

基于 P2P SIP 协议的即时通信系统^①

Instant Message System Based on P2P SIP Protocol

白羽¹ 洪飞² (1.北京建筑工程学院理学院 北京 100044; 2.北京航空航天大学
计算机学院 北京 100083)

摘要: SIP 协议必须有中央服务器的支持,带来了需要手工配置、维护,以及扩展性较差等问题,本文在系统分析 P2P SIP 协议以及相关技术的基础上,给出了基于 P2P SIP 协议即时通信系统的设计,该系统采用基于 P2P 的方法来进行 SIP 注册和资源定位,保证系统可以在一个没有中央服务器的环境中自由通信,测试表明该系统依托 P2P SIP 协议构建的网络可以实现网络自治。

关键词: P2P SIP 即时通信 DHT

目前大多数的即时通信系统采用集中式的客户/服务器模式,其通信实现主要是基于 XMPP 协议和 SIP 协议。在客户/服务器模式下,系统利用一个中央化的资料库,给用户建立一个固定的帐号、以及动态 IP 地址间的映射。由于动态 IP 的网络用户在重新连接网络时 IP 地址改变,因此中央资料库需要搜索每一个使用者帐号以及 IP,并判断是否上线。这种集中式目录在用户基数增加到百万以上之后,资源耗费会非常大,出现搜索失败,服务器停止响应等情况,为了避免这种情况,需要通过大量的服务器资源进行信息集中存储、通讯协调等^[1]。

近年来,基于 P2P 的即时通信技术的研究成为新的热点。互联网的 P2P 技术自从出现以来一直受到广泛的关注,它不采用集中式的客户/服务器模式,有效地避免了上述采用客户/服务器模式的通信系统的弊端^[2]。

1 P2P SIP 协议

基于 P2P 的即时通信技术有两个发展方向,一是利用私有通信协议;另一个是利用标准通信协议,考虑到协议的可扩展性和简单性,SIP 协议被广泛采用。

基于 SIP 协议的 P2P 即时通信采用标准通信协议,因此容易实现与现有的 SIP 系统的兼容。P2P SIP 可以很好的解决现有的即时通信应用架构的不足,具有广

阔的发展空间,值得深入研究充分挖掘它的潜在优势,在即时通信和新的应用中发挥它的作用。目前 P2P SIP 的研究还是一个比较新的领域:IETF 在 2005 年发布了一些相关草案,威廉和玛丽学院和哥伦比亚大学都正在开展 P2P SIP 应用开发的研究,基于 P2P SIP 的技术仍然处在探索和完善阶段。

P2P SIP 协议是基于 P2P 的即时通信应用技术。在网络中端点不仅实现传统的呼叫和接受呼叫的功能,而且一些端点的集合实现了传统 SIP 网络中的注册服务器和代理服务器的功能,包括资源定位,维护位置信息,以及进行呼叫路由等。相比 SIP 终端,P2P SIP 端点具有更高的智能。端点直接与其他一些端点连接成一个重叠网络,每一个端点都保存了传统注册服务器或代理服务器上的一部分信息。它们之间用 SIP 消息进行通信并且能够相互提供服务。利用 P2P 网络模型与 SIP 协议来创建即时消息系统在各方面具有突出的优势^[3]。

在性能上,P2P SIP 实现了在 Internet 架构下的对等网络,它充分利用互联网的边缘资源,即用户的计算能力、存储能力和带宽,改善了通信的流量分布,可以减轻服务器的负担,让系统执行更加顺畅,改善通话音质;甚至,非集中式网络中每增加一个节点,网络的处理能力和带宽都相应增加,对整个系统而言

^① 基金项目:国家自然科学基金(60503037);北京建筑工程学院院科研基金(100704607)

收稿时间:2008-08-21

有相当大的好处；在可靠性上，由于在网络中没有任何服务器，P2P 网络有望变成规模可变的同时对故障拥有高度抵抗力的网络；在网络部署上较为简单，由于网络中没有服务器，基于 P2P SIP 的系统部署起来十分方便，可以在临时应用时迅速构建一个通信网络；在可扩展性上，由于应用 SIP 协议，应用程序具有较强的可扩展性，可以定义并迅速实现新功能。该协议还可以提高不同应用程序互操作的能力[4]。

