

文章编号:1001-9081(2006)07-1532-04

带内信令系统实现 Ad Hoc 网络的邻居预留

舒炎泰,董林芳

(天津大学 电子信息工程学院, 天津 300072)

(donglinfang@163.com)

摘 要:提出使用带内信令系统实现跨层的邻居预留机制(NR-INSIGNIA),该机制在进行带宽预留时考虑 MAC 层的信道竞争。节点为某个 QoS 流预留一定带宽之后,在所有的邻居节点上同时预留等量的带宽,以此来消除信道竞争对带宽预留产生的不利影响。仿真试验对比了没有邻居预留机制的原始 INSIGNIA (In-band signaling system) 系统与 NR-INSIGNIA 系统的性能,结果表明 NR-INSIGNIA 系统可以显著提高 QoS 流的平均预留包比率、投递率,减少端-端延迟。

关键词: Ad Hoc 网络; 服务质量(QoS); 带宽预留; 邻居预留

中图分类号: TP393.04 文献标识码: A

Neighborhood reservation with in-band signaling system in Ad Hoc networks

SHU Yan-tai, DONG Lin-fang

(School of Electronic & Information Engineering, Tianjin University, Tianjin 300072, China)

Abstract: A cross-layer neighborhood reservation mechanism with in-band signaling system (called NR-INSIGNIA) for Ad Hoc networks was proposed, considering the impact of channel contention at MAC layer. When a node reserved bandwidth for a QoS flow, corresponding bandwidth reservation would be reserved at all its neighboring nodes simultaneously to eliminate the negative impact on bandwidth reservation caused by channel contention. This cross-layer neighborhood reservation mechanism can provide a certain guarantee for the QoS bandwidth reservation. Compared with the original INSIGNIA without neighborhood reservation, simulation results show that NR-INSIGNIA system can significantly improve the reserved packet ratio, packet delivery ratio and end-to-end delay of QoS flows.

Key words: Ad Hoc networks; quality of service (QoS); bandwidth reservation; neighborhood reservation

0 引言

如何在 Ad Hoc 网络中保证服务质量变得越来越重要。资源预留是保证 QoS 的一个重要手段。带外信令系统(如 RSVP^[1]等)建立预留连接所需要的资源开销过大,建立连接所需的时间也较长,不能对网络拓扑变化作出快速反应,不适合拓扑频繁变化的 Ad Hoc 网络。INSIGNIA^[2]是一种带内信令系统,将控制信息加载到普通数据包中,响应时间短,开销很小,适应 Ad Hoc 网络拓扑变化频繁的特点和带宽受限的环境。BruIT^[3]是一种分布式协议,在预留带宽时考虑其他传输的干扰,通过周期性地发送广播包来估计可用带宽,然而, BruIT 没有提供有效的方法对已经预留的带宽提供保证。ROC^[4]主要解决网络中 QoS 业务流因为资源预留而引起冲突的问题,是流一级的控制。但是不能解决因链路层信道竞争而引起的问题,而且当节点随机地频繁移动时,ROC 也不能提供有效的解决方案。文献[5]证明确实存在 MAC 层信道竞争对带宽预留的影响问题,提出了邻居预留(Neighborhood Reservation, NR)机制,其思想是:当一个节点为一个 QoS 业务流预留带宽时,该节点在 MAC 层向所有邻居节点广播预留带宽信息,收到该信息的所有邻居节点,在 MAC 层对流经本节点的 Best Effort 业务流的发送速率进行控制,减少与预留带宽的节点的信道竞争,以保证所预留的带宽。

本文提出邻居预留带内信令系统 NR-INSIGNIA, 目的在

于把适应 Ad Hoc 网络拓扑变化特点的带内信令系统 INSIGNIA, 与能够为预留带宽提供保证机制邻居预留结合起来, 即把 INSIGNIA 扩展到 MAC 层, 实现一种既适应 Ad Hoc 网络拓扑变化, 又能有效解决 MAC 层竞争信道对带宽预留的影响, 能够提供有保证的带宽预留的跨层信令系统。

