

MANET 网络 TCP 非拥塞控制识别序列与恢复

王雪飞

(黄山学院信息工程学院, 黄山 245011)

摘要: 针对 MANET 拥塞控制假象与真正的拥塞所需要的区分问题, 提出非拥塞控制的 3 种类型以及 4 类引起包错误的类型。采用普适算法与识别序列, 给出具体映射系数 NET_ERR_0 的数值范围, 启动相关错误恢复算法, 有效解决了真假拥塞对网络的影响。仿真证实识别序列降低了拥塞启动次数。

关键词: 拥塞控制; 错误恢复算法; 拥塞算子

Recognition Sequence and Recovery of TCP Non-congestion Control in MANET

WANG Xue-fei

(School of Information Engineering, Huangshan University, Huangshan 245011)

【Abstract】 Aiming at the question of MANET's real and fake congestion, this paper puts forward three types of non-congestion control and four types of bringing the package error. Adopting universal arithmetic and recognition sequence, gives the particular value to map NET_ERR_0 , and starts interrelated error recovery arithmetic. This paper resolves the influence of real and fake congestion. The simulation approves that the recognition sequence greatly reduces the times of congestion.

【Key words】 congestion control; error recovery arithmetic; congestion arithmetic operators

1 概述

TCP 使用 4 种技术(窗口、序列号、重传定时器、确认)实现 TCP 的 3 个功能: 流量控制, 差错控制, 拥塞控制, 以此保证数据的可靠传输。但由于 MANET 与有线网络差异很大, 引发拥塞控制的原因从单一的链路拥塞, 转向为物理信道、节点处理能力、节点角色变换等多因素。本文对动态下的多因素分析后, 提出影响 MANET 4 类错误类型的识别方法, 区分出非拥塞的特征算法与恢复算法, 仿真结果证实了这一算法的有效性。

MANET 的 TCP 拥塞控制研究是近年的热点问题。文献[1]提出在网络层的 IP 结构中加入 ECN(Explicit Congestion Notification)标志, 指示将发生拥塞。文献[2]研究了链路差错对端到端性能的影响。TCP Westwood^[3]通过修改发送方拥塞窗口的算法来改善有线及无线环境下 TCP reno 的性能。Eifel 算法^[4]消除了意义不明的重发, 解决了由虚假超时和虚假快速重发所引起的问题。文献[5]提出了基于误码丢包的速率调节机制, 根据无线链路丢包情况获取通信链路的误码率, 在误码率上升时降低发送速率, 反之则提高发送速率, 在保证可靠性的基础上提高 TCP 的吞吐量。MAITE^[6]使用链路层信息通知 TCP 高误码产生和链路断开。而采用返回报文(如 TCP-F(TCP-Feedback), TCP + ELFN (Explicit Link Failure Notification), TCP-BuS 等)或预知模型分析, 是单性能识别后的网络拥塞控制, 效率不高。

在对路由的中断识别类型上, 如 Fix RTO 路由技术, 通过分析超时事件, 成功区分路由中断和拥塞引起的丢包。当连续发生超时事件, 发送端认为这是由路由中断而非拥塞引起的; 发送端重传没有被确认的包, 但不改变 RTT_0 的值, 直到

路由重新建立。另一方面, 结合发送端和接收端分别分析接收到的确认包和数据包的序列号来确认数据包乱序和丢包的原因, 并采取相应的措施(如临时性地禁止拥塞控制或立即恢复), 如 DOOR(Detection of Out of Order and Response)。多参数联合检验(Multi Metric Joint Identification, MMJI)依靠对 4 个参数(包间延迟差异、短时吞吐量、数据包乱序比率、数据包丢失率)的联合检验, 提高了判别网络状况的准确性。这些是对路由的网络进行的判断, 但缺少对节点与物理通道、网络分割、重整等的细致描述。

2 常规 TCP 拥塞控制算法

TCP 拥塞控制常采用 4 种算法:

(1)慢启动(Slow Start, SS): 慢启动用于探测网络的可用带宽, 拥塞窗口以指数方式增长。

(2)拥塞避免(Congestion Avoidance, CA): 拥塞避免试图避免拥塞的发生并且尽可能地探测可用带宽, 使用 AIMD (Additive Increase Multiplicative Decrease)方式来改变拥塞窗口的大小。

(3)快速重传(Fast Retransmit, FR)。

(4)快速恢复(Fast Recovery, FR)。

快速重传/快速恢复通过重复确认来探测和恢复丢包。当出现丢包失效时, 重传定时器超时, 是发现并恢复丢包的最终机制。发生超时和多个重复 ACK 确认是影响 TCP 性能的关键原因。对 MANET 的 TCP 拥塞应该避免在网络没有拥塞