2 系统设计

P2P SIP 即时通信系统的设计思想是尽可能的基于标准的 SIP 消息实现 P2P 网络的所有操作，从而使系统兼容现有的 SIP 应用。在这个系统中，所有维护查询表的消息，以及用于信息查询的消息是利用 SIP 消息实现的。SIP 消息的交换用来完成两个目的：一是在节点加入或离开重叠网时维护查询表信息、在节点间传递信息；二是实现重叠网的用户之间的通信，如注册用户，邀请其他用户加入会话等。由于 DHT 可以被当成分布式注册服务器，注册和查询是由 DHT 完成的，一旦目标资源被找到，节点之间就只用基本的 SIP 信令进行通信。这样就将具有 DHT 功能的 P2P SIP 节点和普通 SIP 节点相隔离开来。普通节点只用基本的 SIP 消息通信，传统的不具备 P2P 功能的 SIP 终端会选择一个 P2P SIP 节点作为代理加入 P2P SIP 网络，普通的 SIP 终端可以通过节点代理与其它 P2P SIP 终端进行通信。所以该系统可以很好的兼容现有的 SIP 终端。

本文的即时通信系统是一个支持即时通信的 P2P 架构的 SIP 终端系统，我们实现了一个 P2P SIP 的用户节点 (SIPPeer)，整个即时通信系统实际上就是由这些对等的 SIPPeer 组成，P2P SIP 网络中会提供注册服务器和代理服务器的功能，它可以保证每一个 SIPPeer 随时加入网络并且与其它节点进行通信，任何一个节点都可以随时退出网络而不会影响其它节点的通信。

本系统主要由用户界面模块、用户自我管理模块、DHT 功能模块、SIP 功能模块四部分组成，系统的结构图见图 1。其中，用户界面模块是向用户提供各种界面操作功能，可以保存用户的“好友名单”并且调用用户自我管理模块去定位这些好友的位置，还提供了发送即时消息、注册、注销、登陆等操作。用户自我管理模块架设在用户界面模块与 DHT 功能模块、SIP 功能模块之间，它相当于一个 SIP 注册服务器和一个代

理服务器，负责将用户界面提交的操作转换成各类底层方法调用或者消息传递。用户的寻找、加入、离开的请求都由用户自我管理模块来调用 DHT 功能模块来实现，DHT 功能模块维护了对等节点之间的信息 (Chord Finger Table)，它提供了网络进行通信的底层拓扑结构。REG、INVITE 以及其它的一些消息都直接传递给 SIP 功能模块去处理。SIP 功能模块是整个系统消息传递的基础，它用来进行 DHT 路由信息的维护、定位用户、注册用户、呼叫建立和即时消息传送。

