高技术通讯 2010 年 第 20 卷 第 6 期:577 ~ 584

doi:10.3772/j.issn.1002-0470.2010.06.005

无线自组网 TCP 定时器的分析与改进①

陈志刚② 李庆华③ 邓晓衡 黄国盛

(中南大学信息科学与工程学院 长沙 410083)

摘 要 仿真并分析了无线自组网 TCP 数据流重传超时(RTO)与传输往返时间(RTT)的关系,指出在 RTT 剧烈振荡的无线多跳网络环境下 RTO 的变化会滞后于 RTT 的变化,导致根据 RFC2988 标准做出的 RTO 估计不准确。针对这种情况,利用网络演算理论推导出了 TCP 数据流的最小延迟上界(MDUB),并以此改进了 RTO 的估计算法。仿真实验表明,基于最小延迟上界的 RTO 估计算法能准确地估计数据包的最小传输往返时间上界,减少 TCP 数据传输中的伪重传,提升无线环境下 TCP 协议的性能。

关键词 无线自组网, 802.11, TCP 定时器, 网络演算, 最小延迟上界

0 引 言

近年来,无线自组网(wireless ad hoc network)在网络研究领域受到了广泛的关注。无线自组网没有中心访问节点或者固定的基础设施,网络中的每个节点既是通信节点又是路由节点,各个节点通过分布式控制算法相互协调完成网络的通信功能[1],且能够用传输控制协议(TCP)在不可靠的网络上提供可靠的端到端的数据传输服务[2]。TCP 协议因传输可靠,现已成为互联网上的一种最为广泛配置的协议之一[3]。随着无线网络技术的发展,在无线网络上采用 TCP 协议提供可靠的数据传输服务自然成了人们的选择[4]。然而,无线多跳网络环境下 TCP 数据流的往返时间(round-trip time, RTT)值变化较大,重传超时(retransmission time-out, RTO)估计值的变化滞后于 RTT 变化,从而导致 RTO 估计不准的状况。本文利用网络演算理论并结合网络带宽测量技术推导出了 TCP 数据包传输的最小延迟上界(minimum delay up bound, MDUB),并以此改进了 RTO 的估计算法。实验表明,此改进算法能克服 RTO 变化滞后于 RTT 变化的现象,减少 TCP 数据流的伪重传,提升 TCP 协议在无线自组网环境下的效率。

1 相关研究

TCP 协议源端判断数据发送是否超时的工具就是 TCP 的重传定时器,它通过对 TCP 数据报传输往返时间(RTT)的最小延迟上界(MDUB)估计来判断当前的数据发送是否超时[5]。过低的重传超时(RTO)估计值会导致 TCP 频繁地伪重传,加重网络拥塞,降低 TCP 发送窗口大小;同时过高的 RTO 估计值会在网络发生丢包时使 TCP 发送端超时等待,降低 TCP 的效率[6]。现有 TCP 版本的重传定时器的重传超时时间估算方法是根据 RFC 2988 标准[7]来进行计算的。在 RTT 值变化较平缓的有线网络环境下,根据此标准计算的 RTO 值的变化能准确地反映出 RTT 的变化,使得 TCP 定时器在有线环境下性能较好[8]。同时也因为 RTT 值变化的平滑性,使得基于 RFC 2988 标准[11]的定时器的 TCP 数据流在有线环境下会出现不同数据流同步的问题[8]。

许多研究人员对无线网络环境下的 TCP 协议进行了研究[3,4,9,10]。如有人对无线环境下的 TCP 定时器进行了分析,指出在无线环境下 RTT 值变化较大,容易出现 TCP 伪重传,导致 TCP 协议在无线环境下性能低下,并提出了一种针对 TCP 伪重传的快速恢复方法[6,11]。文献[12]也分析了无线局域网中的伪重传问题,并提出了一种有效的 TCP 重传机制。但是目前已有的研究还没有涉及到无线环境下基于 RFC 2988 标准的 TCP 定时器本身的有效性问题。

本文对基于 802.11 协议的无线自组网 TCP 数据流的 RTT 和 RTO 指标进行了详细的仿真,指出基

① 863 计划(2008AA7034060B),国家自然科学基金(60873082)和总装预研基金(9140A15030308QT4801)资助项目。
② 男,1976 年生,博士,讲师;研究方向:无线网络,服务质量;E-mail: czg@csu.edu.cn
③ 通讯作者,E-mail: neow@163.com
(收稿日期:2009-03-23)

于RTT测量的RFC 2988标准的RTO计算方法在RTT值剧烈振荡的无线网络环境下会出现RTO值的变化滞后于RTT值变化的现象,从而导致RTO估计值过高或者过低,这种现象在TCP数据包大小不等的数据传输过程中还会加剧。为改进无线环境下TCP定时器RTO值估计不准确的情况,本文采用网络演算理论和网络带宽测量的方法推导出了TCP数据包传输的最小延迟上界并以此改进了RTO计算方法。仿真实验表明,基于最小延迟上界的RTO估计算法在无线自组网TCP数据传输过程中不会出现RTO值的变化滞后RTT值变化的现象,并且能减少TCP协议的伪重传,提升TCP协议在无线环境下的数据吞吐率。