作者简介: 王雪飞(1963-), 男, 实验师, 主研方向: ADHOC, WSN, 嵌入式传感器

收稿日期: 2008-06-07 **E-mail:** wxf6@hsu.edu.cn

的时候发生超时和重复确认，而过早地进入拥塞控制。

对于无线通信而言，过久地处于通信状态与发送(接收)数据会极大地缩短网络存活时间，并导致网络分割、重整、崩溃等。

3 MANET 拥塞原因

MANET 产生丢包的原因如下：

(1)无线网络的瞬态扰动、慢衰落、高速移动、切换等，导致数据包损坏甚至丢包造成了高比特误码率(BER)。

(2)无线链路带宽有限与不对称(上行与下行带宽不同)，造成 ACK 丢包。

(3)节点的移动、死亡、退出等，致使路由中断。在重新发起路由过程中，所有的数据包和确认包都会被丢弃。

(4)节点繁忙，导致路由信息或 ACK 发送超时，引发拥塞控制。

(5)网络重整与局部网络重构，造成网络的信息大量丢失(这一点与网络调度策略相关，而不是仅由路由中断引发，这时的路由并不一定发生了中断)，引发 ACK 的丢失造成拥塞。

4 MANET 拥塞识别

拥塞识别是区分出真正的网络拥塞。由于 MANET 的移动性，依靠简单的丢包机制实现的拥塞控制错误使用，增加了无线电发射与接收时间，极大影响网络的寿命。上述分析显示，实际的数据传送中，包丢失并不等同于出现网络拥塞。

4.1 丢包机制的识别

丢包主要有 3 种类型：误码率高，拥塞丢包，路由中断与网络调度引发的丢包。显然，这些是由不同原因产生的丢包，在 TCP 的联接机制中，处理应该是不同的。但在有线的网络中，这三者都理解为丢包，并简单地通过拥塞控制机制解决。误码率高的包可能通过增加校验等提高恢复率；路由的重新发现过程等引起的超时，通过对路径识别与数据队列重整等完成数据的分布式重发，网络移动接入与分离可以通过对脏数据丢弃与通告完成网络重构等。在 MANET 中，识别出这 3 个丢包类型，区别对待，是提高 TCP 性能的关键。

非拥塞丢包引发控制的机制，应该是一种错误恢复机制，它不是对网络的拥塞控制，而是对产生错误原因的消除。

在报文中(路由的或 ICMP 的)增加错误标识，以显示丢包的原因，用以启动恢复算法，而不是拥塞控制。这种机制可以减轻误码和网络中断对 TCP 的影响。

错误标识由 4 种错误产生：比特误码率(BER)，网络切换与重构，路由中断，节点繁忙。4 种错误发生触发的算法是不同的，如表 1 所示。

表 1 错误标识与相关恢复算法

名称	错误原因	算法与对策
误码率(BER)	瞬态扰动、慢衰落、高速移动	反碰撞算法、功率控制或干扰算法、链路层退避算法、跨层控制或设计、同步或异步同步与快速接入等
路由中断	节点移动、死亡、退出、信道衰减、链路阻断	多径路由、快速路由发现、退避重发(起)路由技术、多层控制等
切换与重构	信道与网络切换、重整、网络合并与分割	网络分步重整(闲-忙节点分步重整、簇头角色转换重整)、同步退避、会聚节点控制、无连接分割等
节点繁忙	运算繁忙导致对通信处理超时	路由优先算法、权分配(值、高度序号)网络低能节点发现、带宽标识限制、多径(备份)路由、路由转移或复制

4.2 丢包的路由中断类型

MANET 路由中断分为链路中断与网络中断，其中链路中断包括物理中断与链路中断、双向不对称引发的中断；而

网络中断则主要表现在分割、合并、重整、节点死亡与重生等。路由的中断分为可以恢复与不可以恢复 2 个方面。

路由中断的标识主要由路由报文发出，在这里，路由中断与拥塞具有 2 点区别：(1)拥塞感知发生在端(END 或传输层)，而节点的邻居表更新正常，路由中断一般表示邻居表中相应节点消失；(2)按需的路由发现时间小于慢启动的时间，这样在历史 RTT_0 的时间内，路由是可以重发现的，路由中断时这一时间大于 RTT_0 。

