Trade-off evaluation between fairness and throughput for TCP congestion control mechanisms in a wireless LAN environment

Masafumi Hashimoto, Go Hasegawa and Masayuki Murata
Osaka University, Japan

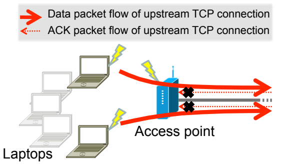
Outline

- Background
- TCP unfairness in WLANs
- Objectives of this work
- A transport-layer solution for alleviating TCP unfairness in WLANs
- New metric for evaluating a trade-off relationship between fairness and throughput
- Experimental evaluation
- Conclusion

Background

- Accessing the Internet through WLANs is becoming a common situation
  - Rail stations and airports and so on
  - Many wireless clients share one access point (AP)
- Many kinds of applications generate both upstream and downstream traffic
  - P2P file sharing and audio/video conference applications
- Problems
  - Fairness among users in WLANs
  - Trade-off relationships between fairness and bandwidth utilization

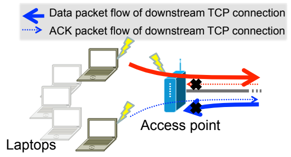
Two kinds of TCP unfairness in WLANs



Unfairness among upstream flows

- ACK packets of upstream TCP flows are discarded
  - Once timeout occurs in a certain flow, the flow maintains low transmission rate

Unfairness among upstream flows



Unfairness between upstream and downstream flows

- ACK packets of upstream TCP flows and data packets of downstream TCP flows are discarded
  - Only TCPs of downstream flows activate congestion control

Unfairness between upstream and downstream flows

Fairness index

- Jain's fairness index [7]

$$F_j(X) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2}$$

  - Evaluates fairness among users, which is independent of the scale of allocations
    - For example; three users are allocated the following values
      - Case 1: 1 Mbps, 2 Mbps, and 3 Mbps (the total is 6 Mbps)
      - Case 2: 2 Mbps, 4 Mbps and 6 Mbps (the total is 12 Mbps)
    - Both cases are same in terms of Jain's index
- Some solutions for alleviating unfairness in WLANs achieve good fairness but may degrade the throughput

Jain's index cannot evaluate it accurately since the index is independent of the scale of allocations

[7] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," Computer Networks and ISDN Systems, vol. 17, pp. 1-14, 1989

Objectives of this work

1. Propose a transport-layer solution for alleviating TCP unfairness in WLANs
   - Activates congestion control not only for data packet losses but also for ACK packet losses
   - The performance is evaluated in a real WLAN environment with several vendor products
2. Propose an index for evaluating trade-off relationships between fairness and throughput
   - Defines *fair and fully-utilized throughput*
   - Represents how the throughput of each user is close to it

Proposed method for alleviating TCP unfairness in WLANs

- Transport-layer solution
  - Basic concept has been proposed in [6]
  - Regards ACK packet losses as an indication of congestion at an AP
  - Extends the concept
    - Support delayed ACK option
- Summary of the Proposed mechanisms
  - Detects ACK packet losses by monitoring the sequence number of received ACK packets
  - Halves the window size when the number of ACK packet losses exceeds a pre-determined threshold

[6] M. Hashimoto, G. Hasegawa, and M. Murata, "Performance evaluation and improvement of hybrid TCP congestion control mechanisms in wireless LAN environment," in Proceedings of ATNAC 2008, Dec. 2008, pp.367-372

Proposed index

- An index how the throughput of each user is close to fair and fully-utilized throughput, $x_f = \frac{C}{n}$

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - x_f)^2$$

  - C : network bandwidth at a bottleneck link
  - n : the number of flows
  - x_i: throughput of i th flow
- The index normalized by x_f:

$$g(X,C) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x_f)^2}}{x_f}$$

Proposed index

Proposed index

$$F(X,C) = \frac{1}{1+g(X,C)^2}$$

$$g(X,C) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x_f)^2}}{x_f}$$

fair and fully-utilized throughput $x_f = \frac{C}{n}$

Jain's index

$$F_j(X) = \frac{1}{1+COV^2}$$

$$COV = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}}{\bar{x}}$$

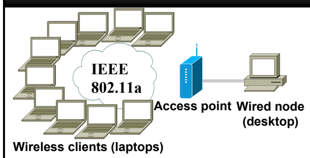
average throughput $\bar{x} = \frac{\sum_{i=0}^{n} x_i}{n}$

coefficient of variance

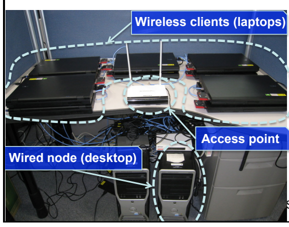
- Fairness and efficiency index

$$F(X,C) = \frac{C^2}{n\sum_{i=1}^{n} x_i^2 - 2C\sum_{i=1}^{n} x_i + 2C^2}$$

Experimental environment



- Ten wireless clients
- One AP
- One wired node



Access Points

Vendor	Product name
Buffalo	WAPS-HP-AM54G54
NEC	Aterm WR8500N
Corega	CG-WLR300NNH

