

层次化移动 IPv6 环境中的 QoS 机制^{*}

赵艳琼 杨寿保

(中国科学技术大学计算机科学与技术系 合肥230026)

摘 要 本文针对层次化移动 IPv6 的网络提出了一种 QoS 信令,为移动节点提供服务质量的保障。在分析了 RSVP 信令的特点及其由于对多播的支持而增加的复杂性的基础上,设计了一个简化的 RSVP 信令,然后把此 RSVP 信令和层次化的移动协议相结合提出了一个综合的 QoS 切换方案,并在层次化的移动环境下,对此 QoS 信令和其它几种 QoS 机制在资源预留时间和信令负荷方面进行了性能比较。

关键词 移动 IPv6,服务质量,层次化移动 IPv6,资源预留协议

Hierarchical Mobile IPv6 Network QoS Scheme

ZHAO Yan-Qiong YANG Shou-Bao

(Department of Computer Science, University of Science and Technology of China, Hefei 230026)

Abstract This paper describes a scheme for the provisioning of QoS in the Hierarchical mobile IPv6 network. We analyze the features of RSVP and it's complexity due to multicast support. Then, based on this analysis, we present a new simple RSVP signaling and combined with Hierarchical Mobile IPv6 protocol to improve QoS performance in the handoff scenario. Finally, we compare the performance of our new QoS scheme with other schemes in the Hierarchical Mobile IPv6 network in terms of resource reservation time.

Keywords MIPv6, QoS, HMIPv6, RSVP

1 引言

随着移动 IPv6 (Mobile IP version 6, MIPv6) 技术的快速发展,用户希望在移动和固定网络中获得同等质量的服务。一些多媒体业务,如视频点播、VoIP 等,对网络的带宽、延迟和延迟抖动比较敏感,希望网络能够提供一种稳定的传输服务,这就需要在移动 IPv6 网络上导入服务质量 (Quality of Service, QoS) 保证技术,以确保各项业务的通信质量,满足用户的需求。

目前在 IP 网络中提供 QoS 保证的技术主要有区分服务和综合服务两种,区分服务主要应用于聚集流,通过对数据流分类、赋予不同的优先级来分配网络资源。综合服务模型^[1] (Integrated Service, IntServ) 主要应用于单个数据流,根据业务的需求为每个数据流分配资源。资源预留协议 RSVP^[2] (Resource Reservation Protocol) 是 IntServ 实现 QoS 的关键, RSVP 采用了一种显示的资源预留方式,对单条数据流实行端到端的资源预留,从而满足实时服务的要求。

由于移动环境的特殊性,在移动环境中采用 IntServ/RSVP 模型实现 QoS 存在很多困难^[3],因为移动节点 (Mobile Node, MN) 在一次会话中会多次变换网络的接入点,不停地切换,为了保证服务质量, RSVP 需要新的链路上重新进行资源预留,这就带来了操作的复杂性。除此之外,移动环境中还存在着其它因素影响 QoS 实现,如: IP 地址的频繁变换,无线链路自身状态的随时变化,以及移动节点之间资源的争夺等问题。

迄今为止,移动协议和 QoS 机制都在各自独立地发展,

RSVP 只是通过周期性地发送刷新信令来修补变化了的路径状态,并不能及时地发现移动带来的路由变化。为了解决上述问题,本文提出了一个综合的 QoS 信令,根据移动环境的特点对 RSVP 信令进行简化,并把它与层次化移动 IPv6 协议^[4] (Hierarchical Mobile IPv6, HMIPv6) 相结合,这使得移动节点在切换后可以快速地建立资源预留,并减少网络的信令负荷。本文介绍 RSVP 信令的特性及 RSVP 在移动环境中具体的工作过程,说明采用 RSVP 实现 QoS 的缺陷;分析 RSVP 信令的构成,设计一种简化的 RSVP 信令;把这个简化的 RSVP 信令与层次化移动 IPv6 协议相结合提出一个 QoS 切换方案,并描述了此方案在 HMIPv6 环境中的具体操作过程;对 QoS 解决方案和其它两种 QoS 机制作具体的性能比较。最后进行了总结。

