TCP Increase/Decrease Behavior with Explicit Congestion Notification (ECN)

Minseok Kwon and Sonia Fahmy

Department of Computer Sciences
Purdue University
{kwonm, fahmy}@cs.purdue.edu
http://www.cs.purdue.edu/~fahmy

Outline

- Motivation
- Background
- ECN(α,β): New ECN Response
- Performance Analysis
- Conclusions

Motivation

- 2 ways of Congestion Indication

Implicit	Explicit
• Time Out • 3 Duplicate Acks • Partial Acks • Increase in RTT (Vegas)	• No unnecessary packet drop • Finer granularity • Distinguish between random losses and congestion losses

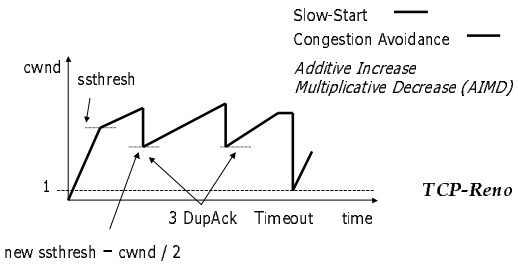
Motivation

- New TCP response to ECN
  - How can we use ECN as *an early warning sign?*
  - Can TCP response to ECN be *more aggressive in the short term while preserving TCP long term behavior?* (Note that RFC 3168 does NOT preclude more aggressive short term behavior)
  - Improved performance gives incentives for hosts to become *ECN-compliant.*
  - *Small changes* to current TCP, *compatible* with RFCs.

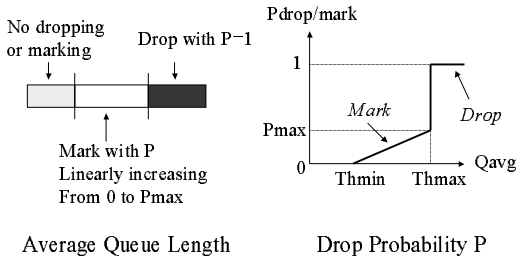
Outline

- Motivation
- Background
- ECN(α,β): New ECN Response
- Performance Analysis
- Conclusions

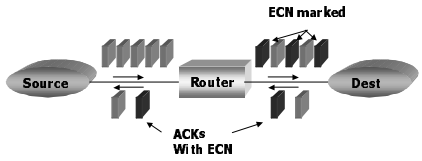
TCP Congestion Control



Random Early Detection (RED)



Explicit Congestion Notification (ECN)



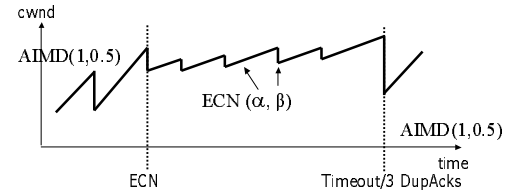
- Problems with non-ECN-compatible equipment:
  2,151 of 24,030 web servers were not accessible to ECN-capable clients (tests in December 2000 using TBIT[2]).

1. K. Ramakrishnan and S. Floyd, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168.
2. TBIT, http://www.icir.org/tbit/

Outline

- Motivation
- Background
- ECN(α,β): New ECN Response
- Performance Analysis
- Conclusions

ECN(α,β): New ECN Response



- The safety of slow responsiveness of TCP-compatible algorithms for deployment is studied by [1].

1. D.Bansal et al., "Dynamic behavior of slowly-responsive congestion control algorithms", ACM SIGCOMM 2001.

ECN(α,β): New ECN Response

Less conservative over short-term while similar to packet drop over long-term

// When an ACK with ECN indication is received:
Reduce *ssthresh* and *cwnd* by β
Set *IncreaseSlope* to α

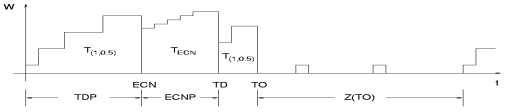
// When a timeout triggers or 3 duplicate ACKs are received:
Reduce *ssthresh* and *cwnd* normally
Reset *IncreaseSlope* to 1

// Congestion avoidance:
cwnd = cwnd + IncreaseSlope / cwnd

$\alpha = 0.2$
$\beta = 0.875$

Modeling TCP Sending Rate

- Evolution of window size of ECN (α, β)



- ECN (α, β) is modeled based on TCP model and assumptions (independent losses) [1,2] in the context of ECN.

1. J. Padhye et al., "Modeling TCP throughput: A simple model and its empirical validation." ACM SIGCOMM 1998, IEEE/ACM Transactions on networking 2000.
2. Y. Yang et al., "General AIMD congestion control." IEEE ICNP 2000.

