Modeling Finite Buffer Effects on TCP Traffic over an IEEE 802.11 Infrastructure WLAN

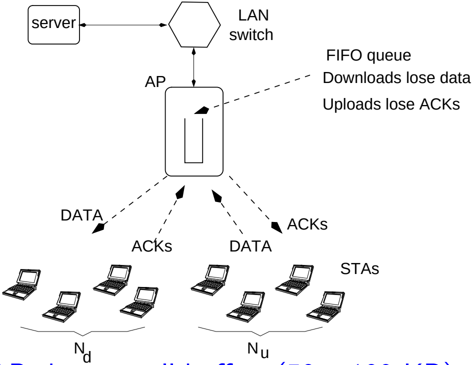
Onkar Bhardwaj[1], G.V.V. Sharma[2], Manoj Panda[3], Anurag Kumar[4]

January 9, 2009

[1]Work done while masters student at IISc, Bangalore
[2]Dept of Electrical Engineering, IIT Bombay
[3]Dept of ECE, IISc Bangalore
[4]Dept of ECE, IISc Bangalore

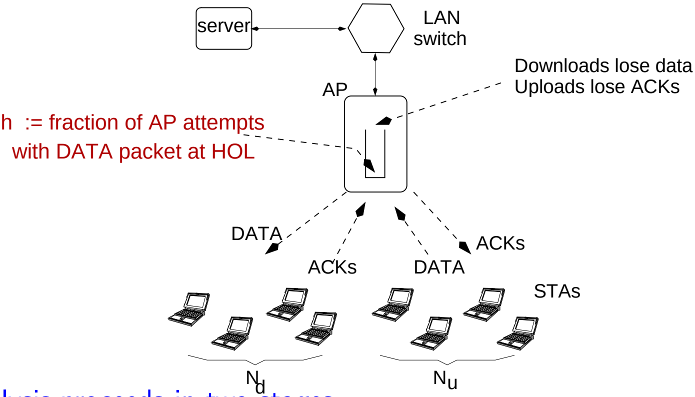
Outline

Problem Statement



- In practice, APs have small buffers (50 – 100 KB) ⇒ packet loss
- Uploading STAs fare better than downloading STAs (Pilosof et al. 2003, Bruno et al. 2004, Gong et al. 2006)
- Reasons:
  - Packets are large, hence suffer higher loss rates
  - ACKs are cumulative, hence ACK loss does not affect TCP progress
- We provide a stochastic model to quantify this unfairness

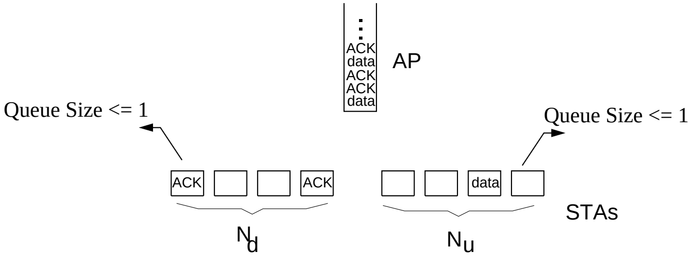
Overview of Modeling Approach



- Analysis proceeds in two stages
- MAC model:
  - AP is the bottleneck $\Rightarrow$ always backlogged
  - Obtain $\Theta :=$ the aggregate packet throughput of the AP
  - Then $\Theta_d = h \times \Theta$, and $\Theta_u = (1 - h) \times \Theta$
- Transport model: Analyze TCP window evolution to obtain h

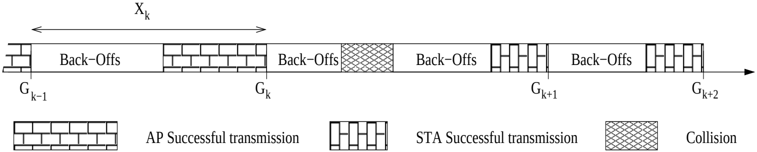
MAC Model

MAC Model: Observations, Modeling Assumptions



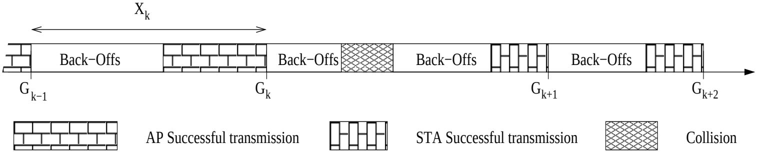
- Undelayed ACK case (delayed ACK case in the paper)
- AP must send 1 packet for each packet sent by an STA
  - IEEE 802.11 MAC give no priority to the AP
- ⇒ modeling assumptions:
  - AP backlogged
  - STAs have 1 packet or no packet
- We model the number of active download and upload STAs

MAC Model: Obtaining AP Throughput Θ



- G_k : end of the kth packet transmission on the medium
- (D_k, U_k) : number of download/upload STAs with packets at G_k
- AP transmits to only empty STAs
- Can be modeled as a discrete time Markov chain embedded at $G_k, k \geq 1$
- AP throughput Θ obtained by Markov Regenerative analysis

