CSE 322: PROJECT PROPOSAL

# DC-VEGAS

## A DELAY-BASED TCP CONGESTION CONTROL ALGORITHM FOR DATACENTER APPLICATIONS

MD ASIF SHAHRIAR #1805040

# Reference Paper

[DC-Vegas: A delay-based TCP congestion control algorithm for datacenter applications (researchgate.net)]

Authors : Jingyuan Wang, Jiangtao Wen, Chao Li, Zhang Xiong, Yuxing Huan

Journal of Network and Computer Applications, April, 2015

# DC-Vegas: Motivation

- Traditional TCP congestion control algorithms like TCP Vegas do not work well in datacenters
- TCP Vegas estimates a current queue length, $q$, and adjusts the congestion control window size in each RTT by comparing $q$ with a threshold
- This binary congestion detection does not work in datacenters because the queue length variation in datacenters are quite uniform
- Thus, TCP-Vegas detects both false congestion and non-congestion in datacenters

# DC-Vegas: Motivation

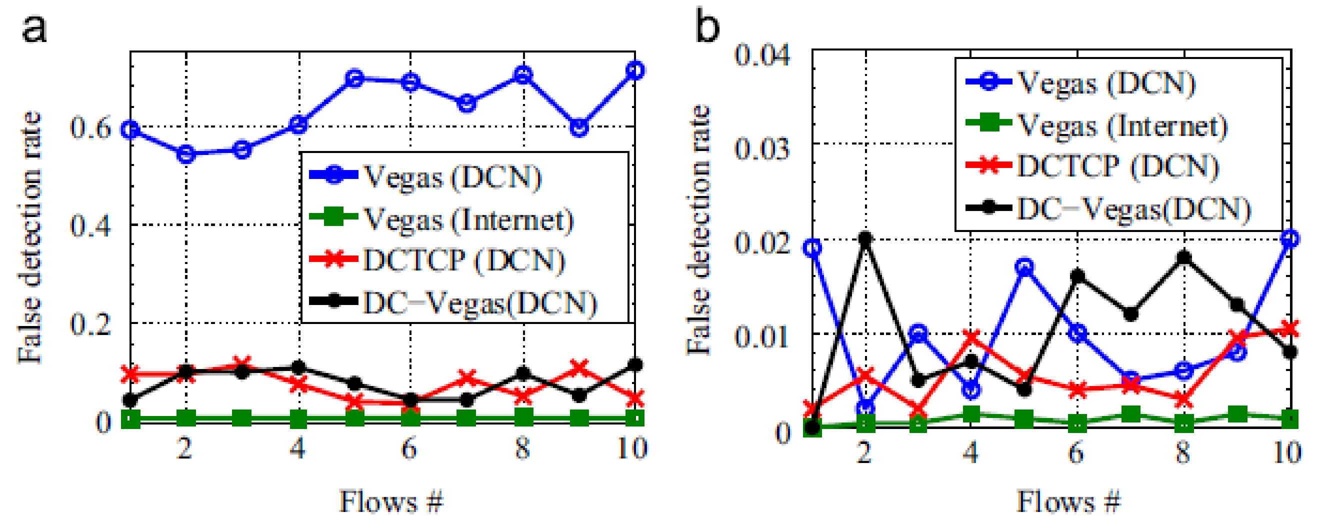


**Fig. 5.** False congestion/non-congestion detection rate.

# DC-Vegas: Motivation

- There are some promising algorithms like DCTCP, which has a significantly lower false rate than TCP Vegas
- Problem: DCTCP requires ECN (Explicit Congestion Notification) support and both sender and receiver modifications, so it is not suitable for already existing datacenters
- Proposed solution: DC-Vegas; which can achieve result close to that of DCTCP without requiring ECN support and by modifying the sender only