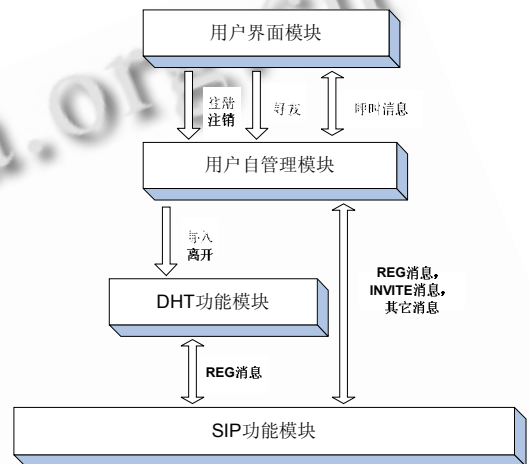


图 1 基于 P2P SIP 的即时通信系统结构图

类图 (图 2) 表示了组成 SIPpeer 的各个类之间的关系。

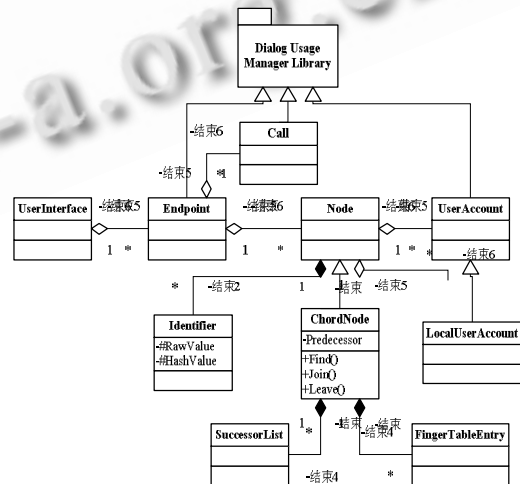


图 2 SIPPeer UML 类图

UML 类图中的 Endpoint 类代表了一个 P2P-SIP 的节点，它继承自 DUM 类库中的 DialogUsageManager 类。Endpoint 类的实例会创

建一个实现了 DHT 算法的 ChordNode 类的实例。Endpoint 对象中包含一个 LocalUserAccount 对象列表和 UserAccount 的映射表。Endpoint 可以获取所有的事件，例如：加入完成事件，或者节点信息改变。

UserInterface 对象实现了用户操作界面。可以在用户窗口执行注册、登陆、发送消息、退出等相应命令操作。

Call 类继承自 DUM 类库的 TransportSelector 类，Call 类负责呼叫控制、设置获取注册状态机、事件通知、呼叫转移等功能。DUM 类库只支持 SIP URI 名称的 DNS 解析功能，因此我们在 CALL 类中添加了对于 P2P 对应的哈希算法的解析的支持。

Identifier 类是一个内部类，它用来存储节点标识或者用户标识的原始值和哈希过的值。例如：Identifier 实例中可以存储一个节点标识 8@node.com 以及调用上述 Call 类中哈希方法返回的哈希值 8。而 Identifier 相应也可以存储 alice@yahoo.com，哈希值 24 的用户标识信息。我们目前将所有形如 数字@主机地址 的节点标识识别为节点标识，其它的标识为用户标识。

Node 类用来定义 DHT 中的一个节点。Node 类中定义了 Find、Join、Leave 虚函数。而子类 ChordNode 使用 Chord 算法实现了 Find、Join、Leave 函数，并且其它 Chord 相关的数据结构。如类图所示这些数据类都是 ChordNode 的组成部分，其中 FingerTableEntry 类代表了 FingerTable 中一项，它包含一个 IdentifierRange，用来表示[起始-终止]的范围，而 NextHOPValue 用来表示下一跳的节点标识。ChordNode 中的 FingerTable 容器可以将一组编号了号的 FingerTableEntry 项组成一个 FingerTable。ChordNode 中的 Predecessor 属性用于存储前驱节点编号，而后继节点则存储在 SuccessorList 链表中。

UserAccount 类用来存储远程用户注册信息。LocalUserAccount 类用来存储本地用户注册信息。其中 UserAccount 类继承自 UDM 中的 Registration 类，而 LocalUserAccount 类又继承自 UserAccount 类。一个 Node 对象中会包含一组 UserAccount 对象和一组 LocalUserAccount 对象。

Endpoint 使用类库中不同的对象使的 SIPpeer 实现了 SIP 用户代理 (UA)，代理服务器 (proxy) 和注册服务器(Registar)的功能，所有的状态和注册信息都存储在内存当中，没有设计外部数据库去存储它们。

DHT 功能是由 ChordNode 类来实现的，其中封装了 Chord 算法。系统启动的时候 Endpoint 会创建一个 ChordNode 对象。DHT 功能模块可以分为初始化、节点发现、加入、稳固、节点关闭、节点故障