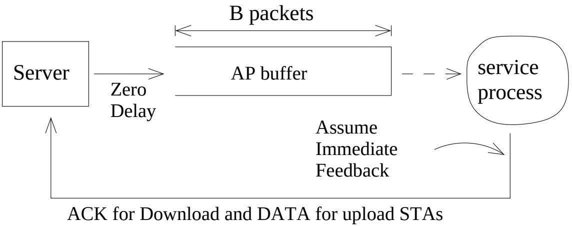
MAC Model: Obtaining AP Throughput Θ



- G_k : end of the kth packet transmission on the medium
- (D_k, U_k) : number of download/upload STAs with packets at G_k
- AP transmits to only empty STAs
- Can be modeled as a discrete time Markov chain embedded at $G_k, k \geq 1$
- AP throughput Θ obtained by Markov Regenerative analysis
- We now need Θ_d and Θ_u
- .... for which we need h
- .... which we get from a transport layer model

Transport Layer Model

Modeling TCP: Model for AP Buffer Evolution



- Above simplification suffices to obtain an approximation to h
  - Avg no. of *active* STAs is 1.5, STA Queue size ≤ 1
  - Almost all packets rest in AP
- The server models the CSMA/CA service to the AP (as modeled earlier)
- One packet sent out from AP, one or two packets immediately arrive in AP

Modeling TCP: Assumptions on Window Sizes

- Upload TCP windows unbounded $\Rightarrow$ downloads choked (Gong et al. RAWNET 2006)
- We take upload TCP windows to be bounded by $W_{\max}$
  - Upload connections reach W_{max} and stay there indefinitely
  - $B = b + \mu, \quad \mu = N_u W_{max}$
- Download windows:
  - Unbounded case: exact analysis of h
  - Bounded case: we obtain an upper bound and an approximation for lower bound on h

Modeling TCP: Synchronized TCP Window Evolution

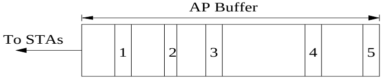


Figure: Full AP Buffer

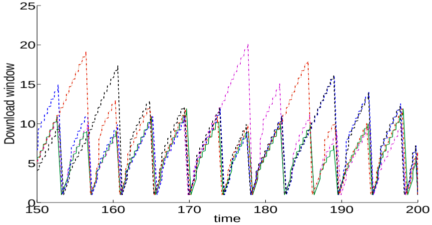


Figure: Window Synchronization for TCP OldTahoe

- TCP window evolution is cyclical
- Its analysis yields expressions for h (next slide)

Modeling TCP: Expressions for h

- TCP OldTahoe

$$h = \frac{\left[(2^r - 1) + \frac{x(x-1)}{2} + 3x\right] N_d + (x+3)\frac{b}{2}}{(r+x+3)\mu + \left[(2^r - 1) + \frac{x(x-1)}{2} + 3x\right] N_d + (x+3)\frac{b}{2}}$$

- TCP Reno

$$h = \frac{\left[\frac{x(x-1)}{2} + 3x\right] N_d + (x+3)\frac{b}{2}}{(x+3)\mu + \left[\frac{x(x-1)}{2} + 3x\right] N_d + (x+3)\frac{b}{2}}$$

- where

$$x = \frac{b}{2N_d} \quad r = log_2\left(\frac{b}{2N_d}\right) \quad \mu = N_u W_{max}$$

- A bound for h is also provided in the paper, for the case when both directions have a TCP window limit

Results

h for Undelayed ACK, TCP Reno

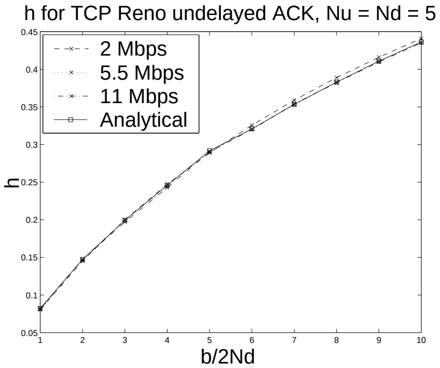


Figure: TCP Reno, Undelayed ACK case: h vs. buffer size (expressed as $\frac{b}{2N_d}$) for $N_u = N_d = 5$

h for Delayed ACK, TCP Reno

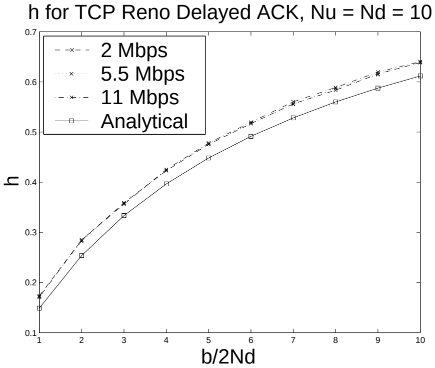


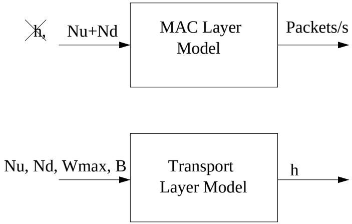
Figure: TCP Reno, Delayed ACK case: h vs. buffer size (expressed as $\frac{b}{2N_d}$) for $N_u = N_d = 10$

Discussion: h for TCP Reno

- Analysis of h is *rateless*
- Only assumption it needs is small number of contending STAs
- Maximum limit on window connections helps fairness !

Results for Throughputs

PHY Rate	Θ (packets/s)		
	Undelayed	$h = 0$, Delayed	$h = 1$, Delayed
2 Mbps	≈ 117	126.11	123.81
5.5 Mbps	≈ 231	261.58	251.95
11 Mbps	≈ 321	377.30	357.63



Final Remarks

- We have provided an analytical model for TCP controlled file transfers in an infrastructure WLAN
  - capturing the effect of finite AP buffers
- Combines a MAC model with a model for TCP window evolution

Thanks !!