1 NR-INSIGNIA 的框架结构及协议命令

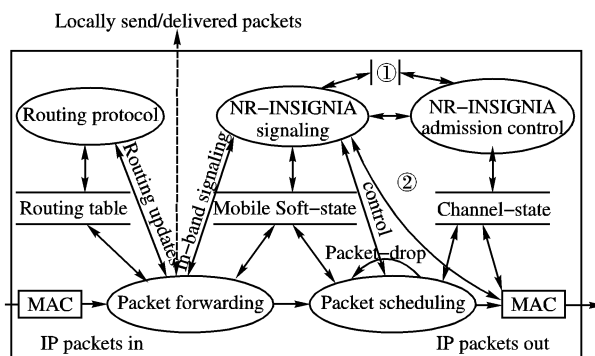


图1 NR-INSIGNIA 框架结构

NR-INSIGNIA 的设计以 INSIGNIA 为基础。在基本框架保持不变的情况下,对其中的几个功能模块做了改动,并根据需要增加了信令模块与 MAC 之间的耦合。如图 1 所示。

Packet Forwarding 模块 该模块负责将到来的包进行分类,然后把它们转发到相应的模块分别处理。信令包被转发

收稿日期:2006-01-05;修订日期:2006-02-28 基金项目:国家自然科学基金资助项目(60472078,90604013)

作者简介:舒炎泰(1942-),男,江西于都人,教授,博士生导师,主要研究方向:计算机网络性能评价、CIMS、计算机实时控制;董林芳(1972-),女,河北张家口人,博士研究生,主要研究方向:无线网络的性能评价、网络建模。

到 INSIGNIA 信令模块;一般的包或者被本地接收,或者转发到调度模块,发送到下一跳节点。

Routing 模块 即一般的路由协议,负责跟踪网络路由的变化。INSIGNIA QoS 框架的设计思想是将该信令模块独立于路由协议,从而适用于各种不同的 Ad Hoc 网络路由协议。

Packet Scheduling 模块 负责根据信道当前状况对包进行调度,INSIGNIA 中使用加权轮询调度算法,NR-INSIGNIA 仍使用该调度算法。

NR-INSIGNIA signaling 模块 该模块取代了原来的 INSIGNIA signaling 模块,它除了负责 QoS 业务流的建立、恢复、调整、释放外,还要负责向邻居节点发送邻居预留通知信息和邻居预留释放信息。NR-INSIGNIA 采用的仍然是带内信令的方法,将控制信息放在 IP 包的头部,这样流就能够被快速地建立、恢复、调整或释放。

NR-INSIGNIA admission control 模块 该模块负责根据业务的带宽请求为各个流分配带宽,在成功获取预留带宽之后,每个流定时地进行预留更新,这由 Soft-state 机制完成。该模块通过测量现有带宽的使用情况,然后根据带宽的申请决定是否准入。准入算法对邻居节点的带宽预留情况给予了充分考虑。

MAC 模块 在 INSIGNIA 中,该模块只是执行介质访问控制功能,而在 NR-INSIGNIA 中,增加了与信令模块的协作,能够根据 NR-INSIGNIA signaling 模块的通知信息做出相应的处理。例如邻居节点收到邻居预留通知信息后,将在 MAC 层控制 Best Effort 包的发送速率,以减少信道竞争。

与 INSIGNIA 的框架结构^[2]相比,NR-INSIGNIA 系统中增加了邻居预留信息表(图1中的①)以及 NR-INSIGNIA signaling 模块与 MAC 模块之间的耦合(图1中②)。

NR-INSIGNIA 系统的协议命令有两类:继承自 INSIGNIA 的协议命令和 NR-INSIGNIA 自身的协议命令。继承自 INSIGNIA 的协议命令有:服务模式、有效载荷类型、带宽指示和带宽请求字段。

为了支持邻居预留,NR-INSIGNIA 引入了以下新的协议命令:

