

基于学习的扩展 AIMD 拥塞控制机制

吴元保¹, 刘振盛¹, 张文良²

(1. 武汉大学电气工程学院, 武汉 430072; 2. 河北临港供电公司, 黄骅 061113)

摘要: 在 Internet 中, 主要传输控制协议采用了和式增加积式减少(AIMD)的拥塞控制机制, 直接应用该机制不能适应多媒体通信的要求, 为了解决好 Internet 中的多媒体通信以及满足媒体流服务质量的要求, 该文针对实时视频传输的实际应用, 在点对点的单播速率控制方面和扩展 AIMD 速率控制基础上, 提出一种以丢包为参数具有一定学习能力的速率控制机制即基于学习的扩展 AIMD 拥塞控制机制, 并研究其实现原理和方法。应用表明, 该机制具有平滑数据发送速率、降低丢包数等特点。

关键词: 扩展和式增加积式减少(AIMD); 拥塞控制; 多媒体通信

Congestion Control Mechanism of Study-based Expanded AIMD

WU Yuan-bao¹, LIU Zhen-sheng¹, ZHANG Wen-liang²

(1. School of Electrical Engineering, Wuhan University, Wuhan 430072; 2. Hebei Lingang Power Company, Huanghua 061113)

【Abstract】 In the Internet, congestion control out of TCP protocol by Additive Increase and Multiplicative Decrease(AIMD) fails to satisfy the need for multimedia communications. To meet the need of multimedia communications in Internet in desirable Quality of Service(QoS), a fresh congestion control mechanism of Study-based Expanded AIMD(SB-EAIMD) rate control with the number of multimedia package loss is put forward in support of the point to point multimedia communication, with detailed research of its operation principle and approaches.

【Key words】 expanded AIMD; congestion control; multimedia communications

在Internet中, 多媒体通信系统得到了广泛的应用, 但由于受到网络带宽的限制, Internet上的多媒体通信受到了制约。网络拥塞控制成为保证多媒体通信服务质量(QoS)的一个关键问题, TCP协议使用的是基于窗口的和式增加积式减少(AIMD)端到端拥塞控制机制。该机制在Internet上进行数据传输时采用尽量好的服务^[1], 但出现拥塞时数据量会突然减半, 致使通信服务质量显著下降, 这对于具有实时特性和时延抖动受限的多媒体通信数据流的传输而言, 显然不能满足其QoS要求^[2]。因此, 本文提出了一种基于学习的扩展AIMD拥塞控制机制(SB-EAIMD), 避免了通信服务质量显著下降的问题。

1 AIMD 的扩展

依据AIMD的速率控制存在着不利于视频实时传输的缺点(如较大的抖动性)。在稳定状态下, 即使不考虑超时和重传, 速率调整也呈现出锯齿形的波动^[3]。如果考虑超时、重传以及慢启动或者快速启动等过程, TCP的速率控制也具有很大的抖动性。在未考虑超时和重传的情况下, 网络稳定状态下按照AIMD控制(乘性因子为 0.5, 加性因子为 1)的发送速率周期抖动情况。假设 W_0 表示网络的实际吞吐量, W_1 表示了发送速率抖动的下限。在网络动态变化的影响下, 即在 W_0 的变化下, AIMD的调节可能出现 $W_1 \rightarrow 0$ 的情况。这种抖动显然不利于实时视频流的网络传输。网络上大量依照AIMD进行速率控制, 形成网络周期性的振荡。不但速率平滑得不到保障, 网络带宽也存在大量浪费^[4]。

针对实时视频传输的特点, 本文提出一种理想的速率变化曲线, 如图 1 所示。假设一个用户所需要的视频传输最大速率为 W_{max} , 即在发送速率为 W_{max} 的情况下, 再增加发送速

率对视频质量没有改善(或者改善甚小, 可以由PSNR中的一个阈值来设定)。另外假设一个用户所需要的视频传输最小速率为 W_{min} , 即在用户可以接受最低质量的视频发送速率为 W_{min} , 如果发送速率低于 W_{min} , 其视频质量用户无法接收, 其占用的网络带宽也是浪费的。此时的处理方式可以有 2 种: (1)以 W_{min} 作为发送速率, 但是不能保证终端用户可以接收到以 W_{min} 发出的视频, 严重的情况下可能导致网络进一步拥塞; (2)处理方法是发送速率降低为 0, 即停止发送视频。此时可以缓解网络带宽的紧张, 保证某些用户可以正常接收视频。

