THE PROTOCOL MODEL FOR THE CONGESTION CONTROL

Ioan POPOVICIU

Naval Academy "Mircea cel Bãtrân", Constantza, Romania

Abstract: *The current Internet allows applications to use the network with arbitrary data rates and congestion response, potentially in a harmful way. Protection of the public network may not be a practically important problem when the majority of Internet applications uses TCP. However, it becomes serious with the growth of delay sensitive applications such as streaming media, which often prefer UDP over TCP as their transport protocol choice. A non-TCP protocol is called TCP-friendly when it yields the same throughput as traditional TCP. TCP-friendly protocols are generally used for multimedia/real-time applications. This paper proposes a TCP-friendly protocol model for the streaming media based on Additive Increase / Multiplicative Decrease Control algorithm (AIMD) .*

Key words: Internet, *TCP, congestion, protocol, network.*
MSC 2010: *68M10.*

1. Introduction

Network congestion is characterized by presence of a large number of packets (load) being routed in all or portions of the subnet that exceeds its link and router capacities (resources) resulting in a performance slowdown

A network is considered congested when too many packets try to access the same router's buffer, resulting in an amount of packets being dropped. In this state, the load exceeds the network capacity. During congestion, actions need to be taken by both the transmission protocols and the network routers in order to avoid a congestion collapse and furthermore to ensure network stability, throughput efficiency and fair resource allocation to network users. Indeed, during a collapse, only a fraction of the existing bandwidth is utilized by traffic useful for the receiver. Congestion collapse is considered, in general, as a catastrophic event. However,

congestion itself is associated with different properties, depending on the characteristics of the underlying networks, the mechanisms of the transmission protocols, the traffic characteristics of the contenting flows, the level of flow contention, and the functionality of network routers. Therefore, the impact of congestion may be temporary and easily controllable; or it may be catastrophic. Consider, for example, a high speed network which hosts a number o competing flows that increases or decreases. The window of each flow also increases and decreases. However, unlike the traditional networks, the time it takes for the flows to exploit the available bandwidth is certainly longer; the amount of loss upon congestion is certainly higher; and the duration of congestion itself throughout the overall communication time may be relatively smaller. Since the nature of acceptable congestion cannot be prescribed or even accurately defined in general, congestion control becomes a complex task. Furthermore,

complexity increases due to the multipurpose-task of congestion control algorithms. They need to control congestion and avoid collapses, maximize bandwidth utilization, guarantee network stability, and ensure fair resource allocation. Considering the network as a black box that only provides a binary feedback to network flows upon congestion, shifts all the burden to end users and calls for solutions that are more generic and perhaps less responsive. That is, a binary congestion signal does not reflect the particular network state. Each sender operates independently and goals to adjust its rate (or window) in a manner that the total bandwidth of the network will be expended fairly and effectively.

For congestion detection we can utilize two technique:

- Notification from packet switches (routers).
- Infer congestion from packet loss:
    o Packet loss can be used to detect congestion because packet loss due hardware failure is very rare.
    o Sender can infer congestion from packet loss through missing acknowledgments.
    o Rate or percentage of lost packets can be used to gauge degree of congestion.

Congestion control methods:

- Traffic Shaping:
    o Heavily used in VC subnets including ATM networks.
    o Avoid bursty traffic by producing more uniform output at the hosts.
    o Representative examples:Leaky Bucket, Token Bucket.
- Admission Control:
    o Used in VC subnets.
    o Once congestion has been detected in part of the subnet, no additional VCs are created until the congestion level is reduced.
- Choke Packets:
    o Used in both datagram and VC subnets.
    o When a high level of line traffic is detected, a choke packet is
    o Sent to source host to reduce traffic.
    o Variation Hop-by-Hop choke packets.
- Load Shedding:
    o Used only when other congestion control methods in place fail.
    o When capacity is reached, routers or switches may discard a number of incoming packets to reduce their load.