1) NR-INSIGNIA 带宽指示 (bandwidth indicator)。NR-INSIGNIA 的作用在于将本节点上某 QoS 流的资源预留量通知给所有的邻居节点,同时各个邻居节点根据这个指示为该流预留相应的资源。NR-INSIGNIA 带宽指示的设置有两种情况:非零值和零值。前者用来通知邻居节点为该流预留或者更新带宽资源,而零值则用来通知邻居节点释放曾经为此流预留的资源。

2) Next hop。下一跳指示让邻居节点判断自己是否处于该流所使用的路径上。邻居节点将 next hop 字段与自己的地址相比较,如果相同则说明本节点位于该流所使用的路径上,因此不执行邻居预留操作,而应当被视为下一跳节点做正常处理;反之,说明本节点的确属于邻居节点,应该进行邻居预留的相关操作。

3) QoS flow ID。邻居节点通过分析 QoS 流的 ID,判断在本节点是否已经为该流预留了带宽。如果在一个节点的预留信息表中有该流的记录,说明该节点是该 QoS 流路径上的上一跳节点,不执行任何操作。如果在一个节点的邻居预留信息表中有该流的记录,说明该节点已经为该流做了邻居预留,则根据 NR-INSIGNIA 带宽指示执行更新或释放操作。如果一个节点的预留信息表和邻居预留信息表中都没有该流的记

录,而这个节点又不是该流的下一跳节点,则这个节点应为该流建立邻居预留。

与协议命令类似,协议操作也分为两类:继承自 INSIGNIA 的协议操作(预留建立、QoS 报告等)和 NR-INSIGNIA 自身的协议操作(邻居预留通知、邻居预留等)。

邻居预留通知 在一个流的预留建立过程中,每当中间节点成功地为该流预留了资源并建立了状态信息之后,NR-INSIGNIA 就会发出邻居预留通知 NRN 广播信息。收到 NRN 的邻居节点都会根据此通知中携带的信息做邻居预留处理,包括邻居预留的建立、更新和释放等。

邻居预留的软状态管理 邻居节点在收到 NRN 后会为 QoS 流做邻居预留,并建立相应的状态信息。这一状态信息同样需要更新、撤销。NR-INSIGNIA 采用的是与 INSIGNIA 类似的软状态管理机制。NR-INSIGNIA 的更新与 INSIGNIA 的预留信息更新是同步的。而邻居预留的释放则有两种方式:显式释放(通过邻居预留通知释放)和定时器超时释放。邻居节点设置了邻居预留信息有效期,如果在该有效期内没有收到新的 NRN,那么该预留信息就会被清除,相应的邻居预留被释放。

2 NR-INSIGNIA 的系统流程

包进入 NR-INSIGNIA 信令系统后,根据包的源、目的地址以及包的类型来判断该包的去向,然后分别交由源端、中间节点、邻居节点、目的端单独处理后,或释放,或交由路由层继续处理(一般的数据包交由下层处理)。