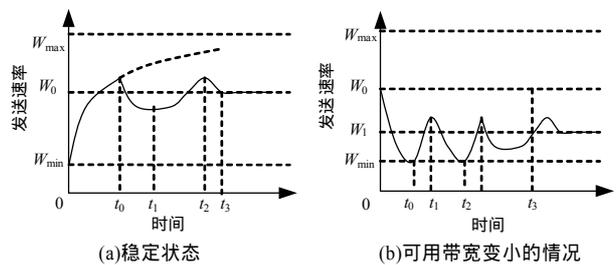


图1 理想的发送速率曲线

本文提出的速率控制的基本思想是: 在稳定状态下, 视频发送速率以 W_{min} 启动, 在低于网络可用带宽 W_0 的阶段快速上升, 在 $[W_0, W_{max}]$ 阶段则缓慢上升, 避免过快地发生网络拥塞; 在发现网络拥塞的情况下(t_0 处), 发送速率迅速减小, 但

基金项目: 国家科技部技术创新基金资助项目(02C26224210095)

作者简介: 吴元保(1963 -), 男, 副教授、博士, 主研方向: 多媒体通信, 信息技术, 通信网络; 刘振盛, 讲师、博士; 张文良, 助理工程师

收稿日期: 2007-05-10 **E-mail:** yuanbaowu@163.com

在区间 $[W_{\max}, W_0]$ 阶段则缓慢减小, 以避免快速减小导致的发生速率振荡。以此模式持续控制, 则在稳定状态下, 发送速率将稳定在 W_0 处, 呈小幅度的振动, 如图 1(a)所示。在带宽变小的情况下, 假如网络可用带宽由 W_0 变为 W_1 , 由于网络发送原来已经稳定在 W_0 附近, 此时在小幅振荡的下限位置(近似于图 1(a)中的 t_3 处)检测到拥塞, 则发送速率在 $[W_1, W_0]$ 间快速下调, 但为防止速率的大幅振荡, 在 $[W_{\min}, W_1]$ 间缓慢下调, 下调范围不会超出 W_{\min} , 最低处停止在 W_{\min} 处, 如图 1(b)的 t_0 处, 如果仍然拥塞, 则按照一定的随机概率将发送速率确定在 W_{\min} 或者停止发送。如果下调到不出现拥塞了, 则按照再次稳定状态的控制过程将发送速率控制在 W_1 附近。整个变化过程如图 1(b)所示。

为了设计具有这种特性的速率控制曲线, 首先要对 AIMD 控制方法进行调整, 增加平滑性的修正。其次增加一系列基于历史的调整规则, 使速率控制更接近理想的速率发送曲线。

先考察在 AIMD 的变化曲线。假设增加的步幅是 b , 乘性递减的因子是 a , 以时间 t 为自变量的拥塞控制窗口 $W(t)$ 在一个控制周期内的变化曲线为

$$W(t) = \begin{cases} W_0 \cdot a^t & 0 < a < 1 \\ b(t - t_0) & b > 0 \end{cases} \quad (1)$$

其中, W_0 和 t_0 为常数, W_0 为 $t=0$ 时的拥塞窗口初始状态; t 是一个最小的往返时间 RTT 的整数倍, 即 $t=n \cdot RTT, n=0, 1, \dots$, 所以拥塞窗口的变化曲线是离散的。在实际的控制中由于受到超时和重传等影响, 拥塞窗口的控制曲线比这更复杂。为了便于分析, 将之简化成连续曲线。

在拥塞状态下, 如(1)描述, AIMD 的控制曲线是一条下降的以 a 为底的指数曲线。曲线的下降下限为 0; 在非拥塞状态下, 其控制曲线是一条以 b 为斜率的上升直线。

