

Enhancements on Adaptive Congestion Protocol for Wireless Networks

Pengxuan Mao^{1, a *}, Lixiang Liu^{2, b}

^{1,2}Science and Technology on Integrated Information System Laboratory,

Institute of Software Chinese Academy of Sciences,

Beijing, China

^ampxbs@126.com

Keywords: ACP; WLAN; Protocol design

Abstract. In high Bandwidth Delay Product (BDP) networks, Adaptive congestion Control Protocol (ACP) is emerging as one potential solution for overcoming limitations inherent to the current TCP algorithm, characterized by unstable throughput, high queuing delay, RTT-limited fairness. However, its performance in wireless network is deeply not investigated. In this paper, we address the challenges when applying ACP to wireless networks, and extend ACP, called iACP to enhance its performance in wireless networks. ns2 simulation results demonstrate that iACP can fairly and efficiently allocate bandwidth in homogenous and heterogeneous network.

Introduction

TCP is inefficient and instable in high bandwidth-delay product environments [1]. TCP leads to throughput oscillation and link under-utilization, especially in a high bandwidth-delay-product environment. To address these problems, researchers have proposed end-to-end schemes with explicit feedback to overcome the limitations of TCP [1-6]. Among them, adaptive congestion protocol (ACP) [5] is a promising end-to-end scheme that is characterized by max-min fairness, high-utilization, small queue sizes and no observable packet drops. With the help of explicit feedback from routers to end hosts, ACP is able to satisfy all design requirements. However, we note that ACP's performance in wireless network is still not deeply investigated. For a wireless link it is very hard to do such kind of setting. ACP is unable to maintain fairness and stability with improper estimation of the link capacity parameter. So, we propose an improved version of ACP (iACP). This scheme can ensure that different flows share the 802.11 bandwidth, only make little modifications.

The rest of the paper is organized as follows. Section II, we describe the motivation of our research in detail. Section III elaborates on iACP in detail. In Section IV, we present stability conditions for iACP system. Section V shows NS2 simulation results, we draw the conclusions.

IACP ALGORITHM

The Congestion Header. In a way similar to ACP [5], the iACP packet carries a congestion header which consists of three fields. The field $H - cwnd$ is the sender's current congestion window, whereas $H - rtt$ is the sender's current RTT estimate. These are filled in by the sender and never modified in transit.

The $H - feedback$ field carries the target congestion window size which the AP or routers computes. This field is initiated by the sender and is then updated by each link the packet encounters in its path. At each link, the value in the field is compared with the target congestion window.

The iACP header. As similar to ACP[5], iACP need maintain two values as follows: $cwnd$ and rtt . $cwnd$ denotes the current congestion window size. rtt represents the estimation of the current round trip time. On packet departure, the sender attaches a congestion header to the packet and sets the

H_cwnd field to its current $cwnd$ and H_rtt to its current rtt . In the first packet of a flow, H_rtt is set to zero to indicate to the AP that the source does not yet have a valid estimate of the round trip time rtt .

The congestion window $cwnd$ is updated every time the sender receives an acknowledgement. When a new acknowledgement is received, the value in $H_feedback$ field, which represents the target congestion window size requested by the network in packet per second, is read and is used to update the congestion window $cwnd$.

The desired target congestion window is the new congestion window requested by the network. We do not immediately set the $cwnd$ equal to the desired congestion window because this abrupt change may lead to burst traffic. The congestion window is updated according to the following equation:

$$cwnd = cwnd + a * (desired_window - cwnd) \quad (1)$$

where a is configurable parameter.

The iACP Receiver. The iACP receiver is identical to the other explicit control protocol. When acknowledging a packet, it copies the congestion header from the data packet to its acknowledgment.

