



分组传输IP网中KSPLD拥塞控制算法研究

李国领 罗天义 叶亚伟 李超杰 秦红涛

中国通信建设集团设计院有限公司第四分公司, 河南郑州 450000

[摘要] 在分组传输网络承载IP网业务中, 需要传输大量的数据信息, 针对网络拥塞控制策略缺乏有效地丢包区分机制, 本文提出一种基于卡尔曼滤波的丢包区分机制, 并将其应用于分组传输网承载以太网业务的拥塞控制策略, 算法根据稳定阶段的平均包到达时延和丢包阶段的包到达时间间隔来进行判断丢包类型, 到达间隔采用卡尔曼滤波进行估计, 减小测量误差; 然后根据丢包类型重新计算丢包率, 从而控制发送端的速率。实验表明该算法能使拥塞控制策略更准确识别网络状态, 提高利用率, 使其能够更好的满足信息流应用。

[关键词] 分组传输网; 丢包区分; TFRC; 拥塞控制; 卡尔曼滤波; IP网

1 引言

随着通信技术以及多媒体技术的发展, 移动互联网多媒体应用日益广泛, 分组传输网需要承载更多的业务的需求, 尤其是 IP 网业务;

在使用分组传送网传输 IP 网业务时, 传统的拥塞控制结果不是很理想, 本次分为两个层次对拥塞进行管理域控制, 一个是链路层等底层对拥塞进行处理, 另一个是在分组传送 IP 网传输层使用拥塞策略。

信息流传输的过程中, 网络拥塞的管理与控制极为重要; 分组传输网的在传输数据信息流, 在传输 IP 网中的业务时, 拥塞控制常常在 3 层以下起作用, 3 层以上的拥塞处理结果不满意, 尤其^[1]TCP 拥塞控制其严格的重传、确认机制, AIMD 拥塞控制其波动的传输速率和时延抖动, 使得虽然能满足数据的可靠传输, 但却无法满足实时数据信息流的延时敏感, 高带宽, 以及丢包容忍。因此, 一些数据信息流传输解决方案采用 UDP 协议进行数据信息流数据的传输。但是由于 UDP 的不可靠传输, 这将会大大降低数据信息流传输的质量, 使其无法满足需求^[2]。在有线网络中, TCP 及 TFRC^[3]等传输协议其核心是拥塞控制, 该机制将丢失的分组视为拥塞的标志, 但这只适用于有线网络, 在无线网络中, 由于其较高的信道误码, 导致误码丢包成为 IP 网络中数据信息流传输不可忽视的部分^[4]。

2 相关工作

2.1 丢包区分

目前基于丢包区分的改进方法主要基于端到端的以及链路层优化, 由于链路层能够对于误码丢包具有很好的识别性, 使得基于跨层的丢包区分算法得到较好发展, 但由于需要涉及对网络节点的改造, 实现较为复杂。基于端到端的丢包区分算法主要包含有 Bias^[5], SPLD^[6], mBias^[7], spike^[8] 和 Zigzag^[7], Bias 算法的实现是通过计算端包到达时间间隔, 然后通过与最小包到达间隔进行比较进行丢包区分。但在共享链路存在多流的情况下, Bias 区分度较低。SPLD 使用平均累计包到达间隔和当前包到达间隔来进行丢包区分, 在单流和多流的情况下, 表现较好, 但吞吐量较低。spike 算法通过单向传输时延和当前拥塞状态来进行丢包区分, 但区分度较低, 不能准确区分拥塞丢包和随机误码丢包。Zigzag 算法时延丢包个数和单向传输时延进行区分, 将单向传输时延和丢包个数进行阶梯状比较, 从而进行丢包区分, 但在大部分场景下表现较差。

2.2 卡尔曼滤波

卡尔曼滤波^[9] 是以最小均方误差为估计的最佳准则, 来寻求一套递推估计的算法, 其基本思想是: 采用信号与噪声的状态空间模型, 利用前一时刻的估计值和现时刻的观测值来更新对状态变量的估计, 求出现在时刻的估计值。卡尔曼滤波的实质是由量测值重构系统的状态向量。它以“预测—实测—修正”的顺序递推, 根据系统的量测值来消除随机干扰, 再现系统的状态。

