Congestion Control Without a Startup Phase

Dan Liu[1], Mark Allman[2], Shudong Jin[1], Limin Wang[3]

1. Case Western Reserve University,
2. International Computer Science Institute,
3. Bell Labs

PFLDnet 2007

Motivation

- Slow start in Internet congestion control
- *Slower* start in high bandwidth-delay product (BDP) networks
  - Many RTTs before reaching an appropriate sending rate
  - More cautious slow start (slower) in high BDP networks
- Can we get rid of slow start? Is it too bold and blunt?
- We present an exploratory study – Jump Start

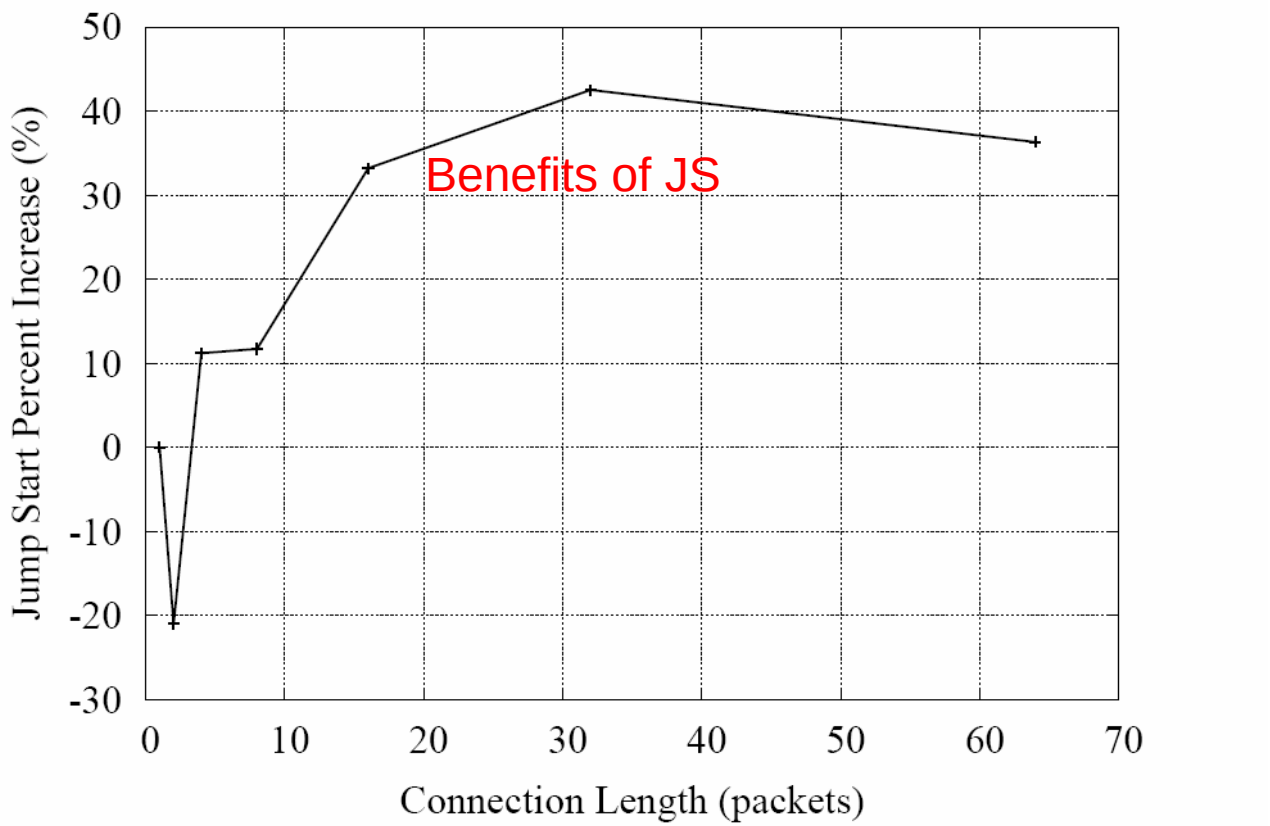
Mechanisms to Improve Slow Start

- Many were proposed and studied to improve TCP's slow start
  - Category 1: use bandwidth estimation techniques
    - e.g., the *Swift Start* algorithm [Patridge et al, 2002] uses the packet-pair technique
    - Disadvantage: network dynamics affect the accuracy.
  - Category 2: maintain/share path capacity information
    - e.g., *Congestion Manager* [Balakrishnan, 1999]
    - Disadvantage: users without recent connection information
  - Category 3: exploit network-assisted mechanisms (negotiate a rate)
    - e.g., *Quick-Start* [Floyd et al, 2006]
    - Disadvantage: a router may not understand the option
- Where to place Jump Start?