MANET 中丢包的路由原因检测，可以由 RTT 与序号检测给出判断，如表 2 所示。

表 2 路由中断的检测

名称	拥塞	路由中断
双向的 RTT 检测	不连续超时	连续超时
序号检测	序号混乱	序号连续

RTT 的变化，如果是连续的不变化或超时，则认为是路由中断，否则是拥塞倾向。

4.3 非拥塞的原因识别

下面提出一个非拥塞的原因导致的错误标识的要素序列，以完成对错误的恢复所需要的知识。

(1)误码率(BER)参数 BER_ERR ：由相应链路层与物理层、网络层，是否具备多层功率控制技术 b_1 、干扰退避算法 b_2 、同步控制 b_3 ，并用状态值表示：

$$BER_ERR = \{b_1, b_2, b_3\}$$

(2)网络判断参数 $Rout_err$ ：由多径路由或快速路由重发 R_1 、按需路由/表驱动路由 R_2 、重构算法的分步 QoS 保证 R_3 、会聚节点控制 R_4 ，完成对参数值的确定。

$$Rout_err = \{R_1, R_2, R_3, R_4\}$$

(3)节点能力与带宽参数 $Node_Br_err$ ：节点能力(吞吐量/时延) N_1 、RTT 值 RTT 、节点邻居数 N_2 、带宽标识限制 N_3 构成对参数值的确定。

$$Node_Br_err = \{N_1, RTT, N_2, N_3\}$$

以上值以离散量(1 or 0)或限制范围决定，构成如下序列：

$$NET_ERR = \{BER_ERR, Rout_err, Node_Br_err\}$$

当 NET_ERR 在确定范围之内时，如果 ACK 还是不断丢失，则启动拥塞控制，这个范围的值域是可以确定的。

采用出错序列控制的用意在于完成对错误的识别，引入上述的识别参数 NET_ERR ，并增加对错误的准确性，完成如下列表：

$$NET_ERR = \{BER_ERR, Rout_err, Node_Br_err, Number_dis\}$$

当丢包产生后，序号的不完整成为其中一个参数，同时 NET_ERR 产生一个值，为此做一个映射，并形成一列表如表 3 所示。其中，参数值为 1 表示高功率，为 0 表示低功率。

表 3 恢复算法识别序列参数映射

类型	说明	参数值
BER_ERR	高/低(多层功率控制) b_1	1/0
	不启动/启动(干扰退避算法) b_2	1/0
	有/无(同步控制) b_3	1/0
$Rout_err$	有/无(多径路由 OR 快速路由重发) R_1	1/0
	按需路由/表驱动路由 R_2	1/0
	不分层/分层(重构算法的分层性) R_3	1/0
	融合/无(会聚节点控制) R_4	1/0
$Node_Br_err$	节点能力(吞吐量/时延) N_1	1...0
	RTT 值($RTT(MAX)/RTT(MIN)$) RTT	1...0
	节点邻居数/最佳节点数 N_2	1...0
	带宽标识限制(大于应用最大带宽为 1) N_3	1/0
$Number_dis$	不完整/完整	1/0
总计和值	范围	12...0

显然当这个值较大时,说明非拥塞的 ACK 丢失控制能力已经很高了,如果还会产生 ACK 丢失,则可以认为是由于网络拥塞产生。

当这个域值为 NET_ERR_0 时,且 $NET_ERR > NET_ERR_0$, 直接启动拥塞控制机制。

在分析一些具体的 MANET 基础上,这一值存在一个值域 NET_ERR_0 , 如表 4 所示。

名称	主要使用功能	NET_ERR_0
蓝牙 Scatternet	多层功率控制、干扰退避算法、按需路由、会聚节点控制、节点能力强、节点邻居数/最佳节点数等	4.5~7.6
PC+Wlan 移动网	同上、快速路由重发、节点能力	6.0~8.5
WAN(zigbeeNET)	多层功率控制、干扰退避算法、按需路由、会聚节点控制、重构算法的分步性	>4.2

在较小 NET_ERR 时(小于 NET_ERR_0), 不启动拥塞控制是可以明显改善丢包的。

4.4 非拥塞恢复算法

当启动错误恢复算法时, 先识别出路由中断并启动一个定时器, 在一定时间等待路由恢复, 如果没有路由中断或没有发现路由中断报文, 启动 TCP_Packet 序号检测是否乱序。有, 则认为 BER_ERR 出错, 提高 BER_ERR 值, 否则增加 $Node_Br_err$ 值。

由于非拥塞的 ACK 丢失与一个或几个节点相关, 采用与源节点最近的节点启动与多节点同时启动(出错则启动)。当 NET_ERR 值已经很大时, ACK 丢失仍然存在, 则恢复算法直接启动拥塞控制中的等待定时器, 如图 1 所示。