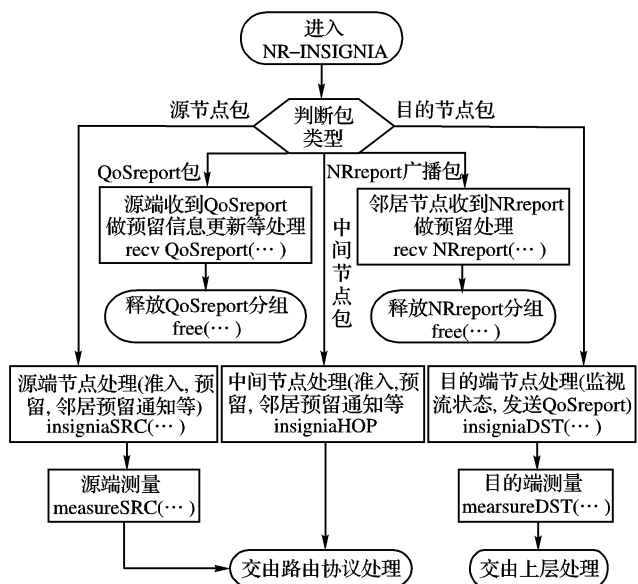


图2 NR-INSIGNIA 系统流程

源端 首先判断要发送的包是否属于 QoS 流,如果不是,说明该包属于 Best Effort 流,立即返回并经过源端测量后直接交由路由协议处理。否则,说明该包属于 QoS 流,那么就查找源端的预留信息表,如果已经为该业务流预留了资源,就根据流状态做相应的调整处理;否则,源端便会调用准入控制模块为此包所属的 QoS 业务流预留资源。如果有足够的带宽资源可用,就为该 QoS 流预留资源并建立一个新的表项,同时向所有邻居节点广播发送一个邻居预留通知 NRreport (1),该通知信息中含有带宽请求量、下一跳地址、流 ID 等信息。完成资源预留后,包经过测量后转入路由模块继续处理。

目的端 首先判断所接收的包是否为 QoS 流的数据包,如果不是就直接返回并经由目的端测量后交由上层继续处理。

否则就查找目的端的资源预留信息表,如果已经为该包所属的 QoS 流预留了资源,那么就根据该包所携带的信息更新预留信息表并向源端发送 QoSreport;如果还没有为此 QoS 流预留过资源,那么就调用准入控制模块重新预留,并向源端发送 QoSreport。完成后返回,再经由目的端测量后,交由上层处理。

中间节点 中间节点是指 QoS 流所经过的路径上的每一个节点。

包进入中间节点后,首先判断它是否属于 QoS 流的数据包(RES 包),如果不是(说明该包为 BE 包),那么立即退出;否则,根据该包在预留信息表中查找是否存在该包所属 QoS 流的表项。

如果不存在该包所属 QoS 流的表项,说明节点需要为该 QoS 流预留资源。如果有足够的带宽资源可用,就为该 QoS 流预留资源并建立一个新的表项,同时向所有邻居节点广播发送一个邻居预留通知,该通知信息中含有带宽请求量、下一跳地址、流 ID 等信息;如果没有足够的资源可供使用,那么就将此包变为 BE 包,同时向所有邻居节点广播发送一个邻居通知,使邻居节点及时释放为此流所预留的资源。

如果存在该包所属 QoS 流的表项,说明此节点已经为该 QoS 流预留了资源。这时就根据包中携带的信息更新预留表项中的流状态信息,然后再根据本节点的状态,更新包中的带宽指示字段,以便目的端可以实时地检测流状态。

除此之外,当中间节点中为该 QoS 流所预留的资源因为节点移动等原因需要释放时,也要相应地发送邻居预留通知 NRreport(2),以通知邻居节点释放相应的资源。