A Short Description of Jump-Start

- General process
  - Takes an RTT sample from the three-way handshake.
  - Determines how many data packets, D, can be transmitted.
    - receiver's advertised window, the amount of data queued locally for transmission, …
  - Paces the D packets over the first RTT.
  - Jump Start terminates when an ACK arrives.
    - At this point the TCP switches to TCP's normal congestion control algorithms
    - Standard loss recovery in the case of loss from the first RTT of data transmission.
- SACK version in our simulations
- Upon loss detection, *cwnd* is halved. However, at the end of loss recovery, we further reduce the *cwnd* to reflect the possible over-aggressiveness of Jump Start.

A First Look at Benefits and Drawbacks of JS

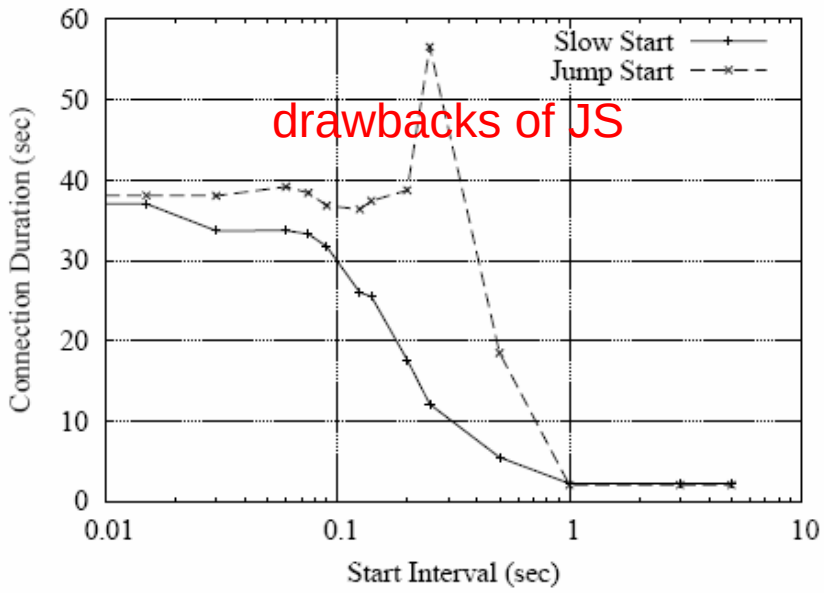


- ns-2, a dumbbell topology (min RTT=64ms, bottleneck capacity=5Mbps)
- DropTail bottleneck with a queue size of 60 packets.
- Either all Slow Start or all Jump Start traffic. Slow Start connections set initial *cwnd* to 3 packets. Jump Start connections send entire transfer.

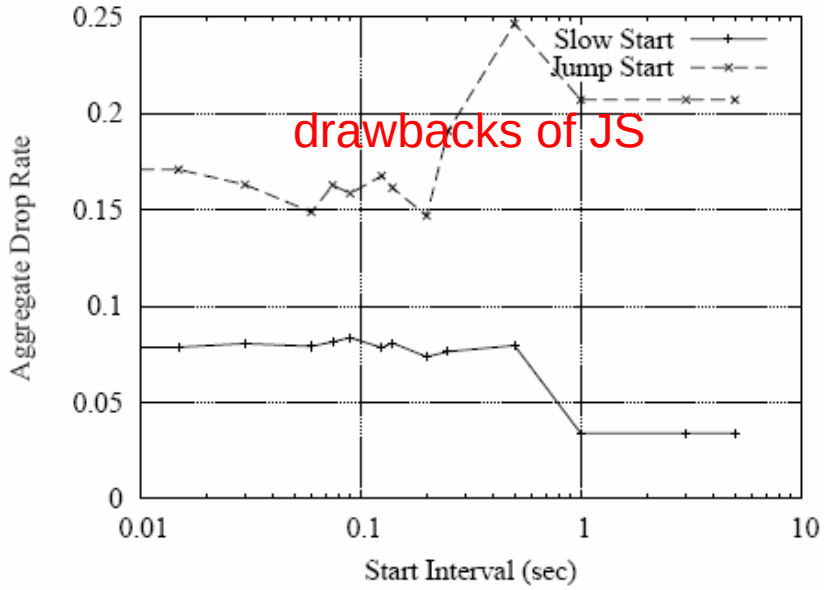
- 100 flows in both direction
- Each flow contains 200 packets
- Vary inter-arrival time of flows
  - Small interval: severely congested
  - Large interval: lightly loaded
- Observations
  - Slow start wins everywhere

SUMMARY: The simulations, though not realistic but illuminative, show that the choice of JS depends on the network/traffic situations.

ARGUMENT: Many ways to cope with flows using Jump Start



(a) Connection duration.