Wireless Interface Cards

Vendor	Product name
Buffalo	WLI-CB-AGHP
NEC	Aterm WL54AG

Experimental method and metric

Experimental method

- Each wireless client generates one TCP connection
  - Bulk data transfer by using Iperf [14]
  - TCP Reno with and without the proposed method
- Keeping the number of concurrent flows at ten
- Changing the ratio of upstream and downstream flows from (0, 10) to (10, 0)

Evaluation metric

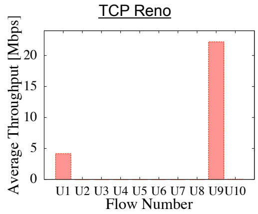
- Fairness
  - Throughput of each flow and average throughput of upstream and downstream flows
- Efficiency of network bandwidth
  - Total throughput
- Trade-off relationship between fairness and throughput
  - Proposed index

[14] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf-the TCP/UDP bandwidth measurement tool," available at http://dast.nlanr.net/Projects/Iperf/

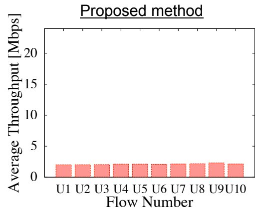
Evaluation – fairness

10 upstream flows and no downstream flow

TCP Reno



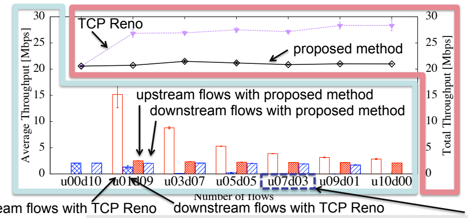
Proposed method



- Few upstream flows occupy all network bandwidth and the other flows are almost starved
- All flows share the network bandwidth equally

Evaluation – fairness and utilization

- Proposed method have almost the same total throughput regardless of the ratio of upstream and downstream flows
- TCP Reno increases the total throughput when one or more upstream flows exist
  - The increased throughputs are distributed among non-starved flows



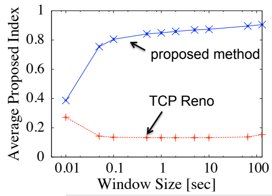
u07d03 means seven upstream flows and three downstream flows

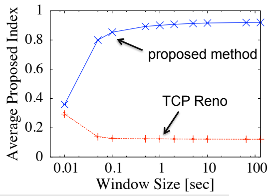
- Proposed method successfully alleviates unfairness between upstream and downstream flows

Evaluation with proposed index

5 upstream flows and 5 downstream flows

10 upstream flows





These figures shows fairness over time-scale of window size in x-axis

$$\frac{C^2}{n\sum_{i=1}^{n} x_i^2 - 2C\sum_{i=1}^{n} x_i + 2C^2}$$

C is set to 29.60 Mbps, the theoretical maximum throughput of 802.11a

- Proposed method achieves a good trade-off relationship between fairness and throughput in terms of not only long-term fairness but also short-term fairness

Conclusion and future work

Conclusion

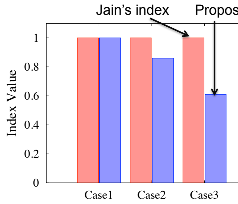
- We proposed the transport-layer solution to alleviate unfairness among TCP flows in WLANs
  - Activates the congestion control not only for data packet losses but also for ACK packet losses
- We proposed the performance index for evaluating the trade-off relationships between fairness and throughput
- We showed that the proposed method is effective regardless of the ratio of upstream and downstream flows in a real WLAN environment

Future work

- Evaluation of the proposed method in environments including wired networks with various traffic scenarios

Thank you for listening

Comparison with Jain's index: simple example



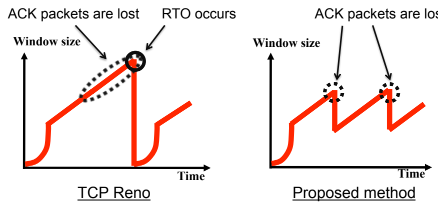
Three users share a bottleneck link with15 Mbps

Case1: each user gets 5 Mbps
Case2: each user gets 3 Mbps
Case3: each user gets 1 Mbps

- Jain's index cannot evaluate the difference in three cases
- Proposed index can evaluate fairness considering the bandwidth utilization
- A special case: when the network bandwidth is fully utilized, i.e., $C = \sum_{i=1}^{n} x_i$

$$F(X, \sum_{i=1}^{n} x_i) = \frac{(\sum_{i=1}^{n} x_i)^2}{n\sum_{i=1}^{n} x_i^2} = F_j$$

An example of the proposed method's behaviors



- TCP Reno
  - Increases the window size regardless of losing ACK packets until retransmission timeout (RTO) occurs
  - Once RTO is caused by all ACK packet losses in a window, TCP sets the congestion window size to one packet
- Proposed method
  - TCP halves the window size whenever it detects some ACK packet losses