时钟同步的 RMD 聚集预留状态管理

黄程波[1],徐益龙[2],易桂生[3]
HUANG Cheng-bo[1],XU Yi-long[2],YI Gui-sheng[3]

1.深圳信息职业技术学院 信息技术研究所,广东 深圳 518029
2.深圳大学 信息工程学院,广东 深圳 518060
3.江西师范大学 数学与信息科学学院,南昌 330022
1.Shenzhen Institute of Information Technology,Shenzhen,Guangdong 518029,China
2.College of Information Engineering,Shenzhen University,Shenzhen,Guangdong 518060,China
3.College of Mathematics and Information Science,Jiangxi Normal University,Nanchang 330022,China

HUANG Cheng-bo,XU Yi-long,YI Gui-sheng.Management of aggregated reservation states for RMD based on clock synchronization.Computer Engineering and Applications,2009,45(12):78-80.

Abstract: Resource Management in Differentiated Services (RMD) is a simple,scalable signaling method for resource reservation within a Differentiated Services domain.However,the algorithms used in RMD cannot accurately maintain the aggregated reservation state.This paper presents a clock synchronization paradigm and the proposed implementation architecture within the router.The simulation result demonstrates that the approach can control the aggregated reservation state accurately.
Key words: Quality of Services;clock synchronization;Differentiated Services(DiffServ);aggregated reservation state

摘 要: 区分服务网络资源管理方案 RMD(Resource Management in Differentiated Services)基于分布式信令,简单、可扩展。分析了RMD 中的聚集预留状态管理算法,指出其无法保证资源状态的准确性,从而对聚集流的服务质量产生不利影响。针对 RMD 存在的问题,提出基于时钟同步的聚集预留状态管理方案,该方案包括时钟同步机制和路由器内的实现方法。仿真实验证明该方案可以保证聚集预留状态的准确性。
关键词: 服务质量;时钟同步;区分服务(DiffServ);聚集预留状态
DOI: 10.3778/j.issn.1002-8331.2009.12.026 **文章编号:** 1002-8331(2009)12-0078-03 **文献标识码:** A **中图分类号:** TP393

1 引言

为了能够在 Internet 上提供端到端的服务质量保证，需要端到端的 QoS 信令协议进行显式和动态的接纳控制和资源预留，该协议能够在异构网络环境中提供 QoS 信令协议支持。RSVP 协议能够作为端到端的 QoS 信令协议,然而,由于 RSVP 协议的复杂性,RSVP 未能广泛采用。另外,其他的服务质量体系结构(如区分服务 DiffServ)的出现,也需要 QoS 信令协议更具有通用性。IETF 最近成立的下一代信令工作组 NSIS(Next Step In Signaling)的主要工作之一就是标准化 QoS 信令协议。RMD(Resource Management in DiffServ)[1-3]作为区分服务的一种 QoS 信令模型,目前正在被 NSIS 标准化[4]。RMD 将一种分布式的基于显式信令的方法引入到区分服务网络,扩展了区分服务的体系结构。但是,RMD 的聚集预留状态管理算法无法保证资源状态的准确性,接入的流的资源预留总和会超过节点的可用资源,从而影响了 QoS,产生 QoS 违例(QoS Violation)[3]现象。针对 RMD 存在的问题,提出基于时钟同步的聚集预留状态管理方案。

2 RMD 聚集预留状态管理算法

在 RMD 协议模型中,所有网络节点(包括边缘节点)为每个聚集类维护两个聚集预留状态变量:*last* 和 *count*。每个区分服务网络中有一个聚集预留状态刷新周期 *T*。RMD 中的网络资源被分解为若干资源单位。每个网络节点的每个聚集类都有一个可以静态配置的表示最大可预留资源单位数目的阈值。

当 Ingress 节点收到一个外部 QoS 请求时，首先计算该请求的资源单位数。如果 Ingress 节点能够满足资源请求,Ingress 节点生成本地资源请求消息和本地资源刷新消息。资源请求消息可以包含多个资源单位,而每个资源刷新消息只包含一个资源单位。对于包含 *BW* 个资源单位的资源请求消息到达 Ingress 节点时,如果该节点能够满足资源请求,则生成 *BW* 个资源刷新消息,这些刷新消息由 Ingress 周期性地发送,发送时间间隔为 *T*/*BW*。例如在图 1 中,消息 0 表示一个 *BW*=5 的资源请求,消息 1,2,…表示相应的周期性发送的刷新消息。当外部资源释放消息到达 Ingress 节点时,Ingress 节点将生成本地显式资源释放消息,该消息可以包含多个资源单位。