2 RSVP 信令的特点及其在移动环境中的工作过程

本节具体介绍 RSVP 协议的基本特性和移动节点在切换时使用 RSVP 协议进行资源预留的操作过程。

2.1 RSVP 信令的特点

RSVP 协议是为固定网络中的综合服务模型所设计的资源预留协议,它通过端对端的资源预留来实现 QoS,并且支持多播通信。它具有如下的基本特性:

(1) 单向性: RSVP 协议在业务流的源端和目的端之间建立资源预留,但它只是做单方向的资源预留,如果是一个双向通信,必须建立两个方向的资源预留。

(2) 目的端发起: 为了容纳不同种类的目的端, RSVP 采用目的端预留的方式,源端发送 PATH 消息收集链路状态信

^{*} 本文得到国家863项目(编号:2001AA121041)的资助。赵艳琼 硕士研究生,研究方向为移动 IPv6,计算机网络。杨寿保 教授,博士生导师,主要研究方向为网络与移动计算、网络安全。

息,目的端收到 PATH 消息后返回 RESV 消息在链路上进行资源预留。

(3)路由独立性:RSVP 并不为数据流的包选择路由,它利用通常的路由协议来决定数据流的路径。

(4)软状态:网络中的预留状态是有时间限制的,如果时间到期了,预留状态就会被删除,这就是所谓的软状态。RSVP 周期性地发送 PATH 和 RESV 消息刷新、维持预留路径上的路径状态与预留状态。

IntServ/RSVP 的上述特性确保了单条数据流端到端的服务质量,但在移动 IPv6 环境中使用 RSVP 却产生了很多问题,下面看一下 RSVP 与 MIPv6 的交互操作过程。

2.2 RSVP 在移动 IPv6 环境中的工作过程

移动环境如图1所示。路由器 A 和路由器 B 分别负责一个子网,移动节点开始在路由器 A 接入的子网 A 中,移动节点与通信节点之间的路径上已经预留过了资源,现在移动节点从子网 A 移动到子网 B,我们观察移动节点切换后 RSVP 与 MIPv6 信令的交互操作过程。

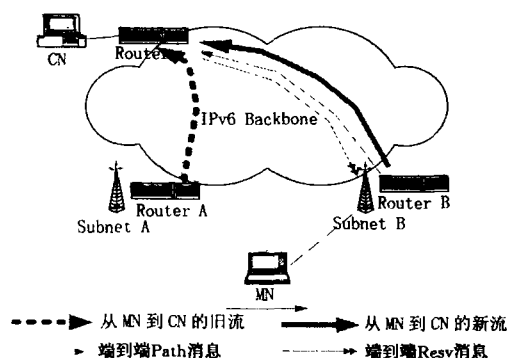


图1 移动节点作为数据流源端时 RSVP 的操作过程

(1)当移动节点从子网 A 切换到子网 B 后,会获得一个属于子网 B 的新的转交地址,于是移动节点发送一个绑定更新消息(Binding Update, BU)给通信节点,通信节点收到 BU 后更新绑定缓存,返回一个绑定确认消息(Binding Acknowledgment, BA)。

(2)因为从移动节点到通信节点的路由发生了变化,RSVP 需要在新的路径上重新进行资源预留。当移动节点作为数据流的源端时,为了在新路径上预留资源,要发送一个 PATH 消息到通信节点,中间的路由器对 PATH 消息进行处理,检查自身有无足够的资源满足业务的需求,这样,PATH 消息收集了沿途的链路状态信息,当通信节点收到 PATH 消息后,原路返回一个 RESV 消息进行资源预留的操作。如果沿途的路由器拥有足够的资源,从移动节点到通信节点这条新路径上的资源预留就建立成功。