2 无线自组网TCP定时器性能仿真与分析

RFC 2988[7]标准计算TCP重传定时器RTO的表达式为

$$RTO = SRTT + \max(G, 4RTTVAR) \quad (1)$$

其中 G 表示TCP的时针滴答,$SRTT$ 和 $RTTVAR$ 的表达式分别为

$$SRTT = (7/8)SRTT + RTT/8 \quad (2)$$

$$RTTVAR = 3/4RTTVAR + |SRTT - RTT|/4 \quad (3)$$

RTT 表示当前数据包传输往返时间,$SRTT$ 表示平滑往返时间,$RTTVAR$ 表示往返传输时间的变化值。从式(1)可以看出TCP定时器的RTO值由数据包传输往返时间RTT的平均值和变化值组成,文献[8]在有线环境下对RTO计算方法的有效性进行深入的仿真和分析,指出在有线环境下RTT值变化较平滑,没有出现较大的波动,所以根据此标准的RTO计算方法在有线环境下性能较好。而对于无线多跳网络,其TCP数据流的RTT值是否还像有线环境下那样变化平滑以及RTO值是否还能准确地根据RTT的变化而变化呢?

本文采用NS2[13]网络模拟器对TCP定时器在无线环境下的性能进行仿真,网络拓扑采用如图1所示的线形拓扑,媒体接入控制(MAC)层协议为IEEE 802.11,TCP版本为Newreno[14],有关仿真的详细参数设置请参考表1。

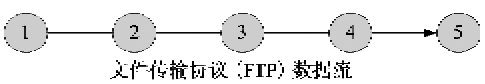

图1 一个5节点的线形拓扑无线自组网

表1 实验参数配置

参数	值	参数	值
应用	FTP数据流	时隙 T_{slot}	20μs
传输层协议	TCP-Newreno	CWmin/CWmax	32/1024
路由协议	DSDV	SIFS/DIFS	802.11默认值
MAC协议	802.11	RTS/CTS	802.11默认值
信道带宽	2Mbps	干扰距离	550m
传输距离	250m	节点距离	200m

首先在节点1和节点5之间建立一条TCP连接,在其上承载文件传输协议(FTP)数据流,FTP数据流的仿真持续时间为80s。图2所对应的实验的TCP数据包长度为500字节,图3对应的实验的TCP数据包长度为1500字节。从图2和图3我们可以看出,在无线网络环境下RTT和RTO值都有较大幅度的振荡,例如在图3中RTT值可以在0.1s内由134ms降到49ms。同时,我们也可以看到,仿真数据流的RTO曲线与RTT曲线总体上呈现出相同的变化规律,但是,RTO出现变化的时间明显滞后于RTT变化的时间约0.2~0.3s,即RTO曲线的变化滞后于RTT曲线约0.2~0.3s。

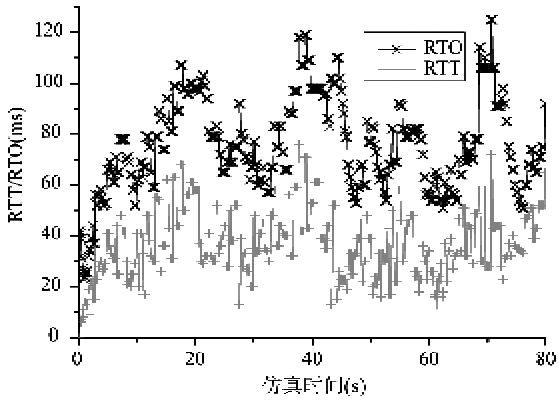

图2 TCP数据包长度为500字节的RTT/RTO值

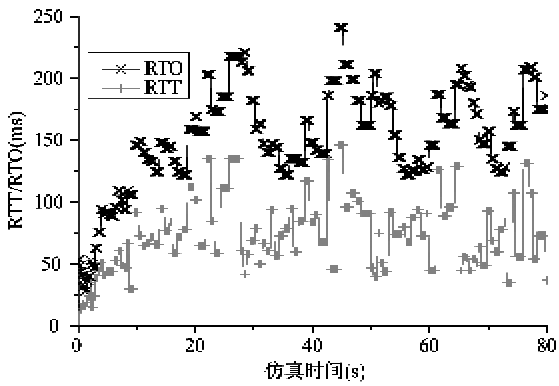

图3 TCP数据包长度为1500字节的RTT/RTO值

上述实验结论是在整个仿真过程中 TCP 数据包长度不变的情况下获得的,在实际的 TCP 数据传输中,所有 TCP 数据包的大小不可能都相等。为此,我们简单修改了 NS2 中 TCP 的实现代码,使 TCP 连接在仿真传输过程中随机产生大小为 500 字节或 1500 字节的数据包。实验还是采用图 1 所示拓扑和表 1 所示参数,其实验结果如图 4 所示。