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60672150);广东省自然科学基金(the Natural Science Foundation of Guangdong Province of China under Grant No.7008733)。
作者简介: 黄程波(1968-),男,博士,副教授,研究方向:计算机网络。
收稿日期: 2008-12-17 **修回日期:** 2009-02-25

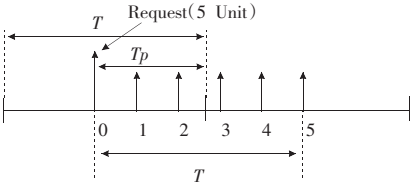


图 1　连接建立时的过渡现象

网络节点状态变量 *last* 和 *count* 由本地资源请求消息，资源刷新消息和资源释放消息负责维护。当包含 *BW* 个资源单位的资源请求消息到达网络节点时，如果 *last*+*BW* 没有超出资源阈值，则为该请求预留资源，*last*=*last*+*BW*，*count*=*count*+*BW*；当一个刷新消息到达网络节点时，只有 *count* 变量加 1；当包含 *BW* 个资源单位的资源释放消息到达时，*last*=*last*−*BW*，*count*=*count*−*BW*。刷新周期 *T* 时间到时，*count* 的值被复制到 *last*，然后 *count*=0。

根据以上算法，在图 1 中，对于资源请求 Request 来说，在其资源请求到达的刷新周期中，资源被过度分配了 2 个资源单位。因此，在资源请求消息到达时，*count* 变量的更新采用近似算法：$count=count+\lceil BW(1-Tp/T)\rceil$，其中 *Tp* 如图 1 所示。类似的过渡现象也出现在连接结束，处理资源释放消息时。此时，*count* 变量的更新也采用了类似的近似算法。近似算法的使用在一定程度上影响了 *count* 变量的准确性。

理想情况下，周期性的刷新消息总是在同一时刻到达同一个网络节点，从而保证 *count* 变量的准确更新。但网络时延抖动使得刷新消息到达同一个网络节点的时间发生偏移，影响了 *count* 变量的准确更新，状态变量的准确性和稳定性受到了相应的影响。当刷新消息仅包含一个资源单位时，时延抖动对状态变量的影响被削弱了。这是刷新消息仅包含一个资源单位的主要原因。

资源单位的大小难以定义，大量的刷新消息增加了额外的信令开销，过渡现象的存在，网络时延抖动等使得状态变量的准确性和稳定性得不到保证，不可避免产生 QoS 违例现象[3]，最终会对接纳控制和资源预留结果产生不利影响，难以保证聚集流的服务质量。

3　基于时钟同步的聚集预留状态管理

本文的一个基本假设是网络时钟必须同步。尽管网络时钟同步仍然是一个还在研究之中的问题，但现有的技术例如 GPS (Global Position System)，SDH(Synchronized Digital Hierarchy)和 SONET(Synchronized Optical Network)可以将网络时钟的同步精确到 10^{-7} s。对于覆盖范围较小的网络，甚至 NTP (Network Time Protocol)也可满足本方案的时间精度要求。在此，不详细讨论有关网络时种同步的问题。

3.1　时钟同步机制

为了便于描述，使用图 2 所示的 DiffServ 域参考模型，而图 3 是本文提出的时钟同步机制的示意图。如图 3 所示，一个 DiffServ 域内的所有节点具有相同的刷新周期 *T*，*T* 被分成若干个长度相等的时钟单位 T_c，所有节点的 T_c 在时间上是同步的。网络不要求节点的刷新周期的启动时间同步，但必须在某个 T_c 时钟的开始或到达时启动。当节点启动刷新周期时，同时也启动一个长度为 T_c 的时钟，每当 T_c 时间到达时，节点将 T_c 时间内到达的所有消息连续地发送出去，并且这些消息必须在 T_c 时间内到达下一个节点。例如，在时刻 t_1 到达 E_1 的消息 req_1 必

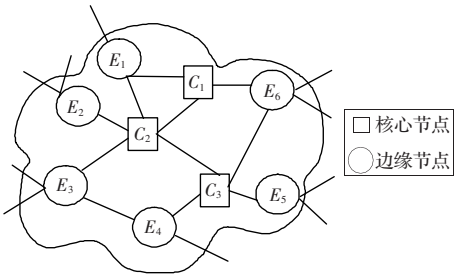


图 2　Diffserv 域参考模型

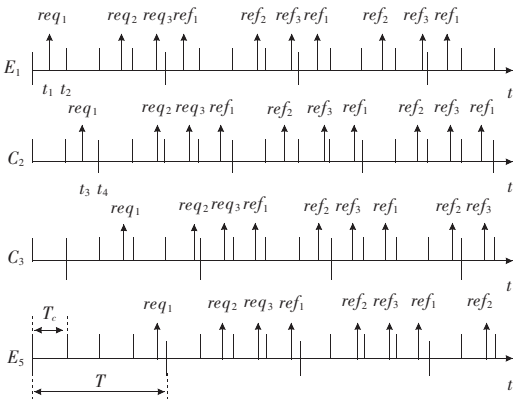


图 3　时钟同步示意图

注：ref_1、ref_2 和 ref_3 分别表示对应于资源请求消息 req_1、req_2 和 req_3 的资源刷新消息；E_1、C_2、C_3、E_5 是图 2 中通信路径 $E_1\rightarrow C_2\rightarrow C_3\rightarrow E_5$ 的节点。