(3)同理,当移动节点作为数据流的目的端时,通信节点会发送一个 PATH 消息到移动节点,移动节点收到 PATH 消息后,返回一个 RESV 消息。如果中间的路由器的资源满足需求,从通信节点到移动节点路径上的资源预留也建立了起来。

(4)当移动节点在同一个小区之内移动时,它的转交地址保持不变,这时,RSVP 的工作过程与在固定网络中相同,它采用软状态的方法维持链路上的资源预留状态,周期性地发送 PATH 和 RESV 信令对路径上的路径状态与预留状态进行刷新。

从上述操作过程可以看出 RSVP 信令在移动环境中使用的缺陷:

(1)建立资源预留需要的信令多。RSVP 采用接收者驱动来建立资源预留,建立一个方向的资源预留需要 PATH 和 RESV 两条消息的交互,建立资源预留的时间长,信令多,加重了中间路由器的处理负载。另外,RSVP 采用软状态的方法维持链路上的资源预留状态,它周期性地发送 PATH 和 RESV 信令对路径上的路径状态与预留状态进行刷新,这种频繁发送的刷新信令无疑会给无线链路带来很大的信令负荷。因此需要对这种资源预留的方式进行改进。

(2)切换时建立资源预留的时间长。移动节点进行切换后,路径状态发生了变化,这样,在新的链路完成资源预留之前,移动节点所进行的业务在新链路上只能获得尽力而为的服务,得不到 QoS 的保障,所以要求移动节点在切换后建立资源预留的时间愈小愈好。但是,基本的 RSVP 信令只是通过周期性地发送刷新信令来修补变化了的路径状态,并不能及时地发现节点的移动,这就使得移动节点在切换后会出现数据流的抖动、丢包等情况,极大地影响实时业务的服务质量。

所以,本文提出对移动节点在切换时使用 RSVP 协议进行资源预留的操作机制进行改进,把移动协议和 RSVP 协议结合在一起,这样可以在切换后快速地建立资源预留。

3 RSVP 信令的改进

本节我们对 RSVP 信令进行分析,去除其对多播的支持,提出了一种简化的 RSVP 信令,减少网络中的信令负载和路由器的处理负荷,使之适应移动环境的需求。

3.1 RSVP 信令分析

虽然 RSVP 支持单播通信的资源预留,但是,RSVP 的许多特性都是为了支持多播通信而设计和优化的^[5],例如:RSVP 信令中的 reservation styles 和 scope 对象,由接收者发起资源预留的特点和路由器中的复杂的状态管理等等。这些设计大大增加了信令的复杂性和路由器的处理负荷。所以就带来这样一种情况,对于一个简单的单播应用,不得不按照多播的方式来处理,事实上,现实生活中的很多应用都是单播通信,根本不需要对多播的支持。现在有很多支持多播的协议(如:PIM-SM)都开始考虑把多播的应用作为在单播基础之上的一个功能部件,而不是把它们作为一个整体,因为对于 Internet 中的信令协议,它需要经过许多节点,从信令的可扩展性考虑,它的设计应该尽量简单,对于不同的情况都可以提供简单而且完备的信令服务^[6]。根据上述原则,我们针对单播应用对 RSVP 信令进行改进,把对多播的支持作为其附加的功能,设计了一种简化的 RSVP 信令协议。

3.2 简化的 RSVP 信令

3.2.1 信令结构 观察 RSVP 协议版本 RFC2205 所定义的 RSVP 对象,主要有 SESSION、SENDER_TEMPLATE、SENDER_TSPEC、ADSPEC、RESV_STYLE、FLOW_SPEC、FILTER_SPEC 和 SCOPE 等。这些对象包括对源端数据流的定义(如:FILTER_SPEC、SENDER_TEMPLATE);对所需 QoS 的描述(如:SENDER_TSPEC、FLOW_SPEC)以及为了支持多播而增加的对象(如:RESV_STYLE、SCOPE)。在单播环境中可以把 RESV_STYLE、SCOPE 等对象取消。重新定义的 RSVP 信令如下:每个 RSVP 消息以一个通用头标开始,后面跟着一系列 QoS 对象^[7],每个 QoS 对

