TP-Planet

A Transport Protocol for InterPlanetary Internet

Santhosh Sundararaman
Department of EECS, University of Kansas
Email: santhosh@ku.edu

Overview

- Introduction
- TP Planet – Concept
- Functional States
- Initial State
  - Immediate start
  - Follow-up phase
- Steady State
  - Congestion control
  - Rate based AIMD
  - Delayed SACK
- Performance Evaluation
- Conclusion

Introduction

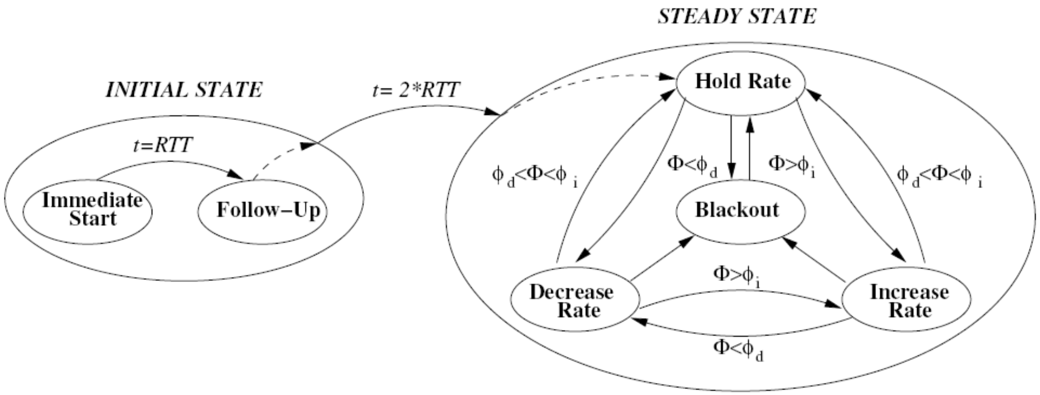
- Challenges
  - Very long propagation delays
  - High link error rates
  - Blackouts
  - Bandwidth asymmetry

- Motivation
  - Poor performance of TCP’s slow start and congestion avoidance

TP Planet – Concept

- Long propagation delay
  - Initial state algorithm for fast and controlled utilization of resources
  - Rate based AIMD
- High link error rates
  - New congestion detection mechanism with loss discrimination
- Blackout
  - Blackout state procedure
- Bandwidth asymmetry
  - Delayed SACKs

Functional States

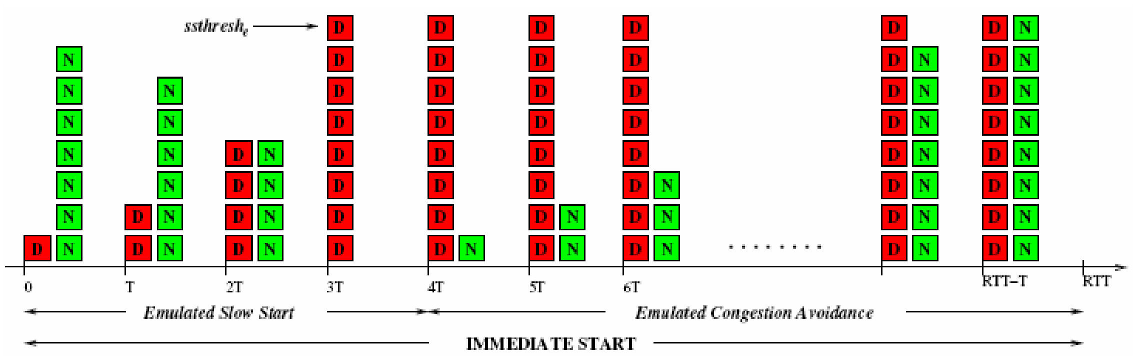


TP Planet – State Transition Diagram

- Initial State
  - Immediate start
  - Follow-up phase
- Steady state
  - Congestion control
  - Rate based AIMD
  - Delayed SACK

Initial State

- Immediate start
  - Probe network to quickly increase transmission rate
  - RTT divided into time intervals of size *T*
  - Conventional slow start and congestion avoidance in interval *T*
  - Two phases : *Emulated slow start and Emulated congestion avoidance*
  - Emulated slow start
    - *cwnd* increased geometrically in each interval until *cwnd* ≤
    - Low priority *NIL* segments transmitted with data to probe the link
    - $cwnd + cwnd_N \leq ssthresh_e$



Initial State

- Emulated congestion avoidance
  - *cwnd* = $ssthresh_e$ until end of emulated congestion avoidance
  - $cwnd_N$ additively increased by 1 per interval T