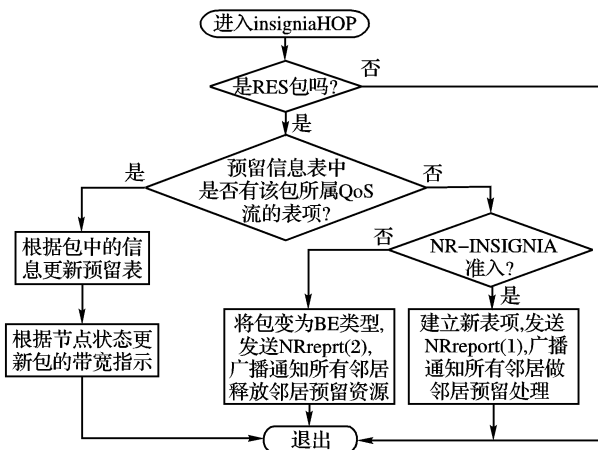


图3 中间节点的信令操作流程

邻居节点 进入邻居节点信令操作(recv_NRreport)的包只可能是邻居预留通知,所以进入邻居节点后,不必再判断它是否为 BE 包,而是直接进入邻居预留操作。首先,用邻居预留通知中的下一跳信息同本节点的地址相比较,看此节点是否为该 QoS 流的下一跳节点。如果是,那么就丢弃此通知包,立即退出并且不做邻居预留。因为当本节点作为 QoS 流的下一跳时,会采用一般的信令操作为流进行资源预留,并建立相应的预留信息表项。如果该节点不是 QoS 流的下一跳,那么还要在预留信息表中查找是否存在该 QoS 流的表项,如果存在那么说明本节点是该流所使用路径上位于本节点前的某一个节点,已经为此流预留过资源,不必做邻居预留处理,这时就丢弃该通知包并立即退出;否则,说明此节点正是需要做邻居预留处理的邻居节点,并继续进行下面的操作。根据通知包中的流 ID 信息在邻居预留表中查找,看是否存在该流的

表项。如果没有,那么就为其预留资源,并通知 MAC 层启动 NR-INSIGNIA 包调度算法对 BE 包进行适当的丢包处理以保证预留带宽,同时建立相应的邻居预留表项,然后退出。如果有,就说明已经为该流做了邻居预留,那么就需要继续判断该通知包中的带宽指示字段是否大于零。如果大于零,也就是说收到 NRreport(1)包,这说明该流正处于生命期,此通知包是用来更新邻居预留表项的,那么,根据此带宽指示字段更新邻居预留表项中的相应字段,调整邻居预留资源,更新表项计时字段,然后退出。否则如果等于零,也就是说收到 NRreport(2)包,这说明此通知包用来通知所有邻居节点,该流已经结束或降级,相应的邻居需要释放为其预留的资源,同时通知 MAC 层撤销 NR-INSIGNIA 包调度算法,并删除相应的邻居预留表项,然后退出。最后,在查找邻居预留表的过程中,需要检查每一项的计时字段,看其是否过期。如果已经过期,则删除表项,并释放其预留的资源。

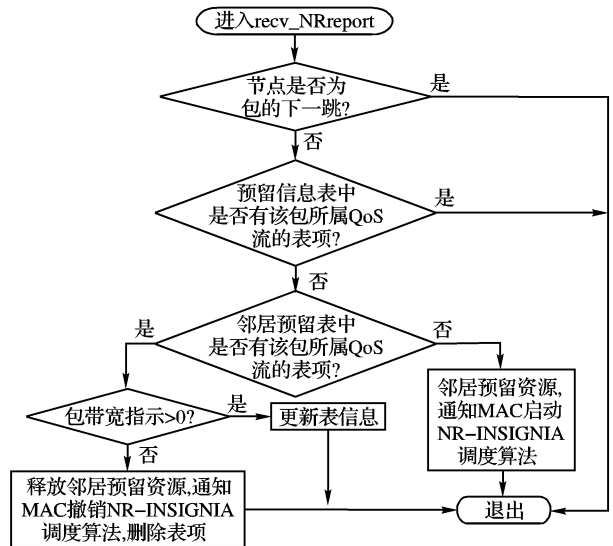


图4 邻居节点的信令操作流程

3 仿真实验及性能评价

为了验证 NR-INSIGNIA 的性能,本文用网络仿真软件 NS-2 做了相关的仿真实验。NR-INSIGNIA 仿真实验代码是在 NS-2.1b3、CMU 的无线扩展以及 INSIGNIA 仿真代码^[6]的基础上,添加了用于邻居预留的几个数据结构(如邻居预留表等)以及这些新的数据结构所对应的操作函数。