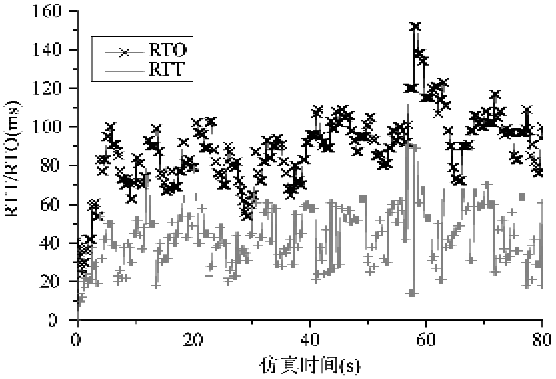

图 4　TCP 数据报长度为 500 或 1500 字节随机值的 RTT/RTO 值

从图 4 可以看出,RTO 曲线与 RTT 曲线总体上还是与图 2 和图 3 呈现相同的变化规律,即 RTT 与 RTO 波动幅度较大,RTO 曲线的变化滞后于 RTT 曲线约 0.2～0.3s。但是,在仿真的 TCP 数据传输过程中多处出现 RTO 值与 RTT 值非常接近和差距较大的情况。通过对实验的数据文件分析,得出了 RTT 与 RTO 非常接近和差距较大的具体情况(见表 2 和表 3)。

表 2　RTO/RTT 值(非常接近)

仿真时刻(s)	RTO(ms)	RTT(ms)
3.9	54	53
11.9	74	73
20	79	78
56.6	92	91
66	72	66

表 3　RTO/RTT 值(差距较大)

仿真时刻(s)	RTO(ms)	RTT(ms)
13.5	99	18
22.7	102	23
41.1	107	21
57	120	14
78.1	91	21

从表 2 和表 3 的实验数据可以看出,RTO 与 RTT 在仿真过程的某些时间非常接近,最小相差仅 1ms。同时,在一些时间点也会出现 RTO 与 RTT 值相差较悬殊的情况。如当 $t = 57$s 时,RTO 为 120ms,而 RTT 仅为 14ms,相差约 8.5 倍。

在无线的数据传输过程中,因为无线信道的共享特点,一个节点的数据发送会干扰其传输范围内的其他节点的数据传输,造成无线节点在不同发送过程中出现不同的发送概率和冲突概率,使得数据包的传输时间和传输往返时间(RTT)产生较大幅度的振荡。而对于数据包大小不一的 TCP 数据流,因为无线节点在发送大的数据分组时其发送成功概率相对于小的数据分组会急剧降低[1],进而加剧 RTT 值的振荡。而基于 RFC 2988 标准的 RTO 估算方法在计算 *RTTVAR* 时只能考虑过去的 RTT 值的变化,没有办法考虑当前数据传输的往返延迟抖动情况。所以当 RTT 值出现由小到大或者由大到小的剧烈变化时,基于 RFC 2988 标准的 TCP 定时器就会出现估计值过小或者过大,造成 TCP 的伪重传或超时等待,降低 TCP 在无线网络环境下的效率。

网络演算理论是近年来研究人员利用最小加代数和最大加代数的一系列结论,在分组交换网络领域发展并期待完善的 QoS 理论[15]。利用它可以分析端到端数据流的延迟上界、延迟抖动上界以及积压数据上界等一些分组交换网络的基本属性[15,16]。网络演算理论求解数据流端到端延迟上界的能力与 TCP 定时器对数据传输往返时间最小延迟上界估计的需求不谋而合。因此,如何通过网络演算理论求解无线数据传输回路时间最小延迟上界的估计就成了我们要关注的问题。

3　基于网络演算的无线自组网定时器改进

网络演算的主要概念包括到达曲线、服务曲线以及最小加代数下的卷积和反卷积等,数据流的端到端延迟/延迟抖动上界可由到达曲线和服务曲线的最大水平距离决定[15]。网络演算理论在网络 QoS 领域的应用已经得到了长足的发展并期待进一步的完善,如王子君等人[17]利用确定网络演算理论推导出了控制网络中的延迟确定上界,张奇智等人[18]利用网络演算求出了交换式工业以太网的最大延迟等等。

TCP 的重传定时器是用来对数据报传输往返时间(RTT)上界的最小估计,一个有效的 TCP 定时器

RTO估计算法应能根据RTT值的波动而波动[15]。然而,上节的仿真实验和分析表明,基于RFC 2988标准的RTO计算方法不能适用无线环境下RTT值的剧烈波动,造成RTO估计值不能准确反映RTT的变化。而网络演算理论可以在分组交换网络领域精确计算数据流的延迟上界,因此,我们将在这一节研究如何利用网络演算来求解TCP数据流的端到端最小延迟上界,从而为无线环境下TCP重传定时器的设计提供理论依据。

3.1 网络演算相关知识

下面给出本文在推导无线自组网TCP数据流端到端最小延迟上界过程中所需要用到的定义和定理,详细的描述读者可以参考文献[15,16]。

定义1 (广义递增函数)对于 $\forall s,t \in (R \cup \{+\infty\})$ 且 $s \leqslant t$,若有 $f(s) \leqslant f(t)$ 成立,则称 $f(t)$ 为广义递增函数。