2. Goals and metrics

The congestion window determines the number of packets that can be outstanding at any time. That is, the number of packets that can be sent without having received the corresponding ACK packets. It is incorporated into the transport layer and controls the number of packets put into the network. The rate describes packets per second or bits per second. The window or rate can be dynamically adjusted as the total load on the system changes, however, the former is strictly based on ACKs. A cycle is the phase between two seriate feedbacks of 1 (indicating congestion). Hence, a cycle consists of one decrease step triggered by congestion and a number of additive increase steps. A step describes a single window adjustment in response to a single feedback (either 0 or 1). The system is in an equilibrium state, when resource usage of all flows in a bottleneck is balanced. AIMD-based congestion control algorithms guarantee convergence to equilibrium [2]. In congestion avoidance algorithms this is not always guaranteed. A non-TCP protocol is called TCP-friendly when it yields the same throughput as traditional TCP. TCP-friendly protocols are generally used for multimedia/real-time applications. Although the sources might discover their fair-

"HENRI COANDA"
AIR FORCE ACADEMY
ROMANIA

GERMANY

"GENERAL M.R. STEFANIK"
ARMED FORCES ACADEMY
SLOVAK REPUBLIC

INTERNATIONAL CONFERENCE of SCIENTIFIC PAPER
AFASES 2011
Brasov, 26-28 May 2011

share early on, the dynamics of real systems in practice prohibit a straightforward adjustment, but instead, they call for continuous oscillations as a means of discovering the available bandwidth. The metrics for the system performance are as follows:

• **Efficiency**: Efficiency is the average flows throughput per step (or per RTT- round-trip time), when the system is in equilibrium.

• **Fairness**: Fairness characterizes the fair distribution of resources between flows in a shared bottleneck link. A well-known metric is: $F(x) = \dfrac{\sum (x_i)^2}{\sum (x_i^2)}$. This index is bounded between 0 and 1.

• **Convergence Speed**: Convergence speed describes time passed till the equilibrium state.

• **Smoothness**: Smoothness is reflected by the magnitude of the oscillations during multiplicative decrease. It depends on the oscillations size.

• **Responsiveness**: Responsiveness is measured by the number of steps (or RTTs-round-trip time) to reach an equilibrium (i.e., to equate the windows in order to be in a fair state).

The difference between Responsiveness and Convergence Speed is that the former is related to a single flow and the latter to the System.

Goals in the evaluation process of a congestion avoidance/control algorithm are:

• To achieve high bandwidth utilization.
• To converge to fairness quickly.
• To minimize the amplitude of oscillations.
• To maintain high responsiveness.
• To coexist fairly and be compatible with traditional widely-used (AIMD based) protocols.

3. TCP Congestion Control

TCP uses a form of end-to-end flow control. In TCP, when a sender send a packet, the receiver acknowledges receipt of the packet. A sending source can use the acknowledgement arrival rate as a measure of network congestion. When it successfully receives an acknowledgment, a sender knows that the packet reached its destination. The sender can then send new packets on the network. Both the sender and the receiver agree on a common window size for packet flow . The window size represents the number of bytes that the source can send at a time. The window size varies according to the condition of traffic in the network to avoid congestion . Generally, a file of size f with a total transfer time of Δ on a TCP connection results in a TCP transfer throughput denoted by **r** and obtained from equation $r = f / \Delta$.

We can also derive the bandwidth utilization, P_u, assuming that the link bandwidth is B, by equation $P_u = r / B$.

TCP has three congestion-control methods: **additive increase, slow start,** and **retransmit.** Chiu and Jain [2] have formulated the congestion avoidance problem as a resource management problem and proposed a distributed congestion avoidance mechanism named 'additive increase/multiplicative decrease' (AIMD). In their work, as a network model they use a "binary feedback" scheme with one bottleneck router.
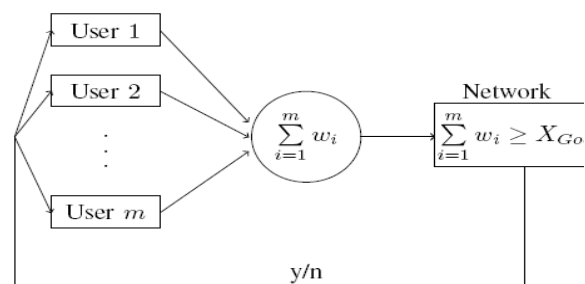
Synchronous control system model of m users sharing a network.