由上述分析可知, 由于下降的指数曲线在初始阶段快速下降而在接近下限的阶段缓慢下降, 这种特性针对于视频传输来说也是比较合适的。考虑到视频的最低接受质量时的速率为 W_{\min} , 把拥塞状态下的速率窗口随时间的变化函数设计为

$$W(t) = W_0 \cdot a^t + W_{\min} \quad (2)$$

表现在控制函数上, 也相应地在拥塞状态下采用乘性降低的办法(其中, D 表示减少; I 表示增加):

$$D : W_{t+\delta t} = a(W_t - W_{\min}) + W_{\min} \quad 0 < a < 1 \quad (3)$$

为了保持 TCP 友好性, 在非拥塞状态下, 也采用最大值限定加性增加的办法进行控制:

$$I : W_{t+RTT} = \min\{W_{\max}, W_t + b\} \quad b > 0 \quad (4)$$

但该控制模式的速率曲线仍然是上升的直线, 使之会停止在速率最大值 W_{\max} 上, 也不能达到初始快速上升、接近 W_{\max} 时缓慢上升的要求。进一步将上升曲线函数中的上升步幅由常量设计为变量, 其变化的幅度和它当前与最大值 W_{\max} 的差相关的设计为

$$I : W_{t+RTT} = W_t + \beta(W_{\max} - W_t) \quad 0 < \beta < 1 \quad (5)$$

式(5)具有与式(3)类似相反的效果。即在初始上升的阶段, 由于 $(W_{\max} - W_t)$ 较大, 因此上升速度较快, 而后由于 $(W_{\max} - W_t)$ 的变小, 上升速度变慢, 只要 β 的取值合适, 即可达到理想曲线的效果。由于在理论上仍然是乘性减少加性增加, 因此通过改进的AIMD控制算法可以保持TCP友好, 即协议间公平性。但改进的式(3)和式(5)算法具有针对视频的平滑考虑,

比初始的AIMD具有振荡幅度较小, 控制更加平滑的速率曲线, 其速率曲线的比较如图 2 所示。

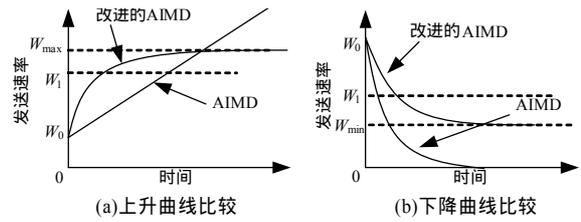


图 2 AIMD 与改进的 AIMD 速率变化曲线比

2 基于学习的扩展 AIMD 拥塞控制策略

定义 1 策略是一种或者多种规则, 描述当特定条件存在时的动作行为。

数学表示为: $Policy = \{Condition, Action\}$ 。

在 SB-EAIMD 中, $Condition$ 除了速率控制状态外, 还包含网络带宽变化趋势。

定义 2 网络带宽变化趋势(BT)为: $BT = \{0: \text{网络可用带宽稳定}; 1: \text{网络可用带宽增加或者未知}; -1: \text{网络可用带宽减小}\}$ 。由于对速率控制历史记录关心的并不是所有的速率控制状态, 而是其中一些比较有用的控制点状态, 因此需要定义一组有用的速率控制历史记录, 称为可用历史记录 VH , 它包含速率平均值 H 、上次拥塞状态下的速率控制状态 S_c 、上一次非拥塞状态下的速率控制状态 S_{nc} 。表述为: $VH = \{H, S_c, S_{nc}\}$ 。

SB-EAIMD 速率控制策略主要包括以下几个方面: VH 的更新; BT 和 $Direction$ 的检测; 新速率 $S.R$ 的设定。其中新速率 $S.R$ 的设定是最终目标。 VH 中的 H 主要用于统计的目的。描述如下:

(1) 初始化

初始化主要是初始速率 $S.R$ 的设定。把初始的 $S.R$ 设定为最低可接受速率 W_{\min} , 这样做的好处是以比较低的速率开始, 减少网络的大幅波动对其他用户的影响。初始 VH 的设置是速率控制波动范围, 初始的网络带宽变化未知。即: $S_c = \{W_{\max}, a, -1\}$; $S_{nc} = \{W_{\min}, \beta, 0\}$; $H = 0$; $S.R = W_{\min}$; $BT = 1$ 。