# DC-Vegas: Algorithm

- When an ACK arrives, the sender first estimates current network queue length $q$ and compares it with a threshold $K_{dcv}$
- When all packets in the same window are acknowledged, DC-Vegas calculates $F_{dcv} = \frac{\#\, of\ ACKs\ with\ q > K_{dcv}}{Total\ \#\, of\ ACKs\ in\ a\ window}$
- DC-Vegas applies the EMA filter to find $\alpha_{dcv} \coloneqq (1 - g) \times \alpha_{dcv} + g \times F_{dcv}$
- DC-Vegas updates its window $w_{dcv}$ in each RTT according to the network congestion level indicated by $F_{dcv}$

# DC-Vegas: Algorithm

- $W_{dcv} = \begin{cases} W_{dcv} - W_{dcv} \times \frac{\alpha_{dcv}}{2}; & F_{dcv} > 0 \\ W_{dcv} + 1; & F_{dcv} = 0 \end{cases}$

- When network congestion is high, the window is reduced significantly to alleviate the congestion
- Like the traditional TCP Reno algorithm, DC-Vegas halves its window when a packet loss event is detected

# DC-Vegas: Performance

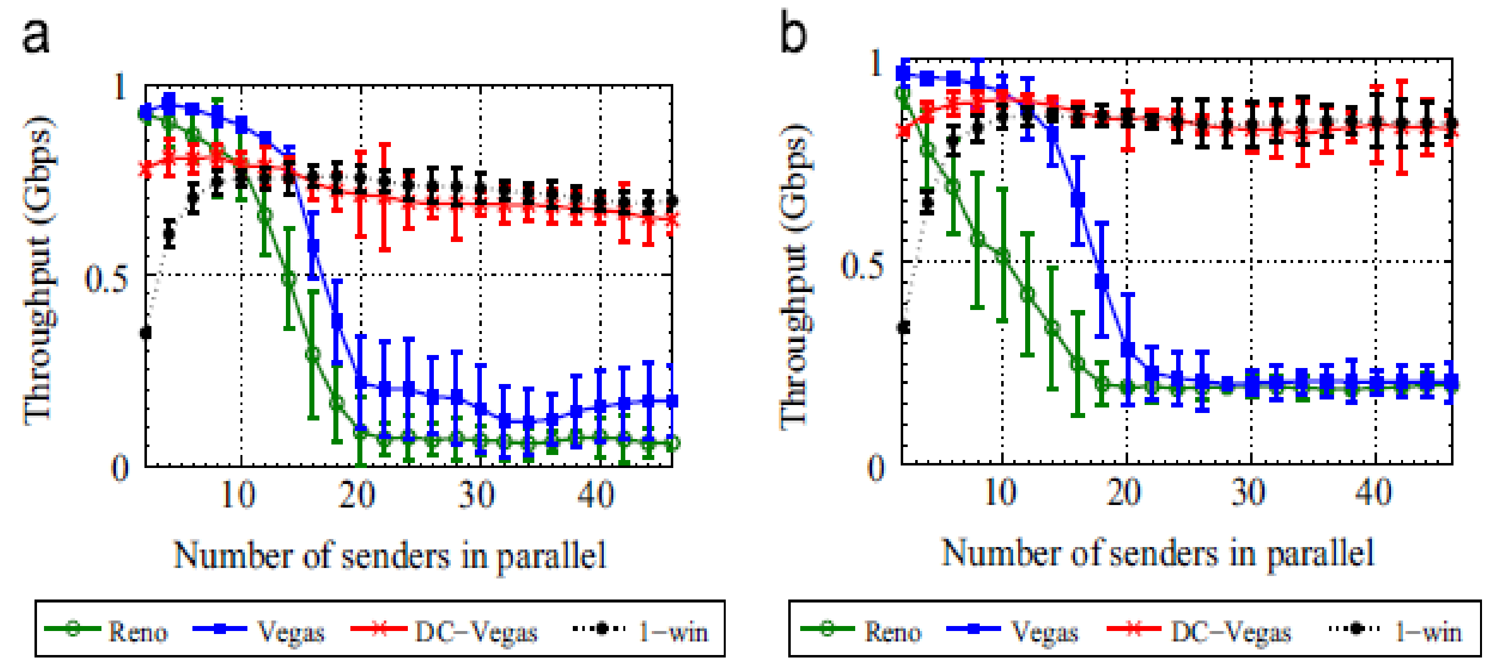


Fig. 8. Incast experiments. (a) Total traffic=1 MB and (b) Total traffic=4 MB.