一个有线性可微方程控制的随机事件可有如下方程进行表示:

$$X(k) = AX(k) + Bu(k) + w(k) \quad (1)$$

$$Z(k) = HX(k) + v(k) \quad (2)$$

其中 A 为状态转移矩阵, 描述系统状态由时间 k-1 的状态到时间 k 的状态之间的转移; H 为量测矩阵, 描述状态经其作用, 变成可量测或可观的; X(k) 为状态向量, 是不可观测的; Z(k) 为观测向量; w(k) 为过程噪声; v(k) 为量测噪声。

为了分析方便, 假设状态变量的增益矩阵 A 不随时间发生变化, w(k), v(k) 都是零均值白噪声, 方差分别是 Q 和 R, 并且初始状态 x0 与 w(k), v(k) 都不相关, 且噪声向量 w(k), v(k) 也互不相关, 即 $w \sim N(0, Q)$, $v \sim N(0, R)$ 同时没有

外界控制输入，也即 $u(k)$ 为 0。

使用卡尔曼滤波估计状态值步骤如下：首先根据经验值做本次的先验估计和协方差即

$$X'(k) = A * X(k) + w(k) \quad (3)$$

$$P = A * P * A' + Q \quad (4)$$

根据式 (4) 计算卡尔曼增益。

$$K_g = (P * H') / (H * P * H' + R) \quad (5)$$

然后根据式 (3) 计算本次状态值如式 (6) 所示：

$$X(k) = X'(k) + K_g * [Z(k) - X'(k)] \quad (6)$$

其中 $Z(k)$ 为 k 时刻测量值，由式 (2) 可得。

最后更新后验估计协方差如式 (7) 所示：

$$P = (1 - K_g * H) * P \quad (7)$$

2.3 友好性拥塞控制协议

TFRC 是基于 TCP 吞吐量模型的端到端的 TCP 友好性拥塞控制协议。他的基本思想是利用吞吐量模型来调节 TFRC 流的速率，从而保证与 TCP 流的友好性。其吞吐量模型如所示：

$$R = \frac{S}{t_{RTT} * \sqrt{\frac{2bp}{3}} + p * (1 + 32p^2) * t_{RTO} * \min(1, 3\sqrt{\frac{abp}{8}})} \quad (8)$$

式中 S 代表包大小， t_{RTT} 表示往返时延， t_{RTO} 表示传输超时时限， p 表示丢包率， b 表示一次 ACK 的确认包数。从公式可见，模型的准确性受反馈参数的影响，这就要求在应用时构成一个合理的反馈环路。

3 算法描述

本文提出一种改进的丢包区分算法 KSPLD，并将其运用于拥塞控制策略上，将网络误码丢包和拥塞丢包作为重点，在接收端使用卡尔曼滤波对包到达时延进行精确估计，然后通过对比稳定时平均累计包到达时延与当前累计包到达时延进行比较，从而提高区分精度和拥塞控制协议的性能。

本文在 SPLD 算法^[6]的基础进行改进，使用卡尔曼滤波来提高包到达间隔时延的测量精度，从而能够更好的提高区分精度，对网络状态有更精确的判断。

其算法如图 1 所示：其 IAT_i 为当前阶段下，经过卡尔曼滤波得到的连续第 i 个没有出现丢包情况下的包到达时间间隔， IAT_{stable} 表示连续包累计到达时间间隔， IAT_{avg} 表示平均包到达间隔， n 为当前阶段接收包个数， IAT_{loss} 丢包前最后包到达间隔， m 表示丢包时刻检测的丢包数。

$$IAT_{stable} = \sum_{i=last}^{now} IAT_i \quad (9)$$

$$IAT_{avg} = \frac{IAT_{stable}}{n-1} \quad (10)$$

$$IAT_{cur} = \frac{IAT_{stable} + IAT_{loss}}{n+m-2} \quad (11)$$

无线误码的包从发送端发送后会由于信道原因出现错误，导致其在接收端的 MAC 层校验出现错误，无法传送到应用层，所以在这期间的包达到时延变化不大，因此当 $IAT_{cur} \geq IAT_{avg}$ 时，程序认为丢包事件产生原因是由于无线误码，否则为拥塞丢包，清除当前状态信息，开始新的阶段。