象对应着源端一个单向的数据流。QoS 对象的结构如图2所示,由以下几部分组成:

(1)数据流的定义:在 IPv6 中采用流标记和源端 IP 地址唯一确定。

(2)QoS 的需求类型:定义用户需要的是保证型服务还是

负载受控型服务。

(3)QoS 需求的具体描述:把 PATH 消息中的 SenderTSPEC 对象和 RESV 中的 FlowSpec 对象统一起来,包括延迟、延迟抖动、数据平均传输速率、令牌桶容量、最大包的长度、最小包的长度和突发业务的最大速率等。

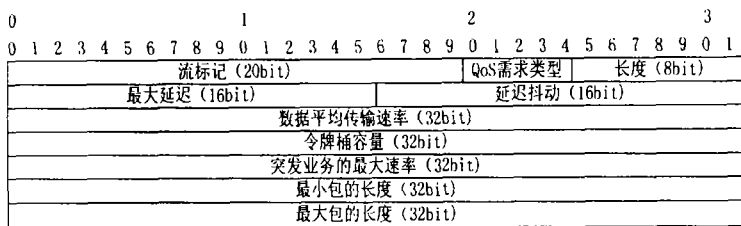


图2 QoS 对象的结构

3.2.2 消息类型 只针对单播应用后就可以改变由接收者驱动进行资源预留的方式,采取发送者驱动的资源预留。由源端发送一条简化的 RSVP 信令给目的端,沿途的路由器进行处理,资源满足需求就直接建立资源预留。一条信令便可以建立一个方向的资源预留。这种简化的 RSVP 信令只需要两种消息类型:

(1)Request Message:携带源端的 QoS 请求,沿途的路由器对此消息进行处理,直接把所需的资源预留下来,失败返回 Error 消息,成功则不返回任何消息。

(2)Error Message:资源预留失败消息,由预留失败处的路由器发给源端,包含此路由器的 IP 地址。

3.2.3 简化的 RSVP 信令特点 这种简化的 RSVP 信令具有如下特点:

(1)面向单播的设计:去除了 RSVP 消息中的 style 和 scope 对象,减少了路由器的处理负荷。

(2)源端发起资源预留:由源端直接发送 QoS Request Message,一条信令即完成资源预留,不同于 RSVP 由接收端进行的资源预留,需要 PATH/RESV 两条信令的交互,减少了信令数,缩短了预留时间。

(3)软状态:同基本的 RSVP 一样,这种简化的 RSVP 信令也采用软状态的方式来维持路径预留信息。

4 移动环境下的 QoS 解决方案

为了使移动节点在切换后能够尽快地建立资源预留,文[8]提出把移动信令和 QoS 信令松耦合的观点,即在路径变化后,由移动信令的发送来引发 QoS 的操作,从而快速地建立资源预留。本文采用移动信令和 QoS 信令紧耦合的思想,即在移动信令中加入 QoS 的信息,在发送移动信令的同时发送 QoS 信令。

目前常用的移动协议有移动 IPv6 和层次化移动 IPv6 两种。层次化移动 IPv6 对移动 IPv6 协议进行了扩充,引入一种叫 MAP(Mobility Anchor Point)的功能实体,MAP 可以使移动 IPv6 的绑定消息(绑定更新、绑定承认消息)处理限制在本地区域内,减少消息处理的开销,这种移动模型对快速切换的支持较好。

为了支持移动 IPv6 的无缝切换,本文把 HMIPv6 协议和简化的 RSVP 协议结合在一起,提出了一个综合的 QoS 解决方案:在移动消息 BU/BA 里增加一个新的 QoS 选项,它是一个 IPv6 的 hop-by-hop 扩展报头,包含一系列的 QoS 对象,每一个 QoS 对象对应发送端一项业务具体的 QoS 需求。这样,在移动节点进行小区切换时,直接发送 BU+QoS 信令就可以建立起从移动节点到通信节点的资源预留。同理,当通信节点收到 BU 后,向移动节点返回绑定确认信令 BA+QoS,可以建立从通信节点到移动节点的资源预留。图3就是 hop-by-hop 的 QoS 选项。