# DC-Vegas: Performance

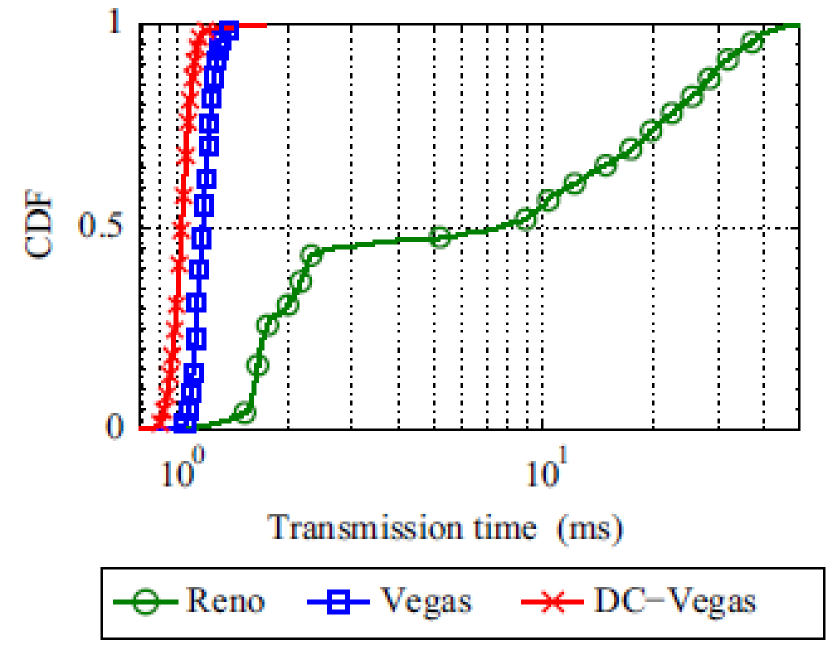


Fig. 10. Transmission time of mouse.

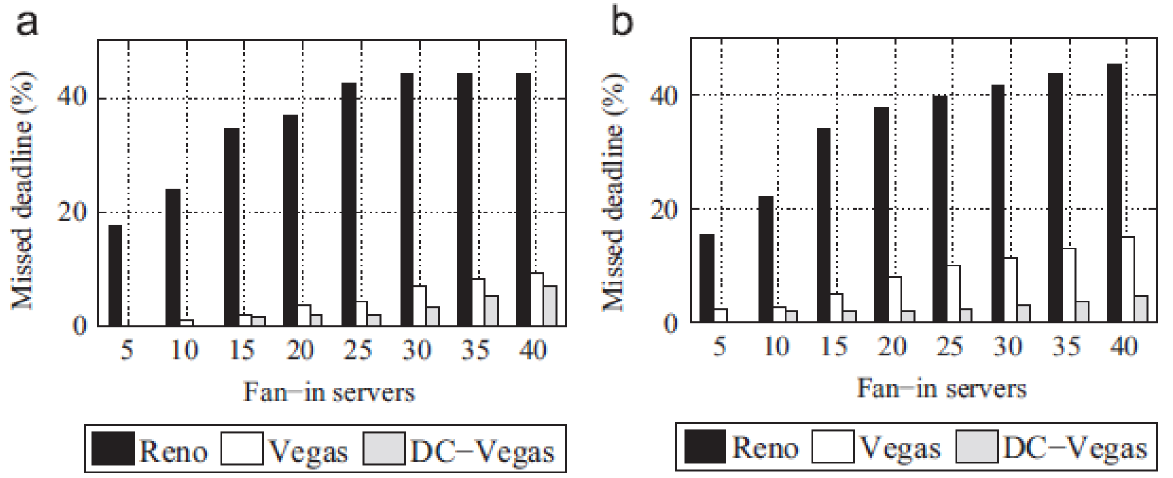


Fig. 11. Missed deadline rate of mouse flows. (a) 100 KB, 5 ms deadline and (b) 1 MB, 50 ms deadline.