The iACP AP and Router. Different from the weighted window method [7] that limits the size of the receiver window of TCP flows, the desired target congestion window is calculated as following equation:

$$desired_window = desired_window + \frac{b * g * (buffer - q_{avg})}{N} \quad (2)$$

where $b, g \in (0, 1)$ are configurable parameters, $buffer$ denotes the maximum queue size maintained by the AP. we actually use the weighted moving average of q to smooth out possible oscillation caused by packet bursts:

$$q_{avg} = w * q + (1 - w) * q_{avg} \quad (3)$$

where w is 0.2. q_{avg} is the average queue size in the AP. N is the number of all the flows. Similar to the method [8], the number of flows can be identified by IP addresses. The $desired_window$ calculated at AP is used to update the $H_feedback$ field in the packet congestion header as following:

$$H_feedback = desired_window \quad (4)$$

On packet departure, the AP compares the desired congestion window with the value stored in the field $H_feedback$ and updates the field with the minimum value.

Stability Analysis

In this section, we will investigate the stability of the proposed system, and develop the guidelines of the parameter configuration. First, we model the iACP system as follows:

$$\begin{cases} W(k + R_0) = W(k) + \frac{b * g * (buffer - Q(k))}{N} \\ - \frac{N}{R_0} W(k) + Q(k) = - C \end{cases} \quad (5)$$

where $W(k)$ denotes the congestion window (packets). $Q(k)$ denotes the queue length (packets), C denotes the link capacity (packets/sec), N represents the number of all the flows, $buffer$ denotes the buffer size in the AP, and R_0 is the round trip time in equilibrium state. It is defined as following:

$$R_0 = \frac{q_0}{C} + T_p \quad (6)$$

where $q_0 = g * buffer$, and Applying z-transform to (7), we can have

$$W(z) + \frac{b}{N(z^{R_0} - 1)} Q(z) = \frac{b \text{ buffer}}{N(z^{R_0} - 1)} \quad (7)$$

$$- \frac{N}{R_0} W(z) + Q(z) = -C$$

So, we have the characteristic polynomial of the system (7) as following:

$$T(z) = \det \begin{bmatrix} 1 & \frac{b}{N(z^{R_0} - 1)} \\ \frac{N}{R_0} & 1 \end{bmatrix} = 1 + \frac{N}{R_0} \frac{b}{N(z^{R_0} - 1)} \quad (8)$$

Let $T(z) = 0$, we obtain the pole $z^{R_0} = 1 - \frac{b}{R_0}$. If the system (5) is stable, the following condition $|z| \leq 1$ must be satisfied. Then we have

$$b \leq 2R_0 \quad (9)$$

Performance Simulation

The Basic Simulations Setup. Our NS2[9] topology as shown in Fig. 1, the network is composed of one AP and the number of wireless nodes, each of which has different number of uplink and downlink flows with the wired nodes. AP is connected to a router which is a gateway to fixed nodes. The capacity of each wired link was set to 1000Mbps, which is much higher than 11Mbps, the capacity of IEEE 802.11. The propagation delay between the router and the AP was set to 2ms, and those of the other wired links were set to be 20ms. The AP employed DropTail with the buffer size of 50 packets. The number of flows is equal to 12. The maximum RTT of TCP is set to 2 seconds. The data and basic rates of the stations are set to 11 Mbps and 1 Mbps respectively. RTS/CTS is enabled. Packet size is set to 1300 byte. Queue length is 49 packets. Each source only sends a flow. a, b, g are set to be 0.012, 0.012, and 0.99 respectively.

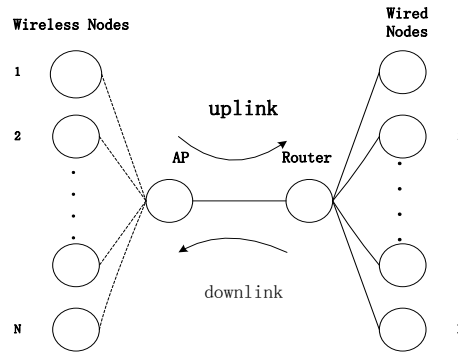


Fig. 1 Network Topology

Performance in the presence of Heterogenous RTT. In this part, the parameter configuration is as following: The delay of the links between the wired nodes and the base station are set to be (20, 30, ..., 130) ms.