NR-INSIGNIA 中主要的数据结构有:邻居预留状态信息表、邻居预留通知包、邻居预留带宽总量、预留信息表、QoS 报告包。

性能评价尺度为:预留包比率、投递率和端-端延迟。

预留包比率 QoS 连接中获得 QoS 服务的包占总业务量的比例。这个比例越高,说明得到的 QoS 服务越多,降级的情况越少,服务质量越好。

投递率 接收端收到的包数与发送端发出的包数的比率。

端-端延迟 每个数据包的端到端延迟的均值。

仿真采用 Ad Hoc 网络动态拓扑结构,网络内的移动节点没有固定位置,可以随机移动,均设有一个最大移动速度,每个节点从一个随机选择的位置开始,随机地选取一个目的点,然后以一定的速度向该点行进,在到达目的点后,停顿一段时间,接着再向另一个随机目的点行进(即随机路点移动模型)。网络覆盖范围是 1500m × 300m,共有 50 个移动节点。链路速率为 2Mbps,节点传输范围为 250m,路由协议采用动

态源路由协议(Dynamic Source Routing, DSR)。

在节点的最大移动速度为 20m/s 情况下,分别仿真了停顿时间为 0s,30s,60s,120s 和 300s 的不同运动模式。对于每个运动模式,根据随机数种子的不同,分别做了 3 个场景的实验,然后取其平均值。仿真所采用的业务类型仍是 CBR 业

务,传送速率为 16Kbps,数据包大小为 512Byte,对应于 UDP 传输层业务,业务量为 20 个业务源,其中包括 6 个 QoS 流,14 个 BE 流。实验通过区分不同的流端口来区分 QoS 业务和 BE 业务,我们将所有 BE 业务的端口设为 0,而 QoS 业务的端口设为 1~4。每个场景的仿真时间均为 300s。

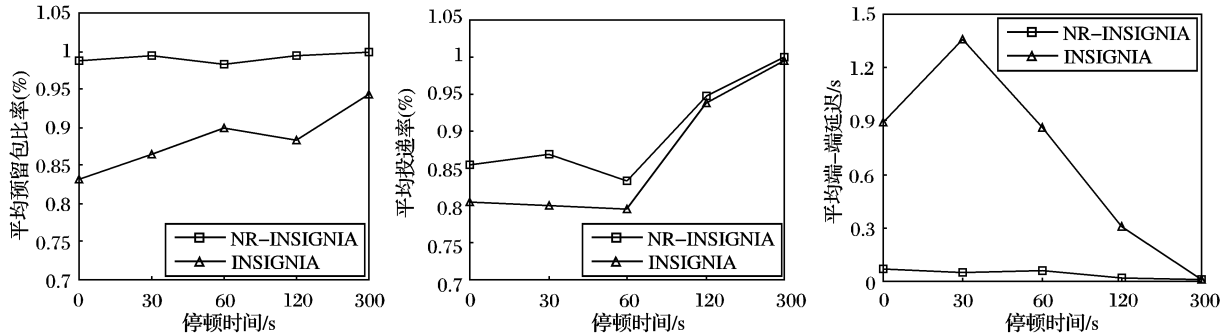


图5 QoS流在NR-INSIGNIA和INSIGNIA机制下的性能比较

图5(a)比较了节点在不同的停顿时间场景下, NR-INSIGNIA和INSIGNIA的平均预留包比率。可以看出,在任何速度时 NR-INSIGNIA的预留分组比率都明显高于INSIGNIA。随着停顿时间由300s到0s,即节点运动速度由慢变快, NR-INSIGNIA的预留分组比率相对INSIGNIA提高了5%到15%。这说明同INSIGNIA相比,在NR-INSIGNIA中, QoS流的带宽预留得到了更好的保证,而且无论节点运动速度如何,性能都非常稳定。

图5(b)比较了不同的停顿时间运动模式下, 6个QoS连接在NR-INSIGNIA和INSIGNIA中的平均投递率。可以看出,在不同停顿时间的移动场景下, NR-INSIGNIA对QoS连接而言表现都要比INSIGNIA好,其投递率平均比INSIGNIA提高了4.3%。在停顿时间比较小时,即节点运动性较强时, NR-INSIGNIA的投递率比INSIGNIA提高得相对较多。这主要是因为,在NR-INSIGNIA中, QoS流的带宽预留得到了较好的保证, QoS包被降级为Best Effort包的比率减少了,从而丢包率就降低了。