(2) VH 的更新

对于 S_c, S_{nc} 的更新比较简单, 分 2 种情况: 第 1 种是在网络可用带宽稳定的情况下, 如果当前速率控制状态 S 中 $Direction$ 为 0(表示未产生拥塞)并且 S_{nc} 中的 R 小于当前的 R , 则更新 S_{nc} 为当前的 S ; 如果 S 中 $Direction$ 为-1, 并且 S_c 中的 R 大于当前的 R , 则更新 S_c 为当前的 S 。 $Policy: Condition: (BT == 0) \text{ and } (S.Direction == 0) \text{ and } (S_{nc}.R < S.R) \rightarrow Action, S_{nc} = S; Condition: (BT == 0) \text{ and } (S.Direction == -1) \text{ and } (S_c.R > S.R) \rightarrow Action, S_c = S$ 。第 2 种情况是监测出网络可用带宽向某方向波动(一般情况是下降)的更新。如果监测到网络可用带宽下降, 意味着产生拥塞, 则将 $S_c.R$ 更新为最小可用带宽 W_{\min} ; 如果检测到网络可用带宽上升, 则可以将 $S_{nc}.R$ 设置为最大发送速率 W_{\max} 。 $Policy: Condition: (BT == 1) \rightarrow Action; S_{nc} = S; S_{nc}.R = W_{\max}; Condition: (BT == -1) \rightarrow Action: S_c = S; S_c.R = W_{\min}$ 。 H 也可以有 2 种更新方式, 第 1 种是从控制开始所有的速率控制中设定速率的平均值。假设当前为第 n 次调整, 其更新算式为: $H = ((n-1)H + S.R)/n$ 。第 2 种方式是记录前 $m(m < n)$ 次速率的平均值。要得到前 m 次速率的平均值, 显然至少需要 m 个空间存放速率调整的 R 的值历史记录。为了减少计算量, 可以采用替换法进行更新。

(3) 新速率 $S.R$ 的设定

速率 $S.R$ 的设定主要可以归纳为以下几条：

1)如果判断为网络可用带宽下降，则采用式(3)设定新的速率。此时新速率减小。

2)如果判断为网络可用带宽未知或者网络带宽上升，则采用式(5)设定新速率，此时新速率增加。

3)如果判断为网络带宽稳定的情况下，则新速率依照历史速率进行设置：

$$S.R = k \times S_{c.R} + (1-k) \times S_{nc.R} \quad (6)$$

其中， k 为权重调整因子， $0 < k < 1$ 。

(4)网络带宽变化趋势 BT 的判断

由新速率设置的策略可知， BT 值的判断至关重要。判断的策略如下：在初始阶段，网络可用带宽未知， $BT=1$ 。如果当前速率状态下检测到拥塞，并且当前速率 $S.R$ 小于速率控制状态历史记录 VH 中的拥塞速率记录 $S_{c.R}$ ，则判断为网络可用带宽下降， $BT=-1$ 。如果当前状态下没有检测到拥塞，并且当前速率 $S.R$ 大于速率控制状态历史记录 VH 中的非拥塞速率记录 $S_{nc.R}$ ，则判断为网络可用带宽上升， $BT=1$ 。如果当前速率 $S.R$ 在速率控制状态历史记录 VH 中的拥塞速率记录 $S_{c.R}$ 和非拥塞速率记录 $S_{nc.R}$ 的区间中，则判断为网络稳定， $BT=0$ 。

(5)网络拥塞的判断

网络拥塞的状态是速率控制的重要参数。判断网络拥塞状态可以有 2 种参数：延时和丢包。网络延时可用下式表示：

$$T = T_s + T_p + \sum T_r + T_d \quad (7)$$

其中， T_s 为本地主机数据处理延时； T_p 为物理线路上信号传输延时； $\sum T_r$ 为中间节点数据处理延时总和； T_d 为远程主机数据处理延时。

网络中的延时主要部分是 $\sum T_r$ ，特别是在网络拥塞状态下，网络中瓶颈节点的排队可以引起较大的延迟。由于实时视频流通常采用UDP作为底层传输协议，并不针对每个数据包报送反馈信息，无法简单确定往返延迟 RTT 。解决的方法是可以采用时间服务器进行同步，然后使用单向网络延时，式(7)中的 T 作为判断标准。接收端向源端发送相应的反馈信息，源端根据反馈信息，可以按照以下公式计算下一时刻的 T_{RTT} ：