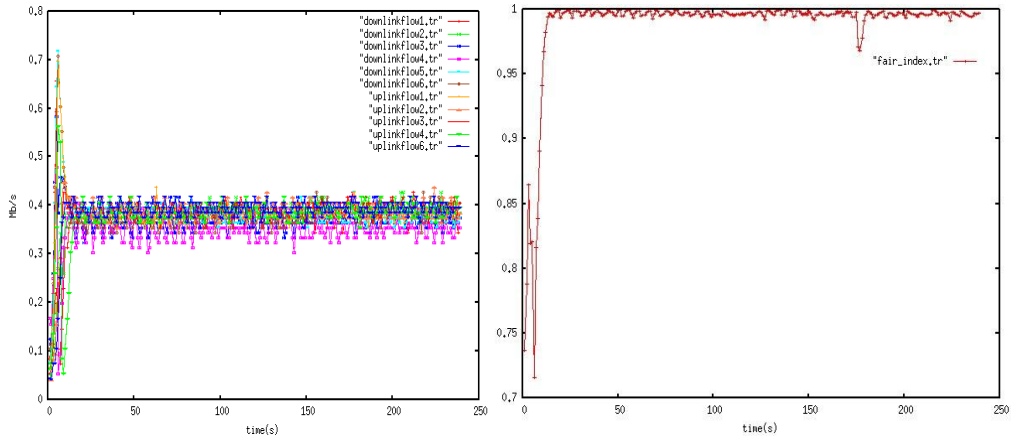


Fig. 2 the throughput (heterogeneous) Fig. 3 the fairness index (heterogeneous)

Fig. 2 demonstrates that iACP can achieve fairness throughput. Fig. 3 shows that the fairness index's value is almost one in all the simulation time.

Impact of Bottleneck Capacity or Feedback Delay. The single link capacity is from 30Mb/s to 210Mb/s. The delay between the router and the AP was set to 2ms, and the other links were set to be 20ms. The round-trip time is from 40ms to 240ms. Fig. 4, 5 show that iACP can achieve high fairness index as link capacity and delay increases.

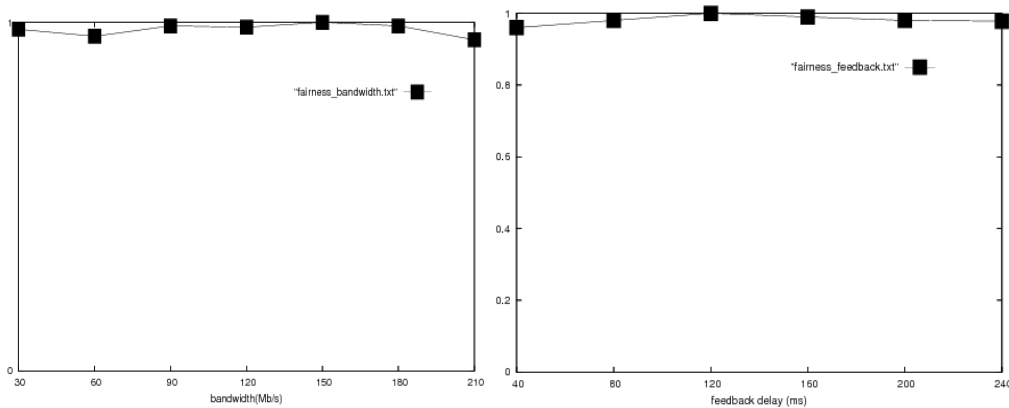


Fig. 4 Fairness vs bandwidth capacity Fig. 5 Fairness vs RTT

Comparison with Other Algorithms. In this scenario, the basic parameter configuration is as follows: The wired links between the router and the base station have a capacity of 40Mbit/s and 2 ms delay. There are 12 uplink flows in the scenario. As shown in Fig. 6, iACP can attain higher fairness index than XCP-b. The results in Fig. 7 show iACP achieves a more stable throughput than XCP-b. This is because that XCP-b's spare bandwidth estimation is based on the variation of instantaneous queue size.