定义2 (广义递增函数集合)若 $f(t)$ 为广义递增函数且满足

$$F = \{f(t) \mid f(t) = 0, \forall t < 0 \quad f(s) \leqslant f(t),\ t \in [0, +\infty]\}$$

则称 F 为广义递增函数集合。

定义3 (最小加卷积)对于 $\forall f,g \in F$ 函数 f 和 g 的最小加卷积运算为

$$(f \otimes g)(t) = \begin{cases} \inf\limits_{s \in [0,t]} [f(t-s) + g(s)], t \geqslant 0 \\ 0, t < 0 \end{cases}$$

其中inf()表示下确界(infimum)运算[15]。

定义4 (到达曲线)给定一个函数 $\alpha(t)$,且 $\alpha \in F, t \geqslant 0$,若输入流函数 R 满足 $R \leqslant R \otimes \alpha$,则称 α 是 R 的到达曲线。

定义5 (服务曲线)对于 $\beta \in F, \beta(0) = 0$,若满足输出流 $R^* \geqslant R \otimes \beta$,则称系统为数据流提供服务曲线 $\beta(t)$。

定理1[15] (串联节点服务曲线)系统1和系统2串联后提供的总服务曲线 β 为系统1提供的服务曲线 β_1 和系统2提供的服务曲线 β_2 的最小加卷积,即

$$\beta = \beta_1 \otimes \beta_2 \tag{4}$$

定理2[15] (延迟上界)给定一个流,进入系统时被到达曲线 $\alpha(t)$ 约束,通过系统时系统提供服务曲线 $\beta(t)$,在一个任意的时间,其延迟上界 $d(t)$ 满足式

$$d(t) \leqslant \sup_{t \geqslant 0}\{\inf\{d \geqslant 0; \alpha(t) \leqslant \beta(t+d)\}\} = h(\alpha,\beta) \tag{5}$$

其中sup()表示上确界(supermum)运算[15],$h(\alpha,\beta)$ 称作曲线 α 和 β 之间的水平偏差。

3.2 基于网络演算的TCP数据流端到端最小延迟上界求解

对基于分组交换的无线自组网,为了保证数据通信的服务质量,当一个数据流在时间 t 通过一个节点时,我们可以认为这个流服从参数为 (σ,ρ) 漏桶管制[15],其中 σ 为漏桶容量,ρ 为限制速率,也就是说对于任意的一个流 $A(t)$,存在一到达曲线 $\alpha(t) = \sigma + \rho t$ 使得 $A(t) - A(s) \leqslant \alpha(t-s)$。同时,我们也可以假定每个节点对于任意的数据流在时间 t 都提供参数为 (R,T) 的速率-延迟服务曲线[16]

$$\beta_{R,T}(t) = \begin{cases} R(t-T) & t > 0 \\ 0 & t \leqslant 0 \end{cases} \tag{6}$$

其中 R 表示节点在时刻 t 为数据流在节点 i 所提供的服务带宽;T 为数据分组在节点的服务时延。

定理3 (节点到达曲线)为保证数据流在节点 i 传输的最小延迟上界,对于服从参数为 (σ,ρ) 漏桶管制的无线自组网节点,其到达曲线可以表示为

$$\alpha_i(t) = P^t + \rho_i t \tag{7}$$

其中 P^t 为 t 时刻TCP数据流流经无线自组网中第 i 个节点的数据分组的字节数,ρ_i 表示数据流进入节点 i 的限制速率。

证明:根据网络演算理论,在第 i 个节点,我们假定节点的限制速率 $\rho_i \leqslant R_i$,基于定理1,我们可以得到公式

$$d_i(t) \leqslant \sigma_i / R_i + T_i \tag{8}$$

其中 R_i、T_i 表示节点的在 t 时刻的服务速率和延迟,在 t 时刻,它们都为定值。在分组交换网络中,σ_i 只能取数据分组的整数值,也即

$$\sigma_i^t = P^t + P^{t+\tau} + P^{t+2\tau} + \cdots + P^{t+n\tau} \tag{9}$$

其中 τ 表示分组到达的时间偏差,根据队列的先进先出(FIFO)原则,只有当 $\sigma_i^{\min} = P^t$ 时 $\sigma_i^{\min} < \sigma_i^t$,根据式(8)可得 $d_i^{\min} \leqslant d_i$,即当节点到达曲线的漏桶容量 σ 等于当前正在处理的数据分组大小时可以保证最小的延迟上界,定理3得证。

定理4 (节点服务曲线)为保证数据流在节点 i 的最小延迟上界,对于为数据流提供参数为 (R,T) 的速率-延迟服务曲线的无线自组网节点,在时刻 t 时其服务曲线为

$$\beta_{R_i^t,T_i}(t) = R_i^t(t - T_i) \tag{10}$$

其中 R_i^t 为节点在 t 时刻为数据流所提供的服务速率,T_i 表示数据流在节点 i 的服务时延。

证明:根据式(6),因为数据流在节点 i 的服务速率为 R_i^t,服务时延为 T_i,并且在实际的数据传输过程中 TCP 的运行时刻 $t>0$,将参数代入式(6)即得式(10),所以定理 4 得证。