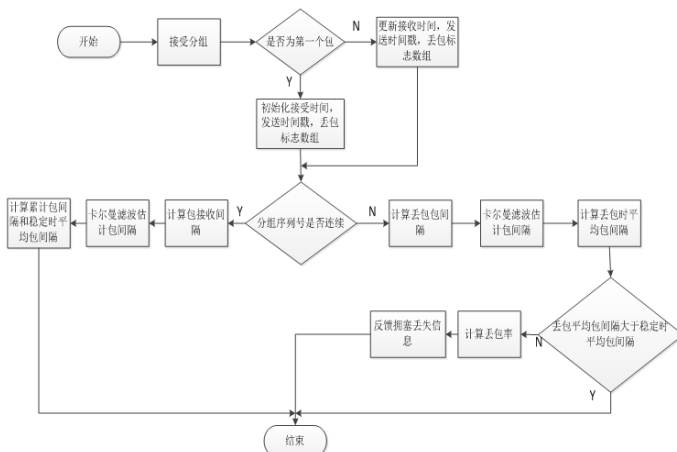


图 1 分组传输 IP 网络丢包区分算法流程图

丢包区分算法得到的结果最终将为拥塞控制策略提供更精确的网络状态的判断，从而使得在满足友好性的同时满足吞吐量的最大化，由于 TCP 本身的拥塞控制策略会导致较低的吞吐量和较高的时延，使其无法满足数据信息流流的基本需求，因此本文采用 TFRC 作为拥塞控制协议。与 TCP 流竞争带宽是相当公平的，但是与 TCP 相比，吞吐量随着时间的变化要小得多，使得它更适合于以相对平滑的发送速率为特征的流媒体应用。因此本文将丢包区分算法应用到 TFRC 拥塞控制之中，提出 KSPLD-TFRC 拥塞控制算法。算法主要修改接收端的反馈机制，当检测到丢包后，通过卡尔曼

滤波区分丢包类型，之后根据丢包类型，当出现拥塞丢包时，按照 TFRC 原始的反馈机制进行反馈，当出现误码丢包时，不立即反馈丢包，而是等待超时反馈。同时更新丢包率。

改进后的 TFRC 关于丢包反馈的伪代码如图 2 所示。

```

TFRC 丢包反馈代码:
if (lossType == CongestionLoss) {
    flag=1;
    update(lossvector, Loss);
} else {
    flag=0;
    update(lossvector, ERR);
}
if (flag) {
    nextpkt(p);
}
TFRC 超时反馈代码:
TfrcNackTimer::expire(Event*) {
    a_ ->nextpkt(-1);
}
    
```

图 2 TFRC 原理框图

4 流传输仿真实验

本文利用 NS2 仿真^[10] 软件验证本文算法的有效性。并与现有的 mBiaz, SpId 区分算法进行比较，同时对比没有区分算法的 TRFC 算法，其算法的性能有以下几个指标衡量：

吞吐量：吞吐量是衡量拥塞控制算法效果的重要指标，在保持与 TCP 协议友好性的前提下，吞吐量越高，拥塞控制的效果越好。本文将吞吐量进行归一化处理，

$$B = \frac{Out}{BC} \quad (12)$$

其中 Out 为当前的数据发送速率，BC 为当前无线网络带宽。

误判率：误判率包括将拥塞丢包视为误码丢包 Mc 和将误码丢包视为拥塞丢包 Mw，以及总的误判率 Mc+Mw，其中 Mc 通过算法误判为误码的包总数与实际总拥塞包数的比值得到。Mw 通过算法误判为拥塞的总包数与实际总的误码包数的比值达到。总的误判率为