- Follow-up phase
  - Feedback received for packets transmitted in Immediate Start
  - Total no. of packets received by Sink (N) in every interval T included in NIL ACKs
  - S = N/T from time (RTT + T) to (2 * RTT)
  - S = min(N_{ACK} , 2 * $ssthresh_e$) / T, at the end of Initial state
  - Low and high priority NIX segments transmitted at rate S

- T = $\sqrt{(RTT/B)}$
- Connection Establishment
  - Data transmission starts before ACK for connection request is received

Steady State

- Internal States
  - *Increase rate, decrease rate, hold rate and blackout*
  - Source begins with hold rate
  - New congestion control algorithm determines state transitions

- Congestion control
  - Low and high priority NIX segments used for loss discrimination
  - Link error results in same loss rate for low and high priority NIX
  - Congestion results in higher loss rate for low priority NIX
  - Sink records N_{high} and N_{low} in a sliding time window T_ω
  - NIX ACKs with (N_{high}, N_{low}) sent to source at the end of τ
  - $\Phi = (N_{high} / N_{low})$ determines internal state transitions
    - $\Phi < \varphi_d$: Transition to Decrease Rate, $S = S \cdot \zeta$
    - $\varphi_d \le \Phi \le \varphi_i$: S unchanged
    - $\Phi > \varphi_i$: Transition to Increase Rate, $S = S + \delta$

Steady State

- Rate-based AIMD scheme
  - Additive increase parameter α calculated as

$$\alpha = \frac{(1+\xi)}{2}\left(B + \frac{1}{RTT \cdot p}\right)\left[\sqrt{1 + \frac{8B^2(1-\xi)}{\left(B+\frac{1}{RTT \cdot p}\right)^2 (1+\xi)^2}} - 1\right]$$

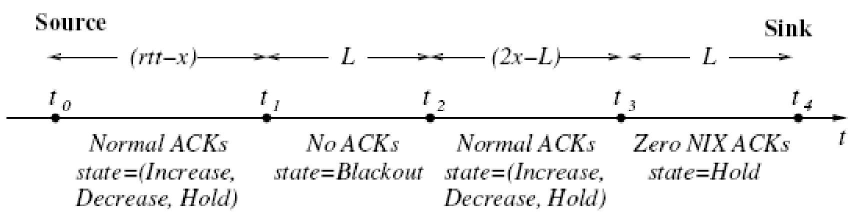
  - ξ: Multiplicative decrease parameter, p: Packet loss probability
    B: Target throughput
  - α and ξ are parameter for period RTT
  - For period τ
    - Multiplicative decrease parameter

      $\zeta = \xi^{(\tau/RTT)}$
    - Additive increase parameter

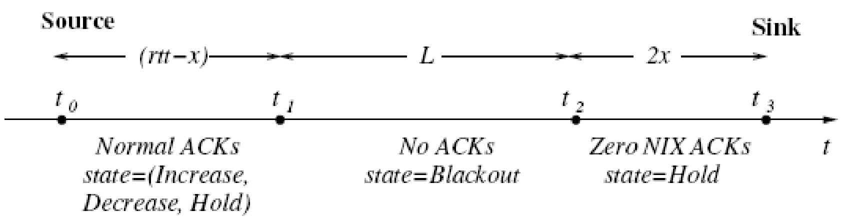
      $\delta = \alpha \cdot \tau/RTT$

Steady State

- Blackout state behavior
  - Blackout inferred if no ACK (or packet) received for T_ω : Blackout state
  - Source sends low and high priority NIX packets at the same rate
  - Sink sends NIX ACKs with (N_{high}, N_{low}) set to (0, 0)
  - A black out occurring at time t_o at a position x seconds from sink with a duration of L leads to two distinct cases
  - L < 2x



  - L > 2x



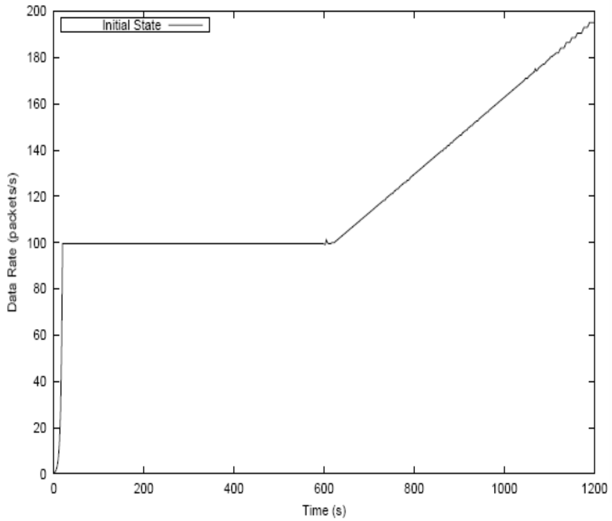
Steady State

- Delayed SACK
  - Sink maintains delayed SACK factor *d*
  - Sack sent for every *d* data packets
  - If no loss, then sink keep delaying SACKs by *d*
  - If loss, then sink sends new SACK with updated blocks
  - *d* controls the amount of traffic on the reverse channel