以图4为例,我们看一下移动节点采用这个 QoS 方案在 HMIPv6 环境中进行域内切换的过程:

(1)移动节点开始在由路由器 AR1 接入的子网1内,链路转交地址为 LCoA1,区域转交地址为 RCoA。然后移动节点开始移动,切换到由路由器 AR2 接入的子网2中,移动节点从接收到的路由器公告知道自己发生了子网切换,于是通过自动配置形成一个新的链路相关转交地址 LCoA2,发送绑定更新消息 BU 给 MAP, QoS Object 选项就随着 BU 发送出去,沿途的每个路由器都对此 hop-by-hop 报头进行处理,检查自身有无足够的资源,如果有的话就直接把资源预留下来,否则就在报头上做个标注,然后把 BU+QoS 消息转发出去,这样,当 BU+QoS 信令到达 MAP 时,它已经在所经过的路径完成了从移动节点到 MAP 的资源预留。

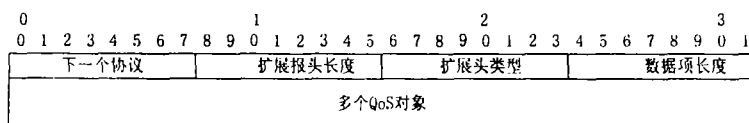


图3 IPv6 的 QoS 选项

(2)当 MAP 收到 BU+QoS 后,更新绑定缓存,然后向移动节点发回一个绑定确认消息 BA,在 BA 中同样包含了 QoS Object 选项,描述了从通信节点发往移动节点各项业务的 QoS 需求,同样,沿途的路由器对这个 hop-by-hop 报头进行处理,把资源预留下来。当 MN 收到 BA 时,从通信节点到移

动节点的资源预留也建立了起来。

(3)如果一个中间路由器检查到自身没有足够的资源满足业务的需求,就会产生一个资源预留失败消息,返回给发送端,在这个消息里会包含它自身的 IP 地址,告诉发送端在哪个地方预留失败。然后此路由器把 QoS Object 报头中一标志

位置位,告诉下面的路由器不再对此 QoS 对象进行处理,然后继续转发给目的端。

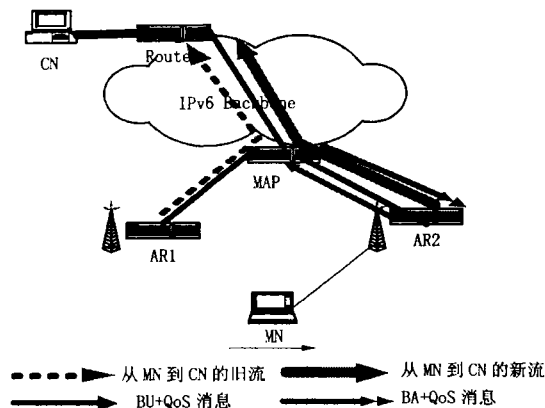


图4 移动节点采用 BU/BA+QoS 信令的切换过程

这种 QoS 方案使得移动节点在发送移动信令的同时建立资源预留,大大缩短了切换后建立资源预留的时间,减少了因切换影响而失去 QoS 保证的报文数,提高了服务质量。

5 QoS 信令时延分析

以 HMIPv6 环境为例,本节分析比较几种 QoS 信令方案在切换时建立新路径上的资源预留所需要的延时,假定从通信节点到移动节点建立资源预留。

5.1 基本的 RSVP 方案的处理延时

基本的 RSVP 通过周期性发送刷新信令进行本地修补,它的处理延时可以用如下公式进行衡量:

$$D_{rsvp} = T_p + D_{path} + D_{resv} \quad (1)$$

其中 T_p 是 RSVP 周期性发送刷新信令的时间, D_{path} 和 D_{resv} 分别是在新路径上传输和处理 PATH 消息与 RESV 消息的延时。