定理 5 (TCP 流系统到达曲线)为保证数据流传输的最小延迟上界,对于服从参数为 (σ,ρ) 漏桶管制的无线自组网节点,其 TCP 数据流的到达曲线可以表示为

$$\alpha_{\text{net}}(t) = P^t + \rho_1 t \tag{11}$$

证明:对于一个流经 n 个节点的 TCP 数据流,可以将其看成如图 2 所示的一个完整的系统。我们用 $\alpha_{\text{net}}(t)$ 表示整个 TCP 数据流系统的到达曲线。

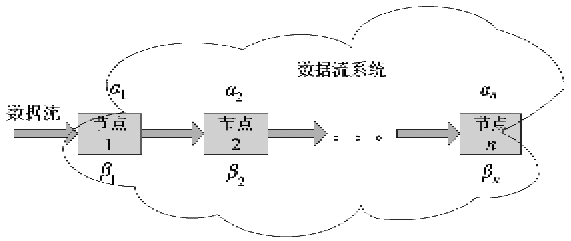

图 5 流经 n 个节点的 TCP 数据流系统

从图 5 可以看出,根据到达曲线的定义 4,TCP 数据流在进入系统时既要受到数据流系统的到达曲线 $\alpha_{\text{net}}(t)$ 的约束,同时也要受到节点 1 的到达曲线 $\alpha_1(t)$ 的约束。因此,TCP 数据流系统的到达曲线就是数据流流经的第一个节点的到达曲线,即

$$\alpha_{\text{net}}(t) = \alpha_1(t) \tag{12}$$

将 $\alpha_1(t)$ 代入式(7)即得式(11)。定理 5 得证。

定理 6 (TCP 流系统服务曲线)为保证数据流最小延迟上界,对于为数据流提供参数为 (R,T) 的速率-延迟服务曲线的无线自组网节点,其 TCP 数据流的服务曲线可以表示为

$$\beta_{\text{net}}(t) = \min_{1\leqslant i\leqslant n}(R_i^t)(t - SRTT/2) \tag{13}$$

证明:根据定理 1,流系统的服务曲线可以通过其流经的各个节点的服务曲线通过最小加卷积运算而得到,即

$$\begin{aligned}\beta_{\text{net}}(t) &= \beta_1^t \otimes \beta_2^t \cdots \otimes \beta_i^t \cdots \otimes \beta_n^t \\ &= \beta_{R_{\text{net}},T_{\text{net}}}(t)\end{aligned} \tag{14}$$

其中

$$\begin{aligned}R_{\text{net}}^t &= \min_{1\leqslant i\leqslant n}(R_i^t) \\ T_{\text{net}} &= \sum_{i=1}^{n} T_i\end{aligned} \tag{15}$$

其中 $\sum_{i=1}^{n} T_i$ 表示数据分组在 n 个节点之间的传输时间,在 TCP 的数据传输过程中,RTT 表示数据分组的往返传输时间的当前值,$SRTT$ 表示往返传输时间的平均值,为此,我们认为

$$T_{\text{net}} = \sum_{i=1}^{n} T_i = SRTT/2 \tag{16}$$

结合式(6)、(15)和(16)我们可以得到式(13),所以定理 6 得证。

根据网络演算理论,数据流系统的端到端最小延迟上界可以由其到达曲线和服务曲线的水平距离决定。因此,我们结合上述的定理 2、定理 5 和定理 6,得到 TCP 数据流在 t 时刻的数据流端到端最小时延上界:

$$d_{\text{net}}^{\min}(t) \leqslant P^t/R_{\text{net}}^t + SRTT/2 \tag{17}$$

其中 P^t 表示当前待处理的数据分组大小,$SRTT/2$ 表示 TCP 数据流的平均延迟,具体到 TCP 协议中即为平均往返时间的一半。R_{net}^t 表示当前 TCP 数据流系统的带宽,根据网络测量的原理,文献[2]得出了无线环境下实时的 TCP 带宽表达式,并仿真验证了其准确性。因此,TCP 数据流在 t 时刻的带宽可表示为

$$R_{\text{net}}^t = \frac{P^{\text{old}}}{\Delta t} = \frac{P^{\text{old}}}{T_i - T_{i-1}} \tag{18}$$

式中 P^{old} 表示 t 时刻之前的上一次发送数据包的大小,T_i 和 T_{i-1} 表示 t 时刻之前的前两次 ACK 包的接收时间。为此,我们根据当前待发送的数据包的大小和式(18)测量得到的 TCP 带宽代入到式(17),就可以计算得到 TCP 数据流单程传输的最小延迟上界 $d_{\text{net}}^{\min}$。