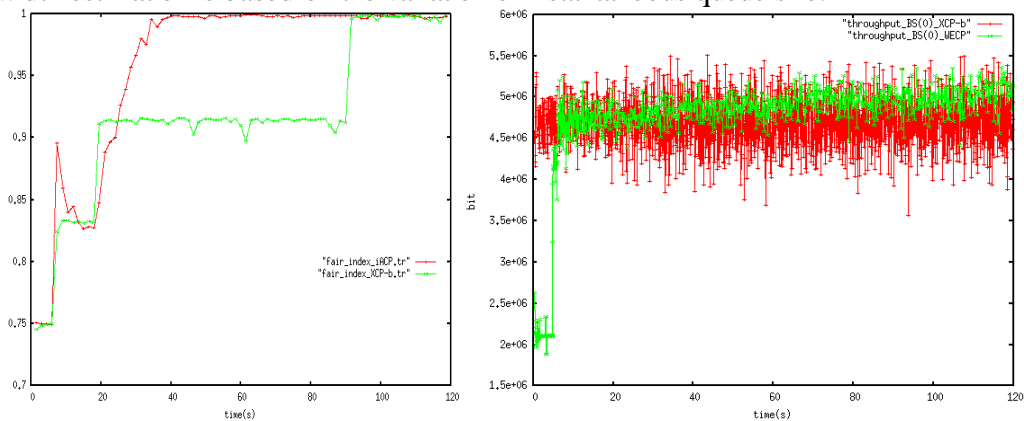


Fig. 6 the fairness comparison

Fig. 7 the throughput comparison

Summary

In this paper, we propose an improved version of Adaptive Congestion Protocol (iACP) which is designed for wireless network. We also give the stability condition for the iACP system. Our simulation results show that the novel protocols can enhance the performance of ACP in high BDP environments. iACP can fairly and efficiently allocate bandwidth in heterogeneous network.

Acknowledgement

This research is sponsored by the National Science Foundation of China under Grant No.91438119 .

References

- [1]. D. Katabi, M. Handley, C. Rohrs, Congestion control for high bandwidth-delay networks. In Proc. of Sigcomm, 2002.
- [2]. Y. Xia, L. Subramanian, I. Stoica, S. Kalyanaraman, "One more bit is enough", IEEE/ACM Transaction on networking, vol.16.no.6 2008.
- [3]. C.Jin, D.Wei, and S.Low, FAST TCP:Motivation, Architecture, Algorithms, Performance. INFOCOM 2004.
- [4]. N.Dukkipati, Rate Control Protocol (RCP): Congestion control to Making Flows Complete Quickly, Ph.D thesis, Department of Electrical Engineering, Stanford University, 2008.
- [5]. Marios Lestas, Adaptive congestion protocol: A congestion control protocol with learning capability, Computer Networks, 51 (2007) 3773-3798.
- [6]. Huangxing Wu, Fengyuan Ren, Wengping Pan, Yadong Zhai, Analysis of efficient and fair explicit congestion control protocol with feedback delay: Stability and convergence, Computer Communications, vol.33, pp.1992-2000, 2010.
- [7]. Seyedzadegan, Mojtaba; Othman, Mohamed, "Weighted window method for TCP fairness in WLANs", Information Technology, 2008. International Symposium on, vol.4, no. pp.1-7, 26-28. Aug. 2008.
- [8]. S. Pilosof, D.R.R. Ramjee, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN", IEEE INFOCOM 2003, San Francisco, CA, USA, 2003, 863-872.
- [9]. "ns-2", <http://www.isi.edu/nsnam/ns/>.
- [10]. Filipe Abrantes, Manuel Ricardo, "Explicit congestion control algorithms for time varying capacity media", IEEE Trans Mobile Computing, vol.10, issue.1, January 2011.