处理子模块。

用户自我管理模块包含了用户注册、登陆、注销、呼叫建立消息代理等子模块。

DHT 管理中节点加入，退出和查询都是通过节点发送包含 user=node 参数的 REGISTER 消息来实现的。该消息目的节点根据消息格式来区分具体的操作，比如 expire 头域为 0 表示节点删除，不包含 Contact 头的 REGISTER 消息表示查询。其中，允许节点加入的接纳节点要将一部分用户信息转移到其他节点，对应图中的 User Register 模块。发送 REGISTER 消息的加入节点会根据响应消息改变自己的后继节点。

为了防止节点非正常退出影响系统稳定性，每个节点周期性的查询它的后继节点，查询节点收到 200 OK 响应消息后判断其中的前驱节点是否为自己，如果不是则删除后继节点并向这个前驱节点发起注册。

3 系统测试

本文系统的验证采用简化的网络，这个网络不是采用 SHA-1 算法生成 2^{160} 的名字空间，而是用 4 位的哈希生成大小为 16 的名字空间。这个系统中的 NodeID 和 ResourceID 是 0 到 15 之间的数。在测试系统中使用 Finger 表全部的 4 个入口，而在实际系统中，建议 Finger 表的行数小于哈希位数。系统测试的部署如图 3 所示：

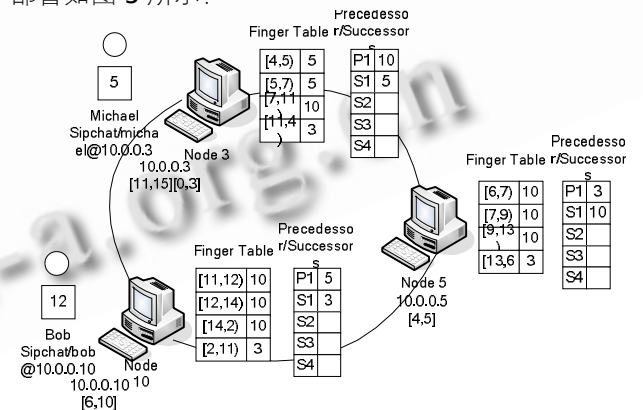


图 3 系统测试部署图

网络中已有 3 个节点。它们的 NodeID 分别为哈希值 3, 5, 10。每个节点的 Finger 表的 4 个入口给出查找节点范围在 $[NodeID + 2^i, NodeID + 2^{(i+1)})$ 与查找请求发送到节点之间的对应关系。比如，对于节点 3 来说，查找节点 4，节点 5，节点 6 的消息应发送给节点 5；查找节点 7，节点 8，节点 9，节点 10 的消息应发送给节点 10；查找节点 11，节点 12，节点 13，节点 14，节点 15，节点 1，节点 2，节点 3 的消息应发送给节点 3。

资源键值为 ResourceID 的信息存储在顺时针方向找到的第一个 NodeID 大于或等于该 ResourceID 的节点上。在这个测试中, 节点 3 负责 ResourceID 在 11 至 15, 0 至 3 范围内的资源信息。节点 5 负责 ResourceID 在 4 至 5 范围内的资源信息, 节点 10 负责 ResourceID 在 6 至 10 范围内的资源信息。

所构成的重叠网名字为 sipchat, 两个用户已经注册。用户 Michael 的 ResourceID 为 5, 所以他的联系信息由节点 5 负责保存。用户 Bob 的 ResourceID 为 12, 他的联系信息则由节点 3 来保存。Bob 的 UA 在节点 10 上, 他的联系信息为 sipchat/bob@10.0.0.10; Michael 的 UA 在节点 3 上, 地址为 10.0.0.3, 他的联系信息为 sipchat/Michael@10.0.0.3。

3.1 节点加入注册

假设一个 IP 地址为 10.0.0.14 的节点想要加入系统, 通过哈希算法得到 NodeID 为 14。节点 14 通过调用 Peer Discovery 方法知道节点 5 的存在。它向节点 5 发送 REGISTER 消息。节点 5 收到消息, 验证 10.0.0.14 对应 NodeID 是 14 是正确的, 它比较 14 是否在前驱节点和自己的 NodeID 之间, 它确定自己不负责 14 后查看 Finger 表, 查出应该将请求发送给节点 3。它向节点 14 发送 Contact 值为节点 3 的 302 响应。