3.3 基于 $d_{\text{net}}^{\min}$ 的无线自组网 RTO 估计

TCP 定时器就是对数据流最小传输往返时间上界的估计[6]。上节推导出的最小延迟上界为数据流的单程传输延迟,因此,作为最小往返传输时间上界的 RTO 应为单程最小延迟上界的 2 倍:

$$RTO = 2d_{\text{net}}^{\min} = 2P^t/R_{\text{net}}^t + SRTT \tag{19}$$

令 $T^{\text{jitter}} = 2P^t/R_{\text{net}}^t$,它表示 TCP 数据包在时刻 t 的传输过程中所经历的延迟的变化值,即延迟抖动。由于数据包在传输过程中存在时间扩展和压缩现象,必须消除噪声干扰[2],因此,我们对 T^{jitter} 采用类似 RFC 2988 标准的低通滤波器实现,即

$$T^{\text{jitter}} = 3T^{\text{jitter}}/4 + |SRTT - RTT|/4 \tag{20}$$

从而得到

$$RTO = T^{\text{jitter}} + SRTT \tag{21}$$

从式(20)和(21)可以看出,基于最小延迟上界(MDUB)的 RTO 计算方法与 RFC 2988 的计算方法

主要差别就是在传输往返时间的变化值上。基于MDUB的RTO估计方法则是根据网络演算的理论求解了当前的最小延迟抖动上界，考虑了当前待发送的数据包大小 P^x 和当前的TCP数据流测量带宽 R^x_{mes}，可以适用于RTT变化较大且频繁的网络环境。而基于RFC 2988标准的RTO计算方法在估计当前的延迟变化时只能根据过去的RTT值的变化来考虑，因此只能适用于RTT变化较平缓的有线网络。

4 性能仿真与分析

我们在NS2平台上实现了基于MDUB计算RTO的TCP定时器，并用此方法与原有的RFC 2988标准算法进行了性能比较。还采用图1所示的拓扑和表1所示的实验参数对改进的RTO算法进行了仿真，并给出了实验结果。

图6和图7给出了基于最小延迟上界的RTO算法和RFC 2988算法的RTO估计值，从中可以看出，基于最小延迟上界的RTO算法比RFC 2988算法具有较小的RTO估计值，同时，基于MDUB算法的RTO变化曲线比以RFC 2988标准计算的RTO曲线要左移0.2～0.3s左右，刚好与RFC 2988 RTO曲线滞后于RTT曲线0.2～0.3s相抵消，表明基于最小延迟上界的RTO算法不会出现RTO的变化滞后于RTT从而导致RTO值估计不准确的现象。图8给出了数据包长度为500或1500字节随机值的RTT与RTO值，通过图8与图4的比较我们也可以得出这样的结论，即不会出现RTO值的变化滞后于RTT值的变化从而导致RTO估计不准确。这是因为基于最小延迟上界的RTO算法在估算当前RTT延迟时考虑了当前数据包的大小和当前的网络测量带宽，即数据流的延迟抖动 T^{jiller} 是根据网络的现时状态计算得来的。而基于RFC 2988的RTO计算方法在延迟的变化方面只能考虑过去状态的RTT变化值，没法对当前的RTT的变化进行估计，从而造成RTO变化状态滞后于RTT的现象。

