用哈希算法来实现节点信息的存储与查询,同时将更新作用于数据持久层。

认证模块负责验证某个节点的身份,引导各个节点加入到整个系统并根据各自的状态各司其职。

数据处理与分析模块负责收集各簇内执行的测量任务的结果,对结果进行持久化处理,对外提供查询接口及可视化界面。为了简化操作,我们使用 MongoDB 来实现数据的存储,使用 SQL 来实现数据的存储和处理。

通讯模块负责收集来自其它节点的消息、对通讯协议的解析以及消息的分发。我们使用基于 epoll 实现的 IO 复用模型,既可以保证良好的并发性,还能避免传统多线程/多进程模式中频繁创建、销毁线程/进程带来的开销,从而提高底层通信模块的性能。

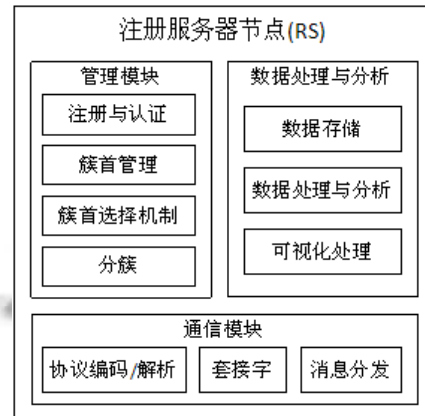


图 5 RS 结构

#### 3.2 簇首节点

簇首节点主要包括:簇首管理、簇内节点管理、备选簇首管理、通讯、任务调度、数据处理等功能模块,见图 6。

簇首管理模块维护了一个列表,包含若干其它簇首节点的状态与信息,用于消息在上层 DHT 网络的高速转发,由注册服务器返回的信息初始化,随后通过主动发送而获取其它簇首节点的响应以及被动接收其它节点的查询完成对列表中各个簇首状态的更新。为了加快节点信息的查询速度,我们使用哈希算法来实现节点信息的存储与查询。

簇内节点管理模块维护该簇的所有可用普通节点的信息,通过接收普通节点的加入请求完成初始化,随后通过被动式心跳协议,接收来自簇内各个普通节点发出的心跳包来更新列表中各个节点的状态。为了高效管理和查询簇内的可用节点信息,使用哈希算法来加快查询速度。

备选簇首维护模块根据簇首选择机制,从可用簇内节点列表中选择一个节点作为备选簇首,一旦簇首意外退出,备选簇首自动成为新的簇首。此外,簇首为列表中的每一个节点动态维护一个时间戳属性,每次接收到心跳包后,用当前的时间更新节点的时间戳;同时使用另外一个线程定时检测,通过比对当前时间和节点时间戳的时间差,若时间差大于某个阈值,则

判定节点已下线, 并从列表中删除对应的信息.

备选簇首管理模块负责从可用普通节点列表中选择-一个作为备选节点, 用于在簇首意外退出后接任为新的簇首.

任务调度模块负责解析和分发任务. 所有的任务先到达任务队列, 然后按照先进先出的原则逐个解析每个任务. 由于大部分任务属于重复性任务, 所以先在缓存和数据库中查找, 若命中则直接提取并返回任务结果, 反之则进入正常的任务查询流程: 首先在上层 DHT 网络中查询并获得对应的目标簇首, 然后通过目标簇首获取该簇的若干可用普通节点的信息.

数据处理模块负责收集簇内各个任务的测量结果, 合并压缩后定时发送给注册服务器做进一步处理.

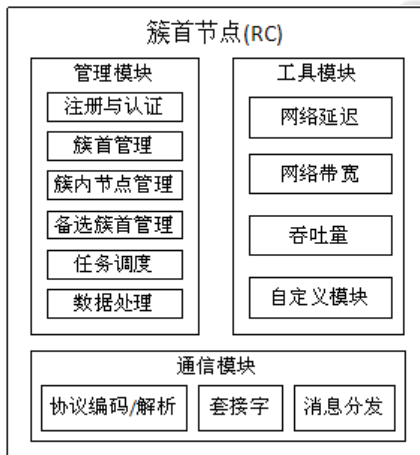


图 6 RC 结构

### 3.3 普通节点

普通节点既可以发起测量任务, 也可以作为服务节点接受任务调度, 它包括通讯、任务管理、工具、心跳包管理等功能模块, 见图 7.



图 7 EP 结构

任务管理模块负责解析任务、向簇首节点请求任务、接收返回节点后发起任务.

工具模块包含各个测量功能的实现, 如网络时延、网络带宽、吞吐量等, 也允许自定义扩展. 我们使用订阅者/发布者模式实现了工具的可配置化, 既提高了模块的可维护性, 也保证了模块的性能.

心跳包模块负责定时向簇首发送心跳包, 辅助簇首完成对簇内可用普通节点的动态维护.