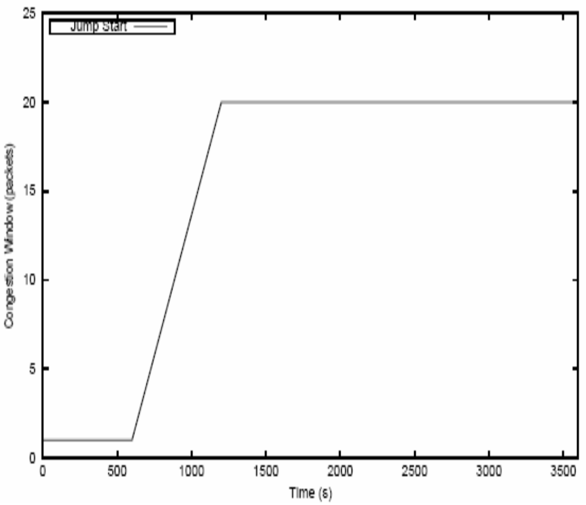
Performance Evaluation

- Simulation
  - Deep space link RTT = 600s
  - Packet loss probability p = 10^{-5}
  - TP Planet compared to TCP Peach+ and TCP NewReno

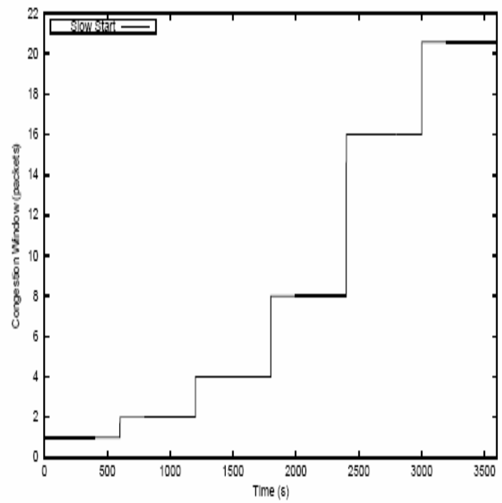
- Initial State
  - Target transmission rate = 100 packets/s