$$T_{RTT} = a \cdot t_{RTT} + (1-a) \cdot (now - ST_1 - PT) \quad (8)$$

其中， t_{RTT} 为当前的往返传输时间； a 为常数，通常取 0.75； now 指根据时间服务器同步得到的当前时间； ST_1 发送端发送数据包的时间； PT 为接收方应答的时间间隔。反馈的ACK信息除了 ST_1 和 PT 外还可以包含数据包丢失率以及接收方数据速率等。另一种方法是在接收端测试相对延迟，即对数据进行RTP封装时，都有相应的同步字和时间戳，如果未接收到下一个包的时间过长，则表示下一个包已经丢失或者延迟过大，可以利用RTCP反馈相关的信息得到相对延迟。对于实时视频传输来说，最大的单向延时 T 容忍度通常为 300 ms。在局域网内，没有拥塞的状态下延迟通常远小于 100 ms。

通过测试表明，局域网内拥塞状况下的延时有可能达到 500 ms。可以设置一个以延迟阈值 T_{ph} 作为参数来判断网络拥

塞状态。在局域网的情况下 $T_{ph}=200$ ms，而在广域网中一般设置为 $T_{ph}=300$ ms。采用丢包率的 P_{loss} 来表示网络拥塞状况也是可行的。数据包在网络中传输丢失与否可以通过成功与失败 2 个状态的Markov链来表示。Markov链中存在 2 种状态。其中， R 状态表示数据包传输成功； L 状态表示数据包传输失败，即数据包丢失； p 和 q 表示 2 种状态转换的概率。设系统处于 R, L 状态的时间分别为 T_R 和 T_L ，根据Markov链，可以求得系统处于状态 L 的概率为

$$P_L = T_L / (T_L + T_{LR}) = q / (p+q) \quad (9)$$

其中，参数 p 和 q 可以在接收端通过对接收数据包的统计和分析得到。由于RTP封装的视频数据包包含时间戳和同步标志，根据数据包之间的时序信息和包依赖关系即可以统计出当前的 p 和 q 。这种计算丢失率的办法具有一定的局限性，由于 p 和 q 是对丢失事件历史的统计，有时并不能正确反映当前网络的最新变化情况。如拥塞状况改善时，虽然其概率 p 和 q 是也会随之变化，但变化常常滞后于网络拥塞状况的变化。为此，可以采用一种对历史丢失率进行指数加权滑动平均的方式来进行计算：定义丢失间隔 s 为 2 个丢失事件间的传输分组数目(包好第 2 个丢失分组)，则分组丢失率为 $p=1/s$ ，对丢失间隔进行指数加权滑动平均如下：

$$S \leftarrow (1-\eta) \cdot S + \eta \cdot s_{rec} \quad \eta \in [0,1] \quad (10)$$

其中， S 为平均丢失间隔； s_{rec} 为最近的丢失间隔。采用这种方式时， η 的取值对计算结果的影响很大。如果取值大了，短暂的拥塞状况即产生过于剧烈的反馈效应，不利于速率控制的平滑。如果过小了，又不能灵敏反映当前网络变化的情况。一种改进的方式是加权平均丢失间隔，对丢失历史进行分段加权的方法来实现。在这种方式中，离当前时间越接近的丢失率，权重比重越大。

3 结束语

本文针对 Internet 中点对点的多媒体通信系统提出一种以丢包为参数基于学习的扩展 AIMD 拥塞控制机制(SB-EAIMD)，解决网络拥塞控制和 QoS 控制等多媒体通信技术问题，并研究和开发了基于 Internet 网络视频平台，具有一定的实用价值。

参考文献

- [1] Floyd S, Fall K. The Use of End-to-end Congestion Control in the Internet[J]. IEEE/ACM Transaction on Networking, 1997, 7(4): 458-472.
- [2] 林 闯. 多媒体信息网络 QoS 的控制[J]. 软件学报, 1999, 10(10): 1016-1024.
- [3] Tan Waitian. Real-time Internet Video Using Error Resilient Scalable Compression and TCP-friendly Transport Protocol[J]. IEEE Transactions on Multimedia, 1999, 1(2): 172-186.
- [4] Kaneko H, Stankovic J A. Integrated Scheduling of Multimedia and Hard Real-time Tasks[C]//Proc. of the Real-time System Symposium. Washington D. C., USA: [s. n.], 1996: 206-217.