$$M = \left(\frac{Nw}{Nac} + \frac{Nc}{Naw} \right) * 100\% \quad (13)$$

其中 Nw 为算法误判为误码的总包数，Nac 为实际总拥塞包数，Nc 为算法误判为拥塞的总包数，Naw 为实际总的误码包数。

公平性：TFRC 协议设计的目的就是为能够与 TCP 保持友好性的前提下，保证流媒体的稳定传输，因此本文定义公平性为两个流吞吐量之间的比值

$$Fair = \frac{Out_1}{Out_2} * 100\% \quad (14)$$

实验拓扑采用无线最后一跳 (wireless last top WLH) 网络拓扑结构。这类型拓扑适用于蜂窝网络或者卫星网络中的通信仿真^[4]。同时无线链路使用 Gillbert-Elliott 模型进行描述网络误码行为。GE 模型中可作为连续性数据遗失模型，通过传输通道的好坏状态来判断当前网络的丢包率，其整体的封包遗失率可用式表示：

$$P = P_G * \frac{P_{BG}}{P_{BG} + P_{GB}} + P_B * \frac{P_{GB}}{P_{BG} + P_{GB}} \quad (15)$$

式中 P_G 表示信道在“好”状态 (G) 下发生丢包的概率，P_B 表示信道在“好”状态 (B) 下发生丢包的概率，P_{BG}, P_{GB} 表示信道状态转移概率。同时对于无线网络不同的传输模式下，网络对丢包的处理方式不同，当采用广播的方式传输封包时，如果出现丢包，传送端不会重传该包，而是传送下一封；当采用单播时，在出现丢包时，发送端会利用重传机制来重传该包，以降低封包遗失率。所以为防止设置封包遗失率与实际不吻合，设置为单播模式，且重传次数为 1。

设置 P_{BG} 为 0.06，P_{GB} 为 0.04，P_G 为 0.001，P_B 为 [0.05, 0.5]，则平均丢包率 P 为 [2.26%，20.06%]。

本文实验具体的仿真图如下所示：

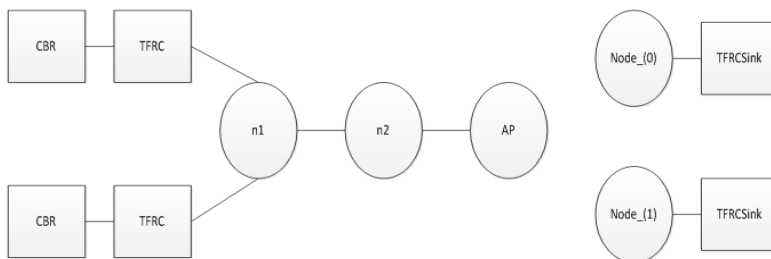


图 3 网络拓扑图

为了计算方便，本文的采用一阶卡尔曼滤波，也即式中 A 为常量 1，H 也为常量 1，滤波中的误差方差 Q 取 0.018s，R 根据实际测量数据的偏差选取 0.005s。具体的参数设置如表 1 所示。

表 1 实验参数设置

无线信道 MAC 层协议	802.11b
信道带宽	11Mb
信道基本传输速率	1Mb
无线链路传输有效距离	250m
链路层最大重传次数	1
共享链路带宽	10Mb
共享链路时延	2ms
基站队列长度	20
CBR 单流传输速率	1Mb
应用层数据包大小	762Byte
CBR 流开始传输时间	100s
CBR 流持续传输时间	100s