It consists of a set of m users each of which send data in the network at a rate $2w_i$. The data send by each user are aggregated in a single bottleneck and the network checks whether the total amount of data send by users exceeds some network or bandwidth threshold *goal X* (we can assume that *goal X* is a value between the knee and the cliff and is a characteristic of the network). The system sends a binary feedback to each user telling whether the flows exceed the network threshold. The system response is 1 when bandwidth is available and 0 when bandwidth is exhausted. The feedback sent by the network arrives at the same time to all users. The signal is the same to all users and they take the same action when the signal arrives. The next signal is not send until the users have responded to the previous signal. Such a system is called synchronous feedback system or simply synchronous system. The time elapsed between the arrival of two consecutive signals is discrete and the same after every signal arrival. This time is referred also as RTT. The system behavior can be defined the following time units: a step (or round-trip time – RTT) is the time elapsed between the arrival of two consecutive signals. A cycle or epoch is the time elapsed between two consecutive congestion events (i.e., the time immediately after a system response 0 and ending at the next event of congestion when the system response is again 0). This network model is quite simple and its assumptions have been evaluated in the Internet for several years. In practice the parameter *goal X* is the network capacity (i.e. the number of packets that the link and the routers' buffer can hold – or in-the-fly packets). When the aggregate flows' rate exceeds the network capacity the flows start to lose packets. If the transport protocol provides reliability mechanisms (e.g. as in TCP) it can detect the packet loss or congestion event. Since the majority of the applications use reliable transport protocols (e.g. TCP), the binary feedback mechanism has an implicit presence: a successful data transmissionis interpreted as available bandwidth, and a packet loss is interpreted as congestion event . Although the system had a

strong impact on the evaluation of congestion avoidance mechanisms (e.g. AIMD), there are some limitations. First, the system considers the responses to be synchronous, which, in terms of real networks means that all flows have the same RTT. This assumption is not real. A second assumption and limitation is that the network response arrives at the same time to all users, even when they have the same RTT. The above assumption is supported by Jacobson experimentally in a low bandwidth network with congestion avoidance mechanisms (TCP-Tahoe) and where flows have the same RTT . Whatever the argument, this assumption is not true for a reason which is the third limitation of the system. The system has only one bottleneck. In reality a connection might go through none, one, or more than one router or bottlenecks. If a flow traverses more than one bottleneck, then it is not guaranteed that at each bottleneck congestion will happen at the same time. Nevertheless, these limitations do not prevent the mechanisms from controlling flows' data rate and avoid congestion which was the major concern in the early stages of the Internet .

4. Additive Increase / Multiplicative Decrease Control algorithm (AIMD) and TCP-friendly protocol for the streaming media

The basic idea of the algorithms to reduce the sending rate/window of the flows when the system bandwidth is exhausted and to increase the sending rates/windows when bandwidth is available. As mentioned in the previous section, when bandwidth is available (i.e. the aggregate rates of the flows do not exceed the network threshold: $\sum_i w_i < goal\ X$) the system attaches the signal 1 to the acknowledgment of each packet. In response, flows increase by one (packet) their windows. A continuous series of positive signals will cause a linear increase in the flows' rate. Obviously, the increase is not unlimited because the bandwidth is fixed. When flows' rate exceed the bandwidth limit (i.e. $\sum_i w_i \geq goal\ X$) the system attaches the 0 signal to the

"HENRI COANDA"
AIR FORCE ACADEMY
ROMANIA

GERMANY

"GENERAL M.R. STEFANIK"
ARMED FORCES ACADEMY
SLOVAK REPUBLIC

INTERNATIONAL CONFERENCE of SCIENTIFIC PAPER
AFASES 2011
Brasov, 26-28 May 2011

acknowledgment of each packet and flows respond to congestion by a decrease in their sending rates/windows. A. Lahanas and V. Tsaoussidis prove that a linear increase/exponential decrease policy is a condition for the increase/decrease algorithms to set (or converge) quickly the system in a fair state where the load oscillates around some equilibrium. The equilibrium state determines also the fairness and efficiency of the mechanism.