(b) Aggregate drop rate.

Heavy-Tailed Traffic Distribution

- Most of the connections cannot place a large burden on the network because they transfer only a small amount of data.
- Traffic trace from ICSI's border for one day (July 27, 2006), with roughly 1.2 million valid connections
  - Only 169K transfers (in either direction) required an initial congestion window of more than the 4380 bytes (3 segments)
  - if Jump Start were used, the fraction of connections imposing a higher load would also be small.
- For these 169K transfers, if we use Jump Start, the Figure shows the amount of data and maximum sending rate

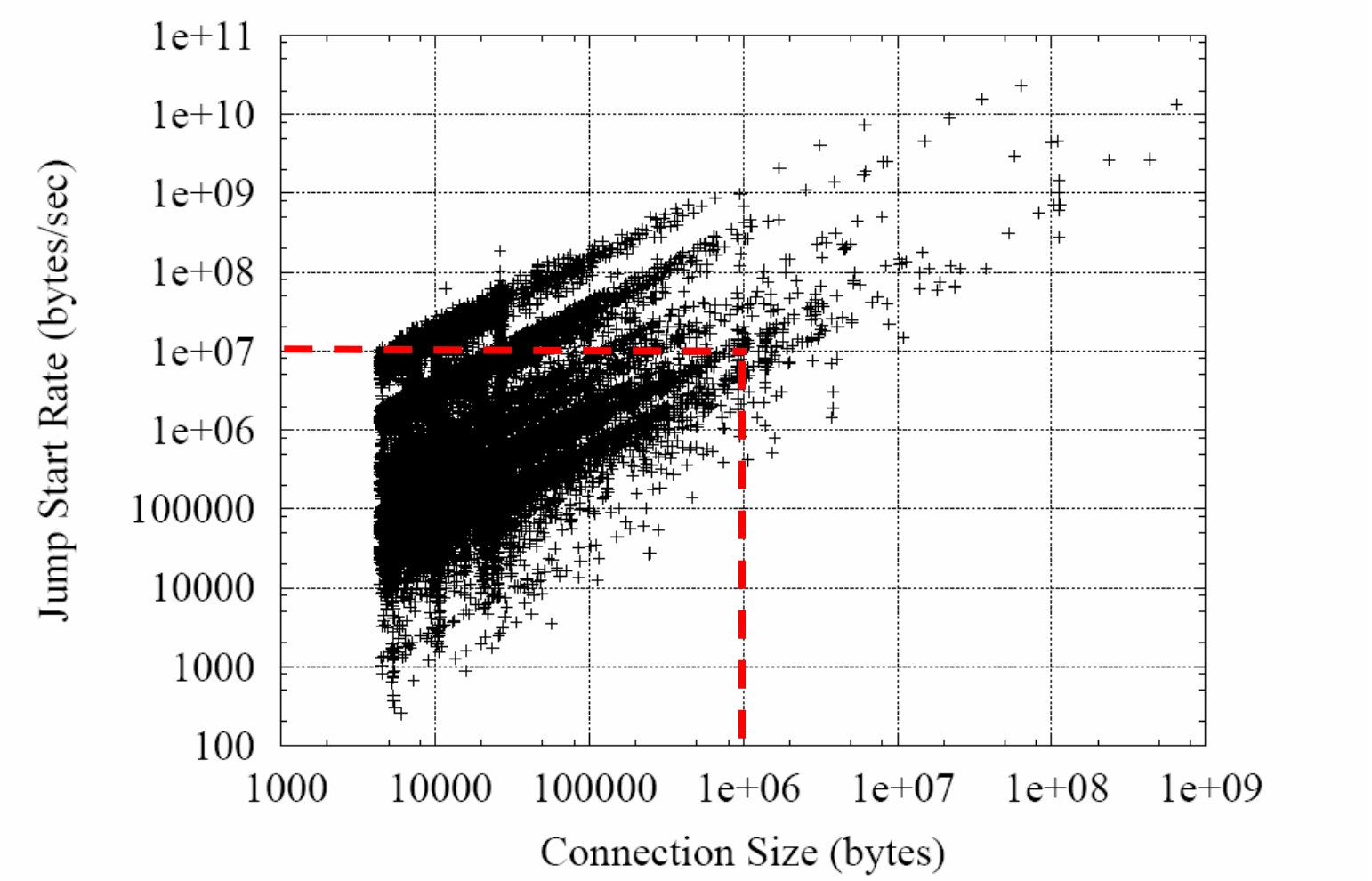
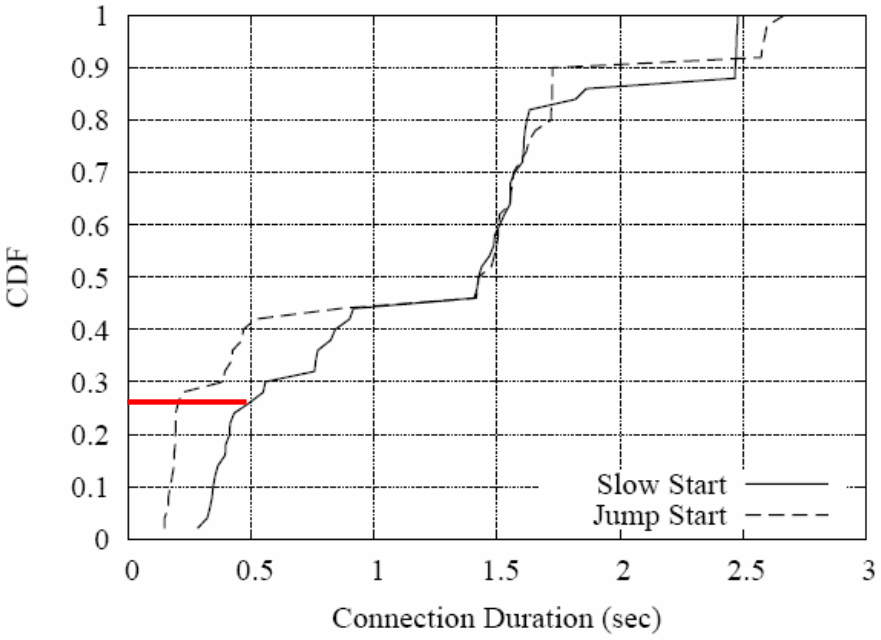
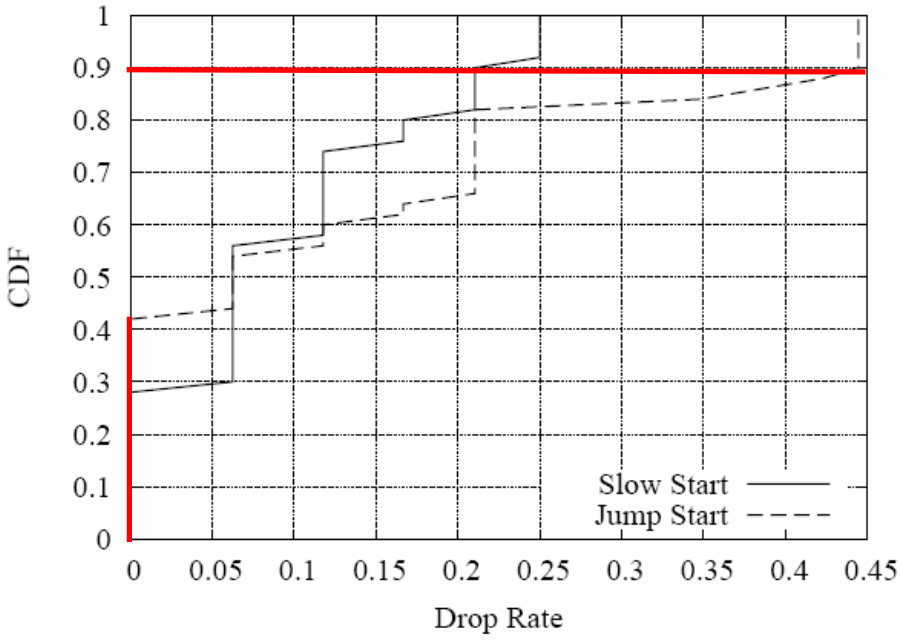


Fig. 3. Observed initial Jump Start rates as a function of transfer size at ICSI's border.

Simulation with Web-like Traffic



(a) Connection duration.



(b) Aggregate drop rate.

- *ns-2*'s web traffic generator creates 50 web sessions consisting of 5 web pages per session on average.
  - Pareto-II web page size (mean=4 objects, shape=1.5)
  - Exponential inter-page time (mean=500ms).
  - Pareto-II object size (mean=4 packets, shape=1.2).
  - Exponential inter-object time (mean=50ms).

Other Coping Factors

- AQM
  - Absorb burst; monitoring/dropping
- Edge bandwidth limits
  - Where is the bottleneck? Advertised window size
- Policy or self-refrained users
  - Both sender and receiver can implement policies
- End system hints
  - Mark Jump Start packets for preferential dropping
- Measurement (more like Swift Start)
  - Packet pair/train
  - Start with large cwnd, and use delays to infer a safe cwnd

Future Work

Jump Start may not be ready for deployment now

- More comprehensive experiments (with more realistic traffic, more variable network bandwidth and RTTs, etc).
- Fairness issue with both Slow Start and Jump Start flows