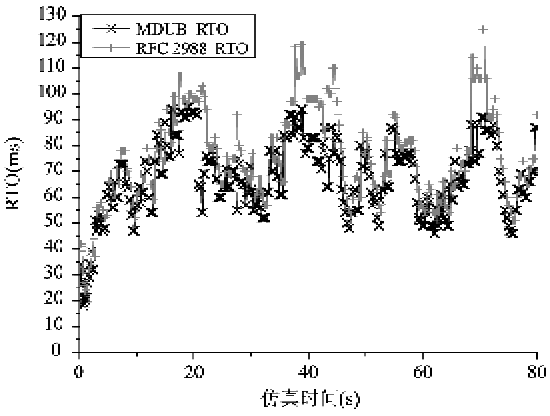

图6 TCP报文数据包为500字节的标准RTO与改进的RTO值

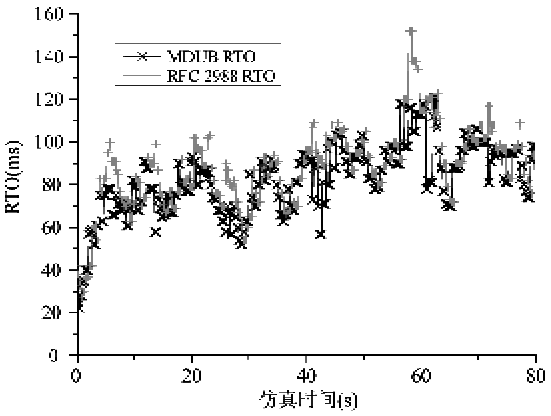

图7 TCP随机数据包为500或1500字节的标准RTO与改进的RTO值

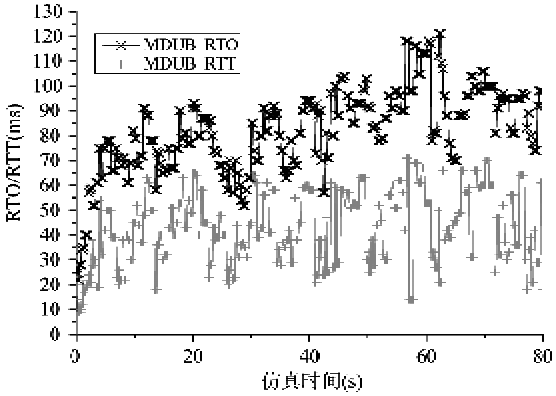

图8 数据包长度为500或1500字节随机值的RTT与RTO值

另外，我们通过表4的实验数据发现基于MDUB的RTO算法所得的RTO平均值要少于RFC 2988标准算法所得的平均值，说明基于最小延迟上界的RTO的定时器性能要优于RFC 2988标准定时器算法。图9也证实了我们的结论，即基于MDUB的RTO定时器在TCP的吞吐量性能上要优于RFC 2988标准[7]的定时器。另外，从图9我们也可以看出，数据包为500字节的TCP数据流在20～35s的时间内有1次伪重传(同样的实验环境，同样的传输控制算法，基于UMDB的RTO定时器数据流没有出现重传而基于RFC标准的RTO定时器数据流则出现重传，据此可以判断为伪重传)，而数据包随机选择的500或1500字节的数据流则出现了2次伪重

传,证实了我们在第 3 节的分析,即当 TCP 数据包长度大小不一时,因为 RTT 变化更大,RFC 2988 定时器的性能会更差;同时也表明基于 MDUB 的 RTO 定时器能有效地适用于 RTT 的波动,从而减少 TCP 数据流在无线环境下的伪重传,提升 TCP 数据流的吞吐量。

表 4　MDUB RTO/RFC RTO/MDUB RTT/RFC RTT 平均值

数据包 (byte)	MDUB RTO(ms)	RFC RTO(ms)	MDUB RTT(ms)	RFC RTT(ms)
500	78.4	81.6	38.8	38.9
1500	146.6	162.4	78.6	78.5
500 或 1500	84.7	92.7	46.4	46.4

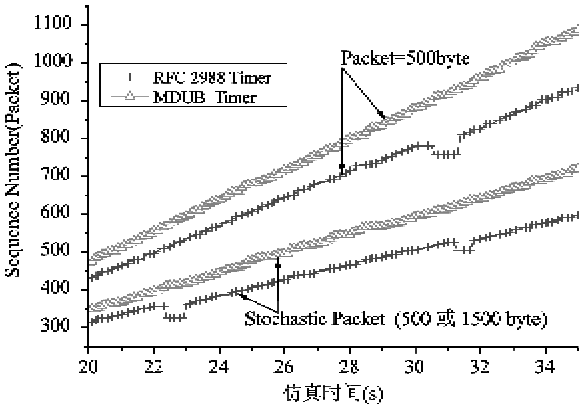

图 9　不同 TCP 定时器的吞吐量性能比较(20 ~ 35s 时间段)

为验证基于 MDUB 的 RTO 定时器在不同拓扑环境下的性能,我们设计了如图 10 所示的网格拓扑[19],在其上同时运行 4 个 FTP 数据流(只统计其中的 FTP1 数据流)。采用 TCP 发送端数据发送的字节数作为不同 TCP 定时器协议性能的评价指标。实验中节点之间距离为 200m,MAC 层选用表 1 所示的参数,路由层采用静态路由机制。

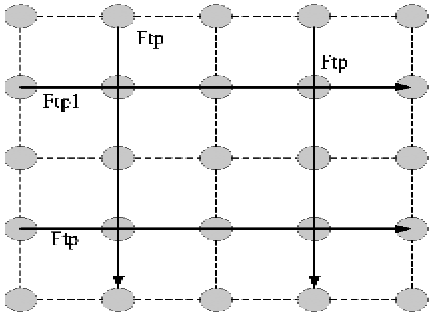

图 10　附加 4 个 FTP 数据流的网格拓扑无线自组网

图 11 给出了网格拓扑环境下基于不同 TCP 定时器的 FIP1 吞吐量比较。从可以看出,在网格形式的无线自组网环境下,基于 MDUB 的 RTO 定时器在 TCP 的吞吐量性能上要优于 RFC 2988 标准的定时器,同时也验证了基于 RFC 2988 标准的 TCP 定时器在数据报大小不一的情况下其性能会更差,而改进的 MDUB 定时器则能较好地适用于数据包大小不同的 TCP 数据流。

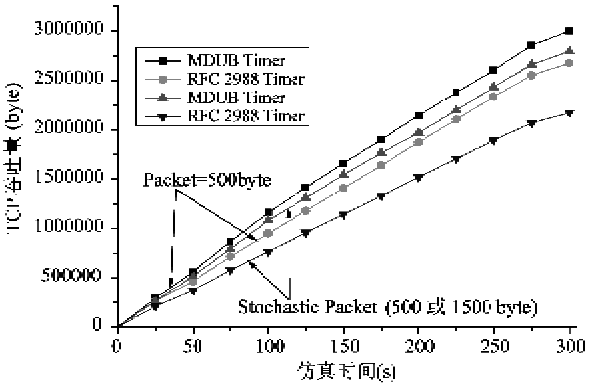

图 11　网格拓扑环境下基于不同 TCP 定时器的 FTP1 吞吐量比较

5　结　论

针对无线多跳网络环境下 TCP 数据流的 RTT 值变化较大,RTO 估计值的变化滞后于 RTT 变化从而导致 RTO 估计不准确的现状。本文利用网络演算理论,结合网络带宽测量技术推导了 TCP 数据包传输的最小延迟上界并以此改进了 RTO 的估计算法。仿真实验表明:基于 MDUB 定时器的 TCP 数据流能有效估计最小传输回路上界,克服了基于 RFC 2988 标准定时器在无线环境下 RTO 变化滞后于 RTT 变化的现象,减少了 TCP 数据流的伪重传,提升了 TCP 协议在无线自组网环境下的效率。