TCP Sending Rate

• ECN (α, β) sending rate

$$B(RTT,p,r)\approx\frac{1}{RTT\left((1-r)\sqrt{\frac{4p}{3}}+r\sqrt{\frac{4(1-\beta)p}{\alpha(1+\beta)}}\right)+T_0\min\left(1,3\sqrt{\frac{3p}{4}}\right)p(1+32p^2)}$$

where r is the fraction of ECN out of total congestion indications, (α,β) are new response parameters, p is the packet mark/drop rate, T_0 is the timeout interval.

13

ECN(α,β) vs. RED-ECN

- ECN(α,β) at sender, RED-ECN at router
- RED model and assumptions in [1] are used: n flows, link bandwidth c is fully utilized.
- We use B(RTT,p,r) as TCP sending rate.

$$\hat{B}(RTT,p,r)=\begin{cases}B(RTT,p,1), & 0<p\le p_{\max}\\ B(RTT,p,0), & p>p_{\max}\end{cases}$$

Gentle RED-ECN

$$\hat{B}(RTT_0+\bar{q}/c,p,r)=c/n$$

Propagation delay

Average queue size

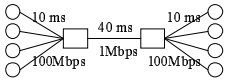
$$p=\begin{cases}0, & 0\le\bar{q}<q_{\min}\\ \frac{\bar{q}-q_{\min}}{q_{\max}-q_{\min}}p_{\max}, & q_{\min}\le\bar{q}<q_{\max}\\ \frac{(1-p_{\max})}{q_{\max}}(\bar{q}-q_{\max})+p_{\max}, & q_{\max}\le\bar{q}<2q_{\max}\\ 1 & 2q_{\max}\le\bar{q}<Q\end{cases}$$

1. V. Firoiu and M. Borden, "A study of active queue management for congestion control" IEEE INFOCOM 2000.

14

Validation

- Simulation Setup
  - The network simulator ns-2.1b6
  - Simple WAN configuration

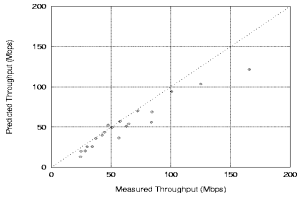


  - 20 unlimited FTP
  - Timer granularity: 100 ms, segment size: 1 KB
  - Gentle RED: 168 KB buffer
  - Total running time: 100 sec

15

Validation

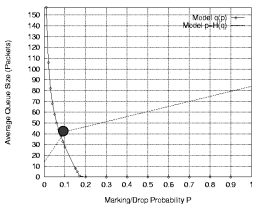
- ECN(α,β) sending rate
  - B(RTT,p,r) vs. measured throughput

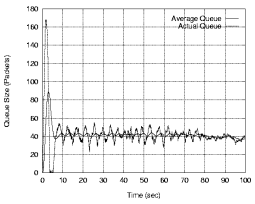


16

Validation

- RED-ECN as a feedback control system
  - Equilibrium point in steady-state





17

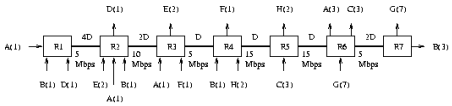
Outline

- Motivation
- Background
- ECN(α,β): New ECN Response
- Performance Analysis
- Conclusions

18

Performance Analysis

- The network simulator ns-2.1b6
- GFC-2 Configuration
  - HTTP, unlimited FTP, UDP (CBR)



- Performance Metrics
  - Web response time, Goodput, Packet drop ratio

Results

Algorithm	Web mean response	Web Goodput	UDP Goodput	FTP Goodput	Packet drop ratio
Reno	14.260	1.509	2.855	38.772	1.637
Reno-ECN	12.194	0.845	2.830	40.805	1.110
ECN (α,β)	11.481	2.339	2.881	41.890	0.854

Results - Responsiveness

- 10 more bulk-data sessions are generated in the middle of the simulation.
- Table shows ECN (α,β) outperforms TCP Reno without ECN and with ECN.

Algorithm	Web mean response	Web Goodput	UDP Goodput	FTP Goodput	Packet drop ratio
Reno	13.607	0.615	5.783	35.789	1.804
Reno-ECN	12.082	0.657	5.958	37.648	1.157
ECN (α,β)	12.841	1.010	5.805	38.494	0.903

Outline

- Motivation
- Background
- ECN(α,β): New ECN Response
- Performance Analysis
- Conclusions

Conclusions & Future Work

- Small changes to current TCP and compatible with RFCs.
- ECN as an early warning sign of congestion.
- More aggressive in the short term, preserving TCP long term behavior.
- Throughput and steady-state drop/marking probability models for ECN(α,β).
- Increased goodput, reduced web response time: incentives for host ECN-compliance.
- Ongoing work: fairness in heterogeneous configurations.