在 CBR 单流情况下，数据从有线节点 n1 到无线节点 node_(0)，其误判率和吞吐量如图 4 图 5 所示：

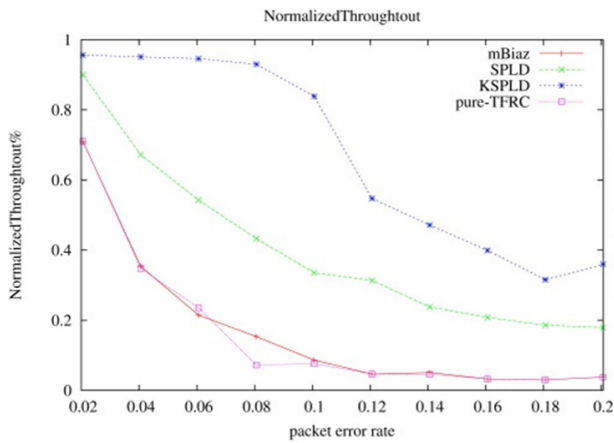


图 4 单流吞吐量

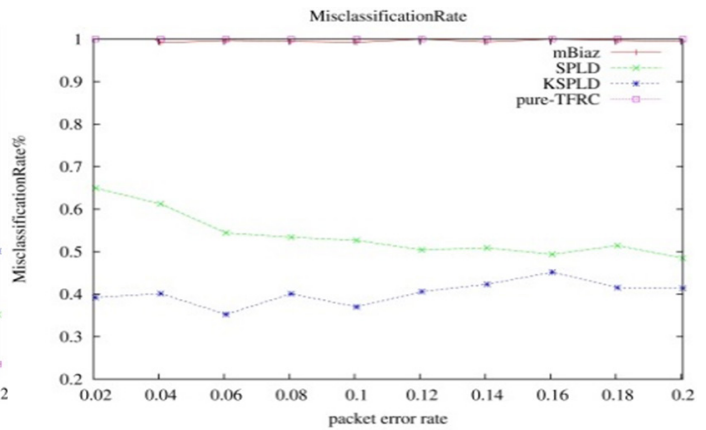


图 5 单流误判率

从图 5 中可以看出，相比于 mBiaz 和 SPLD，本文提出的丢包区分算法 KSPLD，在误判率上有了很大的提升 SPLD 的误判率维持在 0.6 左右，而 KSPLD 算法的平均误判率只有 0.3，提高近一倍的效率，同时，由于 TFRC 本身不具有区分能力，所以误判率为 100%，同时在误码率上升时，吞吐量都有不同程度的下降，相比于 SPLD 区分算法以及原始的 TFRC 算法，本算法更精确的丢包区分使得 TFRC 能将网络效率进行最大化利用，能够准确识别误码，防止误判导致吞吐量下降的太快，数据显示本文所提算法在单流情况下能够维持较高的吞吐量。

在 CBR 双流的情况下，数据流 1 从 n1 接到 node_(0)，数据流 2 从 n1 到 node_(1)，其误判率和吞吐量如图 6, 7 所示。

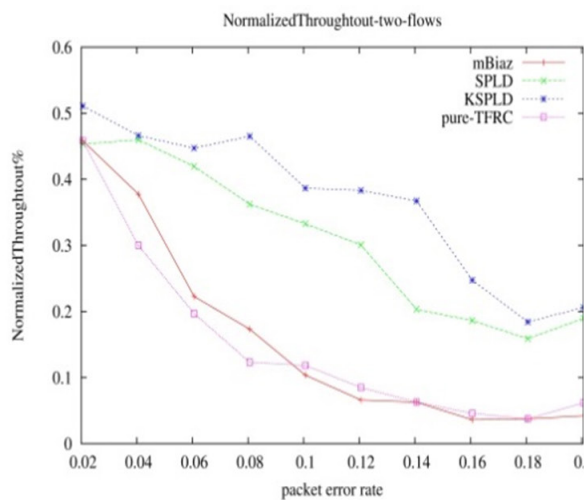


图 6 双流吞吐量

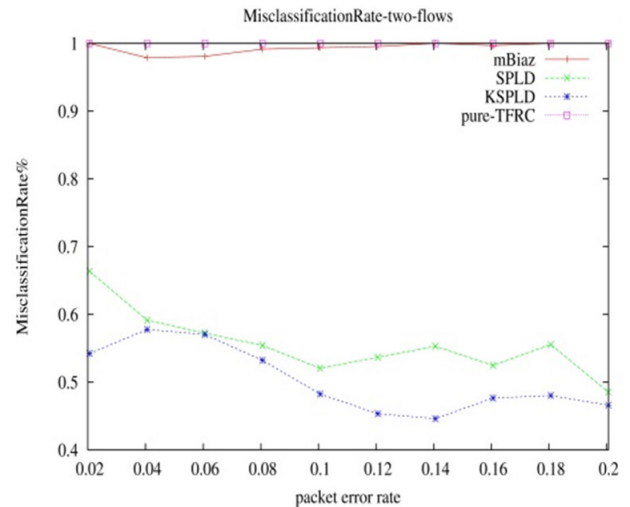


图 7 双流误判率

由于存在多个流，导致各个算法的误判率出现了一定的抖动，但从整体上来看，相比于 SPLD 区分算法，本文提出的 KSPLD 区分算法能维持更低的误判率，同时吞吐量上也能反映出来，由于更低的误判率导致更高的吞吐量。