The TCP congestion control is classified as Additive- Increase Multiplicative-decrease (AIMD) mechanism. Following the notation AIMD(a, b), TCP is AIMD(1, 1/2). The parameter a represents the factor to be added to the congestion window each round trip time in absence of congestion, that is congestion_window + a. On the other hand, the parameter b represents the complement to 1 that should be multiplied to the congestion window when congestion is detected, that is (1-b) congestion window.

An AIMD control algorithm may be expressed as:

Increase: $W_{t+R} \leftarrow W_t + a$, $a > 0$

Decrease: $W_{t+\delta t} \leftarrow (1 - \beta)W_t$, $0 < \beta < 1$

A generalizations of AIMD is binomial control:

Increase: $W_{t+R} \leftarrow W_t + \dfrac{a}{W_t^k}$, $a > 0$

Decrease: $W_{t+\delta t} \leftarrow W_t - \beta W_t^l$, $0 < \beta < 1$

where "Increase" refers to the increase in window as a result of the receipt of one window of ACKs within a single RTT, "Decrease" refers to the decrease in window upon detection of congestion by the sender, Wt the window size at time t, R the flow's RTT, and a, b, k, l are constants. For example, for k=0, l=1 we get AIMD.

AIMD and binomial controls are memoryless since the increase and decrease

rules use only the current window size w_t and constants (α, β, k and l). The window size at the end of the last congestion epoch is useful, not only as an indicator of the current congestion level of the network, but also as a good predictor of the congestion state for the next sequence. Thus, our proposed scheme maintains such a state variable w_{max}, which is updated at the end of *each* congestion sequence. In addition, let w_0 denote the window size after the decrease. Given a decrease rule, w_0 can be obtained from w_t, and *vice versa*. For example, for AIMD, $w_0 = (1 - \beta)w_{max}$. Henceforth, for clarity, we use both w_{max} and w_0 .

We propose to adopt the following window increase function:

$$w(t) = w_0 + c * t^u, u, c > 0 \qquad (1)$$

where $w(t)$ is the continuous approximation of the window size at time t (in RTTs- round-trip time) elapsed since the window started to increase. By definition, $w(0) = w_0$. This window increase function is equivalent to the following window increase rule:

$$w_{t+1} \leftarrow w_t + \alpha / (w_t - w_0)^k, \alpha > 0 \qquad (2)$$

where $k > -1$ and α is independent of t. In particular, $u = 1/(k + 1)$ and $c = ((k + 1)\alpha)^u$. We are interested in congestion control schemes that have various window size increase patterns (different $u's$, or equivalently, different $k's$). Consider three cases. First, if $-1 < k < 0$, the congestion window increases super-linearly. The window is increased cautiously just after the detection of packet loss, and the increase becomes more and more aggressive when no more loss occurs. Second, if $k = 0$, the window increases linearly, i.e., additive increase. The aggressiveness does not change with time. Third, if $k > 0$, the

window increases sublinearly. The connection approaches the previously probed window size fast, but it becomes less aggressive beyond that. These various schemes possess different degrees of aggressiveness, and may satisfy different applications. For example, super-linear increase can support applications that need to quickly acquire bandwidth as it becomes available.

Therefore, we consider the following control rules:

Increase: $w_{t+1} \leftarrow w_t + \alpha(w_{max})/(w_t - w_0)^k$, (3)
$\alpha(w_{max}) > 0$

Decrease: $w_t \leftarrow w_t - \beta w_t^l$, $0 < \beta < 1$

Note that we write α as a function of w_{max} since this is required in the derivation of TCP-friendliness. In the remainder of this paper, we simply write α for clarity. We use the same decrease rule as binomial controls. For the increase rule, we consider $k > -1$, since otherwise the window size increases exponentially or faster and we consider it unstable. For the decrease rule, we consider $l \leq 1$, since otherwise $(w_t - \beta w_t^l)$ can be negative when w_t is large enough.