参考文献

[1] 张磊,王学慧,窦文华.无线自组织网络中 TCP 流公平性的分析与改进.软件学报,2006,17(5):1078-1088

[2] 邓晓衡,陈志刚,张连明,朱从旭.TCP Yuelu:一种基于有线/无线混合网络端到端的拥塞控制机制.计算机学报,2005,28(8):1342-1350

[3] Hanbali A A, Altman E, Nain P. A Survey of TCP over Ad Hoc Networks. http://www-sop.inria.fr/maestro/personnel/Philippe.Nain/PAPERS/TCP/TCP-Survey-Ad-Hoc.pdf.

[4] Zheng F, Petros Z, Luo H Y, et al. The impact of multihop wireless channel on TCP throughput and loss. In: Proceedings of the IEEE International Conference on Computer Communications, San Francisco, USA, 2003.7753-7803

[5] Sarolahti P, Kojo M, Raatikainen K. F-RTO: an enhanced recovery algorithm for TCP retransmission timeouts, In: Proceedings of the ACM Special Interest Group on Data Communications, New Delhi, India, 2003. 328-339

[6] Gurtov A, Ludwig R. Responding to spurious timeouts in TCP. In: Proceedings of the IEEE International Conference on Computer Communications, San Francisco, USA, 2003. 6652-6661

[7] Paxson V, Allman M, Computing TCP's Retransmission Timer, RFC 2988, May 2000

[8] Psaras I. Tsaoussidis V. Why TCP timers (still) don't work well. *Computer Networks*, 2007,51:2033-2048

[9] Holland G, Vaidya N. Analysis TCP Performance over Mobile Ad Hoc Networks. *Wireless Networks*, 2002,8(2):275-288

[10] 周建新,邹玲,石冰心. 无线网络TCP研究综述. 计算机研究与发展,2004,41(1):53-59

[11] Zhu Y J, Jacob L K. On making TCP robust against spurious retransmissions. *Computer Communications*, 2005,28:25-36

[12] Tamura, Tobe Y, Tokuda Y. H. EFR: a retransmit scheme for TCP in wireless LANs. In: Proceedings of the 23rd IEEE Annual Conference on Local Computer Networks, Belgium, 1998. 653-659

[13] Gresis M. The Network Simulator ns 2. http://www.isi.edu/nsnam/ns/: ISI,2010

[14] Floyd S, Henderson T. The newreno modification to TCP's fast recovery algorithm. Technical Report, RFC 2582, Internet Engineering Task Force, 1999

[15] Le boundec J Y, Thiran P. Network Calculus. London, Britain: Springer Verlag, 2004

[16] Cruz R L. A Calculus for Network Delay, Part Ⅰ: Network Elements in Isolation. *IEEE Transactions on Information Theory*, 1991,37(10):114-130

[17] 王子君,许维胜,王中杰等. 控制网络的确定性延迟演算理论研究. 电子学报,2006,34(2):380-384

[18] 李庆华,陈志刚,张连明等,基于网络演算的无线自组网QoS性能确定上界研究. 通信学报, 2008, 29(9):32-39

[19] Robinson J, Knightly E W. A performance study of deployment factors in wireless mesh networks. In: Proceedings of the International Conference on Computer Communications, Anchorage, Alaska, USA, 2007.5324-5331

Analyzing and improving the TCP timer in wireless ad hoc networks

Chen Zhigang, Li Qinghua, Deng Xiaoheng, Huang Guosheng

(School of Information Science and Engineering, Central South University, Changsha 410083)

Abstract

Based on the simulation and analysis of the relationship between the retransmission time-out (RTO) and the round-trip time (RTT) of TCP data flow in wireless ad hoc networks, the paper points out that the RTO estimation according to the RFC 2988 is not accurate as the variation of the RTO value is behind the variation of the RTT value over wireless ad hoc networks. And inview of this, it derives the TCP transmission minimum delay up bound (MDUB) using the network calculus theory and then improves the RTO estimation algorithm based on the MDUB. The simulation results show that the MDUB-based RTO algorithm can estimate TCP minimum transmission round-trip time up bound accurately, so the TCP data flow based on the MDUB timer can reduce the TCP spurious retransmission and then promote the TCP performance in wireless ad hoc networks.

Key words: wireless ad hoc network, 802.11, TCP timer, network calculus, minimum delay up bound