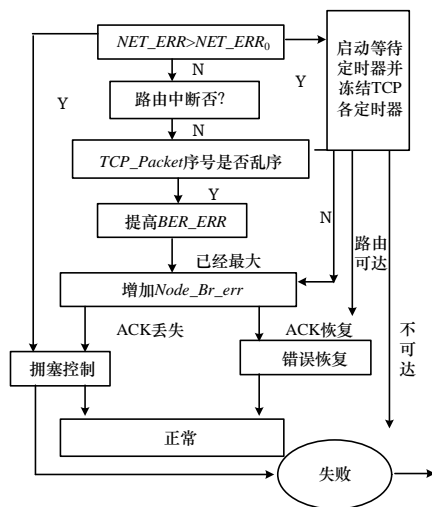


图 1 基于识别序列的恢复算法

恢复的结果有 3 个: 错误恢复, 启动拥塞控制, (路由中断不可修复)失败。

5 识别下拥塞控制与错误恢复的验证

5.1 Bluetooth Scatter Net 下的实验验证

考虑 3 个 Bluetooth Scatter Net 组成的分层 MANET, 选用 Blue-tree 路由算法(协议), 并在路由发现程序中引入错误判断生成程序, 在路由维护程序中引入错误判断与 NET_ERR_0 。给出由 21 台笔记本组成的节点, 每 7 台构成一个

Bluetooth Scatter Net, 3 个簇头构成一个簇头网, 获取实验如表 5 所示。

表 5 Bluetooth ScatterNet 实验结果处理数据(20 min)

拥塞控制	拥塞次数	错误恢复次数
有线网络 TCP 机制	214	-
ATCP	137	47
错误恢复下的拥塞控制	12	106

5.2 仿真处理中的验证

在 Blue-tree 的路由下, 对一个有 10 个簇头网(scatter net)构成的 MANET 网络, 增加识别序列 $NET_ERR = \{BER_ERR, Rout_err, Node_Br_err, Number_dis\}$, $NET_ERR > 5(NET_ERR_0)$ 时, 启动拥塞控制算法, 并与无恢复算法对比仿真。

仿真用时延表示对拥塞区分与不区分时的统计。在使用恢复算法后, 有效地避免了相应拥塞控制产生的过大 RTT, 而没有使用恢复算法时延 RTT 则呈现频繁启动, 导致 RTT 过大; 仿真如图 2 所示。

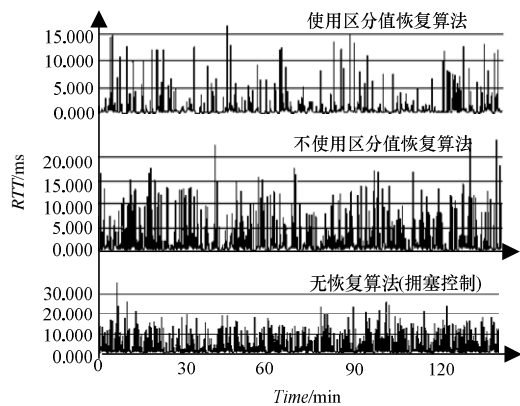


图 2 仿真结果

6 结束语

由 OPNE 仿真软件在时延上给出 Blue-tree 的仿真结论, 同样支持以下结论: 对 MANET 丢包识别的拥塞启动与错误恢复算法, 说明因为网络错误或其他错误导致的“丢包”误判, 在使用恢复算法后可以有效改善网络通信状态。

参考文献

- [1] Ramakrishnan K K, Floyd S. A Proposal to Add Explicit Congestion Notification(ECN) to IP[S]. RFC 2481, 1999.
- [2] Dawkins S, Montenegro G, Kojo M, et al. End-to-End Performance Implications of Links with Errors[S]. RFC 3155, 2001.
- [3] Gerla M, Sanadidi M, Wang Ren, et al. TCP Westwood Window Control Using Bandwidth Estimation[C]//Proc. of IEEE Global Telecommunications Conference. San Antonio, Texas, USA: [s. n.], 2001: 25-29.
- [4] Ludwig R, Katz R H. The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions[J]. ACM Computer Communication Review, 2000, 30(1): 30-36.
- [5] 徐昌彪, 隆克平, 杨士中. 无线网络中基于误码丢包的 TCP 速率调节策略[J]. 计算机学报, 2002, 25(4): 438-444.
- [6] Karn P, Partridge C. Improving Round-trip Time Estimates in Reliable Transport Protocols[J]. ACM Trans. on Computer Systems, 1991, 9(4): 364-373.