TP Planet



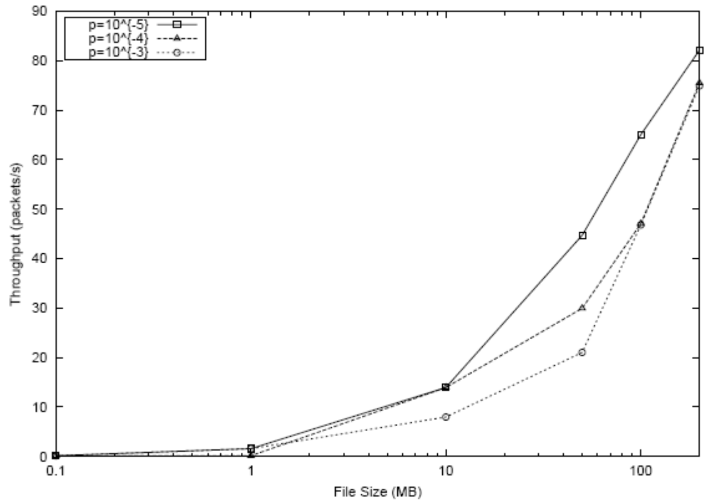
TCP Peach+



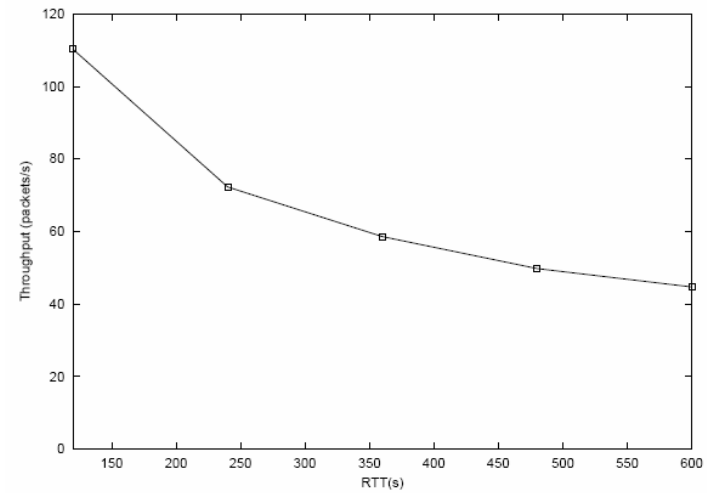
TCP NewReno

Performance Evaluation

- Throughput
  - Link capacity 1Mb/s
  - Data size varied from 100KB to 200MB
  - p = 10^{-5}, 10^{-4}, 10^{-3}
  - TCP protocols achieved ~30B/s with p = 10^{-3}
  - TCP Peach achieved ~93B/s
  - TP Planet



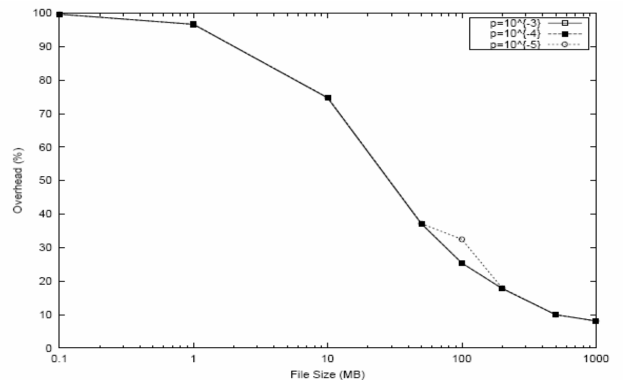
Throughput for varying p and file size with RTT = 600s



Throughput for varying RTT with file size = 50MB, p = 10^{-5}

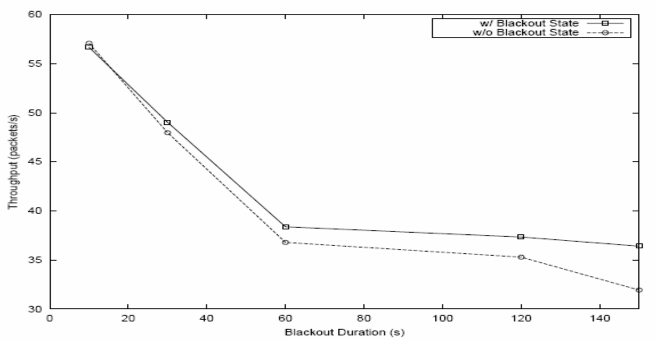
Performance Evaluation

- Overhead
  - NIL and NIX segment transmissions causes overhead



Overhead for varying file size and p with RTT = 600s

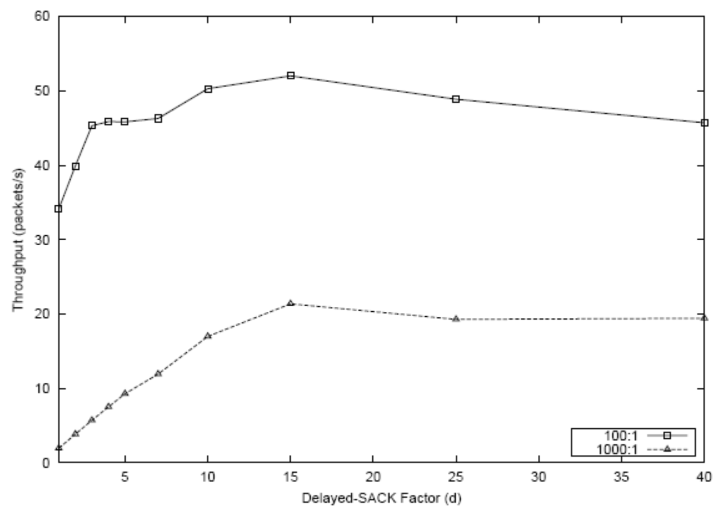
- Blackout



Throughput for varying blackout duration with RTT = 120s and $p = 10^{-5}$

Performance Evaluation

- Bandwidth Asymmetry



Throughput for varying *d* with RTT = 120s and p = 10^{-4}

Conclusion

- TP Planet shows performs better than TCP protocols when the connections is subjected to
  - Long delays
  - High link error rates
  - Blackouts
  - Bandwidth asymmetry

- Suitable for interplanetary networks

Primary Reference

- Ozgur B. Akan, Jian Fang, and Ian F. Akyildiz, *"TP-Planet: A New Transport Protocol for InterPlaNetary Internet,"* IEEE Journal of Selected Areas in Communications (JSAC),Vol. 22, Issue 2, pp. 348-361, February 2004