须在时刻 t_2 到达时发送，然后 req_1 在时刻 t_3 到达 C_2，且必须在时刻 t_4 发送出去。在入口节点，请求消息和刷新消息同时产生。状态变量 *count* 和 *last* 的操作与 RMD 中一致，但本文的方案中不需要将资源分解为资源单位，刷新消息和请求消息包含的资源数量是相同的。基于以上时钟同步机制，可以保证同一个连接的刷新消息总是在同一个刷新周期内的同一个 T_c 内到达同一个节点，不存在 RMD 的过渡现象，不需要使用资源单位，避免了资源单位大小难以定义问题，也不会由于大量的刷新消息增加额外的信令开销。

为了实现以上时钟同步机制，路由器必须有相应的实现结构以控制信令消息的时延，下面讨论路由器内的实现。

3.2　路由器实现

为了控制信令消息的时延，在路由器内必须将信令消息分组和数据分组的数据路径分开，其实现方法如图 4 所示。路由器的每一个输出端口为信令消息分组设置一个独立的队列，该

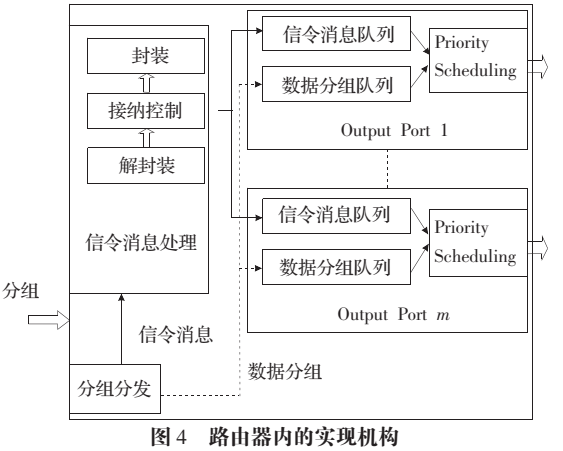


图 4　路由器内的实现机构

队列和数据分组队列之间采用 PS(Priority Scheduling)调度算法,信令消息分组具有最高优先级,当 T_c 时刻到达时,信令消息分组立刻被连续地发送出去。因此,信令消息与网络流量完全隔离了,不会受到网络时延抖动的影响。另外,为了减少信令消息的数量,还可以将多个消息封装在一起,因此,在路由器内需要对消息进行封装和解封装。

下面讨论如何适当定义参数 T_c。

如图 5 所示,信令消息经历的时延 d_{nodal} 可分为发送时延、排队时延、传输时延和处理时延,分别记为 d_{trans}、d_{queue}、d_{prop} 和 d_{proc}:

$$d_{nodal}=t_s+d_{proc}=d_{trans}+d_{queue}+d_{prop}+d_{proc} \quad (1)$$

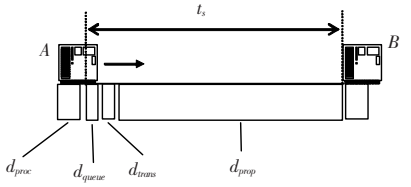


图 5 分组经历的时延示意图

考虑到区分服务域的各节点的分组处理能力各不相同,定义 T_c 如下:

$$T_c \geqslant \mathrm{MAX}(d_{trans}+d_{queue}+d_{prop}+d_{proc})= L_{max}/R_{min}+M_{max}/R_{min}+D_{max}/S_{min}+k(N_{max}/C_{min}) \quad (2)$$

其中 L_{max} 是分组的最大长度,R_{min} 是最小链路带宽,M_{max} 是信令消息的最大突发量,D_{max} 是节点间的最大距离,S_{min} 是链路的最小传输速度,N_{max} 是最大的需要连续发送的消息分组数量,C_{min} 是节点的最小处理能力(MPPS)。由于处理消息分组时不需要复杂的算法,可以将消息分组的处理时延定义为 k 倍的数据分组处理时间。式(2)中对 T_c 的计算采用了最保守的计算方法,足以保证信令消息在 T_c 内到达下一个节点。