节点 14 向节点 3 发送新的 REGISTER 消息, 节点 3 验证 10.0.0.14 对应 NodeID 是 14 是正确的, 在确认自己负责节点 14 的注册之后, 回复 200 OK 消息。最后, 节点 3 上的 Bob 信息应该转移到新加入的节点 14 上, 节点 3 代表 Bob 向节点 14 发送第三方用户注册消息, 来转移用户注册信息, 节点 14 响应 200 OK 消息。

3.2 节点退出

节点 14 加入网络, 系统稳定后, 它的前驱节点为节点 10, 后继节点为节点 3。将节点 14 上运行的程序停止, 它发送 REGISTER 节点删除消息给节点 10 和节点 3。这两个节点响应 200 OK。

3.3 用户注册

用户 Alice 在节点 5 上启动 UA。Alice 的联系方式是 Alice@10.0.0.5, 她的用户名是 Alice@infolead.cn。节点 5 对 Alice@infolead.cn 哈希得到它的 ResourceID 为 11。Alice 的用户信息应保存在负责 ResourceID 为 11 的节点, 在这个验证中应是节点 3。

Alice 的 UA 按照前面资源注册的方法向负责 ResourceID 为 11 的节点发送 REGISTER 消息。Alice 的 UA 查看它的 Finger 表, 判断应将注册请求发给节点 10。它发送请求给节点 10, 节点 10 确认自己不

负责 ResourceID 为 11 的注册, 然后查看 Finger 表, 判断应将注册请求发给节点 3。它回复 Contact 为节点 3 的 302 消息。节点 5 替 Alice 生成新的 REGISTER 消息, 发给节点 3。节点 3 确认自己负责 Alice 的注册, 回复 200 OK 消息, 消息中含有节点 3 的一些后继节点, Alice 可以向它们发送冗余的注册请求。

3.4 会话建立

我们使用 Bob 呼叫 Michael。Bob 所在的节点对 Michael 的用户名进行哈希计算, 得到 Michael 的 ResourceID 为 5。Bob 的节点查询 Finger 表, 确定呼叫 ResourceID 为 5 的请求应该发向节点 3。它生成 REGISTER 查找消息并把消息发给节点 3。节点 3 判断自己不负责 Resource 为 5 的节点, 它查看 Finger 表确定应该把请求发给节点 5, 它发回 Contact 设为节点 5 的 302 响应。Bob 所在的节点向节点 5 重新发送 REGISTER 消息, 节点 5 负责保存 Michael 的信息, 它发送 200 OK 响应, 响应消息的 Contact 为 Michael 的联系地址 — sipchat/Michael@10.0.0.3。Bob 最后向 Michael 的 UA 发送 MESSAGE 请求, 然后依照标准的 SIMPLE 协议进行通信。

4 总结

本系统实现的 P2P SIP 即时通信系统充分利用了标准的 SIP 协议和日趋成熟的 P2P 技术: 前者基于标准协议, 又易于扩充; 后者实现智能节点, 有传统信息共享的技术优势, 两种技术的融合, 互为补充, 为用户提供一体化的应用和服务。

参考文献

- 1 Bryan DA, Lowekamp BB. SOSIMPLE: A SIP/SIMPLE Based P2P VoIP and IM System. : Williamsburg, Computer Science Department, College of William and Mary, 2004:4-5.
- 2 Waterhouse SR, Doolin DM, Kan G, Faybishenko Y. JXTA Search: a distributed search framework for peer-to-peer networks. IEEE Internet Computing, 2002,6(2): 68-73.
- 3 Stoica I, Morris R, Karger D, Kaashoek F, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications. Proceedings of conference in ACM Special Interest Group on Data Communication. San Diego:ACM press. 2001:4-5.
- 4 吴新安,王芙蓉,戴彬.基于 P2P 的 SIP 研究与实现.现代计算机,2006,9(1):1-5.