## 4 实验与结果

我们将注册服务器(61.187.64.7)设在位于湖南长沙的湖南大学信息科学与工程学院的实验室主机中, 然后分别在长沙和武汉各自启动 5 个节点加入到系统. 在注册服务器的引导下, 按照时间顺序, 第一个加入系统的长沙节点(61.187.55.100)成为长沙簇首(RCs), 另一个先加入系统的武汉节点(61.183.207.100)成为武汉簇首(RCwh), 随后加入的其它 8 个节点分别向所属的簇首 RCcs、RCwh 报道并加入到簇内结构, 完成了系统的初始化工作, 如表 1 所示.

表 1 节点列表

	数量	位置
RS	1	61.187.64.7(Changsha)
RCcs	1	61.187.55.100(Changsha)
RCwh	1	61.183.207.100(Wuhan)
EP	8	Changsha, Wuhan

使用该系统, 我们测量了长沙到武汉的网络延迟, 以下是本次实验的结果:

首先, 在某个长沙的节点(61.187.64.118)上, 为了测量长沙到武汉的网络延迟, 我们启动程序向注册服务器发起注册请求. 注册服务器接收到注册请求后, 通过 IP 识别该节点属于长沙, 然后检测簇首列表, 由于长沙簇首已存在, 所以返回簇首信息, 根据返回的簇首信息, 自动向簇首进行报道并加入簇内网络.

注册完成后, 由于我们要进行一次端到端的网络测量, 所以我们需要获得一个武汉的节点, 于是我们向簇首节点 RCcs 发起任务请求, RCcs 接到任务请求后解析任务, 识别目标节点属于武汉, 于是检查簇首列表中是否存在武汉簇首, 如果不存在, 则在上层 DHT 网络中转发查找. 在获得武汉簇首节点 RCwh (61.183.207.100)后, 先更新簇首列表, 然后向武汉簇首 RCwh 请求若干可用普通节点. 武汉簇首 RCwh 获

得请求后,从维护的可用普通节点列表中返回若干,长沙簇首 RCes 获得响应结果后返回给任务发起者。

于是我们从簇首 RCes 获得了某个武汉节点 (61.183.207.49),然后系统自动调用内置工具向对方发起测量任务,测量结果如图 8 所示。

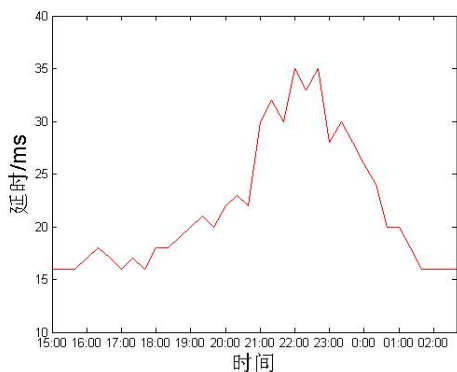


图 8 延迟随时间变化

## 5 结语

本文设计了一个基于混合 P2P 模式的端到端网络测量系统,实现了测量节点的自动化部署,在保证查询速度的同时,尽可能降低单点故障,提高系统的可扩展性。当然,系统本身还有需要完善的方面,下一步任务是优化簇内的动态簇首维护机制,降低模块的复杂度。

### 参考文献

- 1 De Roure D, Gil Y, Hendler J. Guest editors' introduction: E-science. *Intelligent Systems IEEE*, 2004, 19(1): 24-25.
- 2 张潇丹,李俊.一种基于云服务模式的网络测量与分析架构. *计算机应用研究*,2012,29(2):725-729.
- 3 杨家海,吴建平,安常青. *互联网络测量理论与应用*.北京:人民邮电出版社,2009.
- 4 高磊,杨家海,张辉,等.基于 Netflow 的网络流量测量基础设施建设. *广西大学学报(自然科学版)*,2011,36(A01):78-82.
- 5 Kalidindi S, Zekauskas MJ. Surveyor: An infrastructure for Internet performance measurements. *Proc. of the INET'99*. San Jose. 1999. 1-6. [http://www.isoc.org/inet99/Proc./4h/4h\\_2.htm](http://www.isoc.org/inet99/Proc./4h/4h_2.htm).
- 6 Xu D, Liu T, Qian D, et al. A new P2P-like architecture for large scale end to end network measurement. *International Conference on Systems and International Conference on Mobile Communications and Learning Technologies. ICN/ICONS/MCL 2006*. IEEE. 2006. 62-62.
- 7 Anderson DP, Cobb J, Korpela E, et al. SETI@home: An experiment in public-resource computing. *Communications of ACM*, 2002, 45(11): 56-61.
- 8 Turner S, Hobson J, D'Auria D, et al. DipZoom: The internet measurements marketplace. *Infocom IEEE International Conference on Computer Communications*. IEEE. 2006. 1-6.
- 9 Rowstron A. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms Middleware*. 2001. 329-350.
- 10 吴麒,冯伟森.基于 Kademia 协议的 DHT 网络的测量研究. *四川大学学报(工程科学版)*,2011,43(5):114-119.