4 仿真实验结果

采用 NS2[5]进行实验仿真。仿真实验目的是验证本方案能否保证聚集预留状态的准确性。仿真拓扑如图 6 所示,链路的传输时延标识在图中,所有链路的带宽都是 100 Mb/s。刷新周期 T 为 10 s,T_c 为 10 ms,连接请求的带宽为 64 Kb/s。每个资源请求成功的连接的持续时间服从指数分布,均值为 600 s。图 7(a)和(b)分别显示了连接请求到达的时间间隔服从均值为 0.3 s 和 0.5 s 的指数分布时的实验结果,可以看出,本方案(SRMD)能够保证聚集预留状态的准确性,而 RMD 的聚集预留状态出现明显的振荡,并且预留资源会超出最大可用带宽,出现过度预留。

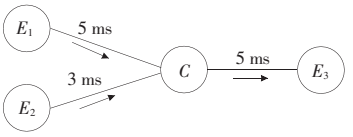
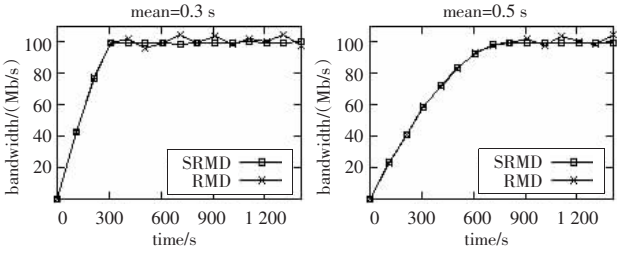


图 6 仿真拓扑



(a)均值为 0.3 s (b)均值为 0.5 s

图 7 仿真实验结果

5 结语

本文在网络时钟同步的基础上,提出了聚集预留状态的管理方案。该方案克服了网络时延抖动的影响,不存在 RMD 中的过渡现象,不需要使用资源单位,避免了资源单位大小难以定义问题,也不会由于大量的刷新消息增加额外的信令开销。网络仿真实验证明该方案能够保证状态变量的准确性。在下一步的工作中,将考虑如何将该方案集成到 RMD 中,并对其性能进行评估。

参考文献:

[1] Westberg L,Császár A,Karagiannis G,et al.Resource Management in Diffserv(RMD):A functionality and performance behavior overview[C]//LNCS 2334:Proceedings of PfHSN'2002,Seventh International Workshop on Protocols for High-Speed Networks,Berlin,Germany,2002:17-34.

[2] Karagiannis G,Báder A,Pongrácz G,et al.RMD-a lightweight application of NSIS[C]//Proceedings of the 11th International Telecommunications Network Strategy and Planning Symposium,Vienna,Austria,2004:211-216.

[3] Marquetant A,Pop O,Szabo R,et al.Novel enhancements to load control a soft-state,lightweight admission control protocol[C]//Proceedings of the 2nd International Workshop on Quality of Future Internet Services,Coimbra,Portugal,Sept 24-26,2001:82-96.

[4] Bader A,Westberg L,Karagiannis G,et al.RMD-QOSM-the resource management in Diffserv QoS model.draft-ietf-nsis-rmd-13.txt,Work in Progress,July 14,2008.

[5] The Network Simulator NS2.http://www.isi.edu /nsnam/ns/.

(上接 77 页)

参考文献:

[1] Shamir A.Identity-based cryptosystems and signature schemes[C]//LNCS 196:Advances in Cryptology-Crypto 1984.[S.l.]:Springer,1984:48-53.

[2] Boneh D,Franklin M.Identity based encryption from the Weil pairing[C]//LNCS 2139:Advances in Cryptology-Crypto 2001.[S.l.]:Springer,2001:213-229.

[3] Bellare M,Rogaway P.Random oracles are practical:A paradigm for designing efficient protocols[C]//Proceedings of the First ACM Conference on Computer and Communication Security,1993:62-73.

[4] Bellare M,Boldyreva A,Palacio A.An uninstantiable random oracle model scheme for a hybrid-encryption problem[C]//LNCS 3027:Advances in Cryptology Eurocrypt 2004.[S.l.]:Springer,2004:171-188.

[5] Nielsen J B.Separating random oracle proofs from complexity theoretic proofs:The noncommitting encryption case[C]//LNCS 2442:Advances in Cryptology-Crypto 2002.[S.l.]:Springer,2002:111-126.

[6] Boneh D,Boyen X.Efficient selective-ID secure identity based encryption without random oracles[C]//LNCS 3027:Advances in Cryptology-Eurocrypt 2004.[S.l.]:Springer,2004:223-238.