We show that this control can be TCP-friendly by appropriately defining α as a function of the constant β and the state variable w_{max}. This control is radically different from binomial controls, because binomial controls generalize AIMD, but they are still in the memoryless space.

We show that this control scheme using the control rules in (3) can be TCP-friendly. The notion of TCP-friendliness refers to the relationship between throughput and packet loss rate. We consider a random loss model, where the losses are Bernoulli trials; packets are dropped uniformly with a fixed probability, and following definition of α to make congestion control scheme TCP-friendly:

When the window size variation is small, i.e., the window decrease is small, $\beta w_{max}^l \leq w_{max}$, we can simplify α and c as

$$\alpha \approx \frac{3}{2(k+1)} \left(\frac{\beta}{\Gamma\left(\frac{1}{k+1}+1\right)} \right)^{k+1} w_{max}^{kl+l-1} \quad (4)$$

$$c \approx \left(\frac{3}{2}\right)^{\frac{1}{k+1}} \frac{\beta}{\Gamma\left(\frac{1}{k+1}+1\right)} w_{max}^{l-\frac{1}{k+1}} \quad (5)$$

where the Gamma function $\Gamma(.)$ is a constant, α is a constant factor of w_{max}^{kl+l-1} and c is a constant factor of $w_{max}^{l-(l/k+1)}$. When $k = 0$ and $l = 1$, from (4) we have $\alpha_{AIMD} = 3\beta(2-\beta)$. If $\beta \leq 1$, $\alpha_{AIMD} \approx 3\beta/2$ it degenerates to the memoryless TCP-friendly AIMD control.

5. Conclusion

We proposed a TCP-friendly protocol model for congestion controls. They are TCP-friendly and TCP-compatible under queue management. They possess different smoothness, aggressiveness, and responsiveness tradeoffs. Thus, instances from this applications can be chosen as the transport schemes of various applications, for example, streaming applications on the Internet which are required to be TCP-friendly.

REFERENCES

[1] D. Bansal and H. Balakrishnan, *"Binomial Congestion Control Algorithms"*, in Proc. IEEE INFOCOM'01, Anchorage, Alaska, USA, April 2001.

[2] D. Chiu and R. Jain, *"Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks"*, Journal of Computer Networks and ISDN, vol. 17, no. 1, June 1989, pp. 1–14.

[3] S. Jin, L. Guo, I. Matta, and A. Bestavros, *"TCP-friendly SIMD Congestion Control and Its Convergence Behavior"*, in Proc. ICNP'2001, Riverside, USA, November 2001.

[4] R. Karp, E. Koutsoupias, C. Papadimitriou, and S. Shenker, *"Optimization Problems in Congestion Control"*, in Proc. IEEE Symposium on Foundations of Computer Science, Redondo Beach, USA, November 2000.

[5] A. Lahanas and V. Tsaoussidis, *"Additive Increase Multiplicative Decrease - Fast*

"HENRI COANDA"
AIR FORCE ACADEMY
ROMANIA

GERMANY

"GENERAL M.R. STEFANIK"
ARMED FORCES ACADEMY
SLOVAK REPUBLIC

INTERNATIONAL CONFERENCE of SCIENTIFIC PAPER
AFASES 2011
Brasov, 26-28 May 2011

Convergence (AIMD-FC)", in Proc. Networks 2002, Atlanta, USA, August 2002.

[6] A. Lahanas and V. Tsaoussidis, "*Exploiting the Efficiency and Fairness Potential of AIMD-based Congestion Avoidance and Control*", Computer Networks, Elsevier, vol. 43, no. 2, October 2003, pp. 227–245.

[7] A. Lahanas and V. Tsaoussidis, "*_-AIMD for Asynchronous Receiver Feedback*", in Proc. IEEE ISCC 2003, Antalya, Turkey, June 2003.

[8] S. Low, F. Paganini, and J. C. Doyle, "*Internet Congestion Control*", IEEE Control Systems Magazine, vol. 22, 2002.