图5(c)比较了不同的停顿时间下, 6个QoS连接在NR-INSIGNIA和INSIGNIA中的平均端-端延迟。可以看出, NR-INSIGNIA的端-端延迟明显小于INSIGNIA。这是由于在NR-INSIGNIA中, QoS流的带宽预留得到有效保证, QoS包被降级为Best Effort包的比率下降,所以QoS包的端到端延迟明显下降。

图6比较了在NR-INSIGNIA和INSIGNIA机制下BE包的平均投递率和端-端延迟。图6(a)显示NR-INSIGNIA机制下BE包的投递率低于INSIGNIA机制的BE包投递率。这是因为NR-INSIGNIA机制下邻居节点需要控制BE包在MAC层的传输速度,在竞争到信道以后对BE包进行随即丢包,从而保证为QoS流预留的带宽,因此BE包的投递率降低了。图6(b)显示NR-INSIGNIA机制下BE包的平均端-端延迟小于INSIGNIA机制下BE包的平均端-端延迟。由于邻居节点控

制了BE包的传输速度,需要传输的数据包总数降低了,竞争信道的时间变短了,冲突概率也降低了。因此, NR-INSIGNIA机制下QoS流和BE流的平均端-端延迟都降低了。

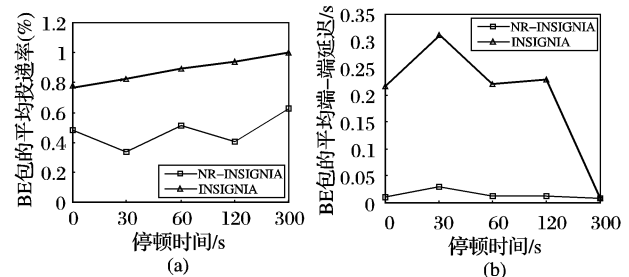


图6 BE流在NR-INSIGNIA和INSIGNIA机制下的性能比较

参考文献:

- [1] BRADEN R, ZHANG L, BERSON S, *et al.* RFC 2205, Resource reservation protocol (RSVP) - version 1 functional specification [S]. IETF, 1997.
- [2] LEE S-B, AHN G-S, ZHANG X, *et al.* Insignia: an ip-based quality of service framework for mobile ad hoc networks[J]. Journal of Parallel and Distributed Computing, 2000, 60(4): 374-406.
- [3] CHAUDET C, LASSOUS IG. BRuIT: Bandwidth Reservation under Interferences Influence[A]. Proceedings of European Wireless[C], 2002. 466-472.
- [4] YEH C-H. ROC: A Wireless MAC Protocol for Solving the Moving Terminal Problem[A]. Proceedings of IEEE International Conference on Wireless LANs and Home Networks[C], 2001. 182-189.
- [5] WANG G, SHU Y, ZHANG L, *et al.* An Efficient Resource Reservation Mechanism in Ad Hoc Networks[A]. Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering [C]. Montreal, Canada, 2003. 1655-1658.
- [6] INSIGNIA NS-2 simulation code[EB/OL]. http://comet.columbia.edu/insignia/source_code.html, 2005.

(上接第1510页)

- [15] CALVANESE D, GIACOMO GD, LENZERINI M. Answering queries using views in description logics[A]. Working notes of the KRDB Workshop[C], 1999.
- [16] IVES ZG, HALEVY AY, WELD DS. Integrating network-bound XML data[J]. IEEE Data Engineering Bulletin, 2001, 24(2).
- [17] DOAN A, HALEVY A. Efficiently ordering query plans for data

integration[A]. Proceedings of ICDE[C], 2002.

- [18] IVES ZG, FLORESCU D, FRIEDMAN M, *et al.* An adaptive query execution system for data integration[A]. SIGMOD'99 [C], 1999. 299-310.
- [19] ZHOU Y. Adaptive Distributed Query Processing [A]. Proceedings of the VLDB[C], 2003.