为了评判公平性，本文采用使用相同区分算法的流进行吞吐量的比较，该值越接近于 1，说明其公平性更好，其吞吐量公平性如图 8 所示：

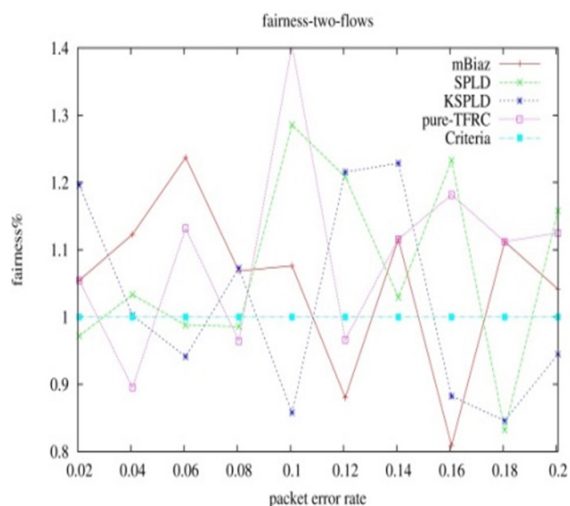


图 8 双流公平性

在吞吐量较高的情况下，本文所提出的算法，也能很好的满足公平性的要求，使得多流之间能够更充分的利用网络带宽。

5 结论

本文提出一种基于卡尔曼滤波的丢包区分算法，该算法在接收端对包到达时延间隔进行滤波估计，根据稳定时段的平均包到达时间间隔和丢包时的时间间隔来判断当前状态的丢包类型，同时根据不同的丢包类型重新计算丢包率。

基于该区分算法，本文结合 TFRC 协议，并在其基础上进行仿真实验，根据实验结果可以得出，本文提出的拥塞控制改进策略能够更好的适应分组传输 IP 网，更准确的区分不同的丢包类型，从而使得发送端对网络有更准确的感知，提高网络吞吐量。

[参考文献]

- [1] 靳玉宝, 董育宁. 异构网络中的丢包区分和拥塞控制机制 [J]. 应用科学学报, 2015, 33(3):234-242.
- [2] Carlucci G, Cicco L D, Ilharco C, et al. Congestion control for real-time communications: A comparison between NADA and GCC [C]. Control and Automation. IEEE, 2016:575-580.
- [3] Floyd S, Handley M, Padhye J. Equation-based congestion control for unicast applications [C]. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. ACM, 2000:43-56.
- [4] Lee H, Choi C H. A Loss Discrimination Scheme for TFRC in Last Hop Wireless Networks [C]. IEEE Wireless Communications and NETWORKING Conference. IEEE, 2007:3082-3086.
- [5] Biaz S, Vaidya N H. Discriminating congestion losses from wireless losses using inter-arrival times at the receiver [C]. Application-Specific Systems and Software Engineering and Technology, 1999. ASSET '99. Proceedings. 1999 IEEE Symposium on. IEEE, 1999:10-17.
- [6] Min K P, Sihm K H, Jeong J H. A statistical method of packet loss type discrimination in wired-wireless networks [C]. Consumer Communications and NETWORKING Conference, 2006. Ccnc. IEEE Xplore, 2006:458-462.
- [7] Cen S, Cosman P C, Voelker G M. End-to-end differentiation of congestion and wireless losses [J]. IEEE/ACM Transactions on Networking, 2003, 11(5):703-717.
- [8] H. Tokuda, A. Molano, Y. Tobe, Y. Tamura and et al. Achieving moderate fairness for UDP flows by path-status classification [C]. Proceedings 25th Annual IEEE Conference on Local Computer Networks, LCN 2000(LCN), Tampa, Florida, 2000, 252.
- [9] Greg Welch, Gary Bishop. An introduction to the Kalman filter [EB/OL]. <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>. [2007-03-08].
- [10] 柯志亨, 程荣祥, 邓德隽. NS2 仿真实验: 多媒体和无线网络通信 [M]. 电子工业出版社, 2009.