5.2 松耦合的 QoS 方案处理延时

在松耦合的 QoS 方案中,移动节点位置的变化引起路由信息的更新,然后引发 RSVP 的本地修补机制。它的处理延时可以用如下公式进行衡量:

$$D_{rsvp} = T_w + D_{path} + D_{resv} \quad (2)$$

其中 T_w 是在 PATH 消息发出前需要等待的时间,在此方案中等于发送 BU 和 BA 信令的延迟的延时 $D_{bu} + D_{ba}$, D_{path} 和 D_{resv} 分别是在新路径上传输和处理 PATH 消息 RESV 消息的延时,所以,它的处理延时可以表示为:

$$D_{rsvp} = D_{bu} + D_{ba} + D_{path} + D_{resv} \quad (3)$$

5.3 本文提出的紧耦合的 QoS 方案处理延时

移动节点采用 BU/BA 与 QoS 合并的方案进行资源预留的时间为:

$$D_{qos} = D_{bu+qos} + D_{ba+qos} \quad (4)$$

其中 D_{bu+qos} 是在新路径上传输和处理 BU+QoS 信令的时间, D_{ba+qos} 是传输和处理 BA+QoS 信令的时间。因为 BU/BA 信令是端到端的信令, QoS 消息是其中的 hop-by-hop 选项,和 PATH/RESV 信令一样在每个中间路由器中处理,所以 D_{bu+qos}/D_{ba+qos} 近似等于 D_{path} 和 D_{resv} , (4) 式可以写为:

$$D_{qos} = D_{path} + D_{resv} \quad (5)$$

可以看出, (5) 式比 (1) 式减少了 RSVP 两次刷新信令之间的时延,比 (3) 式减少了链路传送 BU 和 BA 信令的时延,当节点是域间移动时,信令经过的链路长,延迟大,所以中间路由器的处理时间相对可以忽略不计,这时 (5) 式比 (3) 式相比减少了近一半的时间。

结束语 本文提出的 QoS 信令解决方案使移动节点在切换后可以快速地建立资源预留,它的实际应用将极大地改善移动环境中的服务质量。将来,我们要对 RSVP 信令结构和 QoS 参数做进一步的加强和完善并考虑和 DiffServ 的结合应用。

参考文献

- 1 Braden R, et al. Integrated Services in the Internet Architecture: an Overview. IETF, RFC1633, June 1994
- 2 Braden R, Zhang L, Berson S, et al. Resource ReSerVation Protocol (RSVP)—— Version 1 Functional Specification, 2205, 1997. Internet Draft, Oct. 2002
- 3 Burness L, Hepworth E, López A, Manner J. Architecture for Providing QoS in an IP-based Mobile Network. IST Mobile Summit 2001, Sept. Barcelona, Spain
- 4 Soliman H, Ericsson, Castelluccia C. INRIA Hierarchical Mobile IPv6 mobility management (HMIPv6)
- 5 Karsten M. Experimental Extensions to RSVP-Remote Client and One-Pass Signalling. IWQoS, 2001. 269~274
- 6 Fu X, Kappler C. Towards RSVP Lite: Light-weight RSVP for Generic Signaling. In: Proc. of the 17th Intl. Conf. on Advanced Information Networking and Applications (AINA) 2003, IEEE Computer Society Press, Xi'an, China, March 2003. 619~622
- 7 Chaskar H, Koodli R. A Framework for QoS Support in Mobile IPv6. Internet Draft, work in progress, Mar. 2001
- 8 López A, Velayos H, Manner J, Villaseñor N. Reservation Based QoS Provision for Mobile Environments. In: Proc. 1st IEEE Workshop on Services and Applications in the Wireless Public Infrastructure, Evry (Paris) France, July 2001