Université de Nice - Sophia Antipolis
Ecole Doctorale STIC

Performance Evaluation of TCP congestion control

Chadi BARAKAT

Laboratory:
Mistral team
INRIA - Sophia Antipolis

Advisor:
Eitan Altman

TCP: Importance and objectives

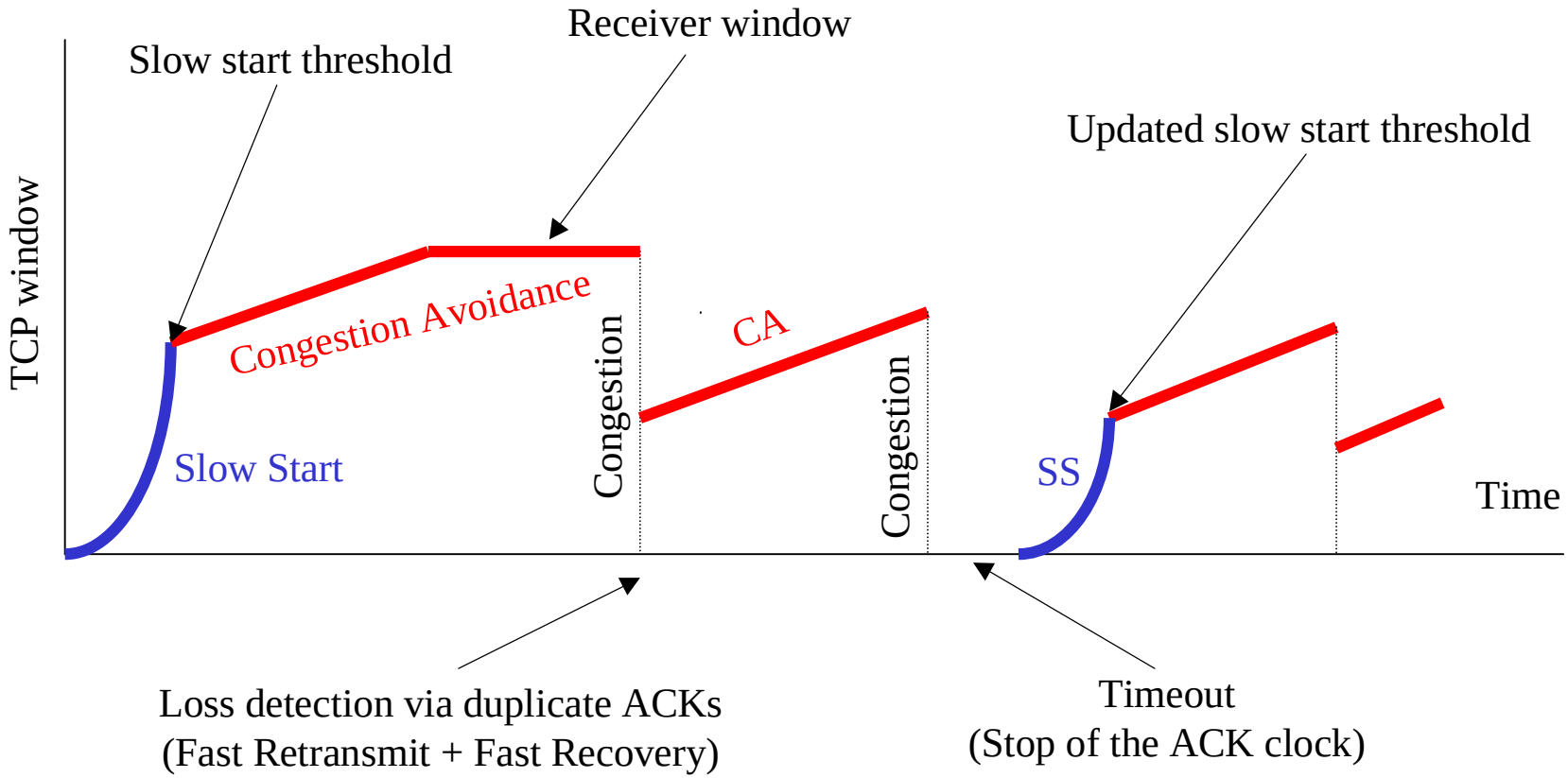
- TCP (*Transmission Control Protocol*) congestion control mechanisms are very important for the stability of the Internet:
  - Used by most of Internet traffic (*90% of all packets*).
  - Flows not using TCP are supposed to adapt their rates so as to be friendly with TCP flows.

- TCP (*or any other congestion control protocol*) objectives:
  - An efficient utilization of network resources (*high overall throughput, short queues in routers, low retransmission ratio, low drop probability*).
  - Fair sharing of resources between users (*e.g., max-min fairness*).

  and this must hold for any size of the network and for any technology used to transmit Internet traffic between IP routers (*e.g., satellites, mobile networks*).

TCP mechanisms

- A window-based flow control protocol.
- Window size adapted as a function of network conditions ...



TCP congestion control principles

- The window is increased linearly until a buffer in a network router overflows (*no avoidance of network congestion, but this permitted a simple addition of TCP congestion control to the Internet*).
- All packet losses in the same round-trip time are considered as a single congestion notification (*no reaction in less than one RTT*).
- During congestion, the window is divided conservatively by two in order to reduce the load on the network and to converge quickly to fairness.
- Packets are only transmitted when ACKs are received (*no packet spacing*).
- A TCP source tries always to keep a window size of packets in the network. If it happens that the network drains off, slow start is to be used so as to reestablish the ACK clock.

TCP congestion control problems

But, TCP has different problems that prohibit it from scaling to a large and heterogeneous Internet [Barakat et al., Jan. 2000]:

- Unfairness in case of connections of different RTT (*slow window growth for connection of long RTT*).
- Poor performance for short transfers (*due to slow start*) on long delay paths (*e.g., satellite links*).
- Low throughput on paths where packets are lost for other reasons than congestion (*e.g., transmission errors in wireless networks*).
- Low throughput, burstiness, and unfairness in networks where the reverse path is not fast enough to carry the flow of ACKs (*e.g., satellite and cable networks*).

Result: A large number of models and solutions to evaluate and improve the performance of the protocol ...

Approaches for studying TCP

End-to-end approach:

- Consider the network as a black box.
- Analyze the protocol and improve its mechanisms on end-to-end basis without any help from the network or any knowledge of its content.
- Examples:
  - End-to-end models for the calculation of TCP throughput.
  - Works on improving the slow start phase.
  - Works on improving the phase of recovery from errors.
  - Works on anticipating network congestion.

.
.
.

Approaches for studying TCP

Network-specific approach:

- Consider the content of the network while studying the performance of TCP:
  - Characteristics of transmission media.
  - Scheduling and buffer management techniques in IP routers.
- Propose modifications to the network (*and possibly to TCP*) to help the protocol in his operation.
- Examples:
  - Optimization of link-level protocols (*MAC,FEC, ARQ,etc*.).
  - Buffer management techniques for distributing drops intelligently on TCP and non-TCP flows (*e.g., RED,Flow RED,Longest Queue Drop*).
  - Helping the TCP connection at the TCP level (*e.g., splitting the connection, filtering ACKs, spacing ACKs*).

.
.

Our objectives

Use of modeling tools (*validated by simulations and real measurements*) to:

- ☞Conduct a thorough and a precise evaluation of TCP performance.
- ☞Evaluate the performance of mechanisms proposed (*at the source and/or inside the network*) to help the protocol in some challenging environments.
- ☞Propose new mechanisms and guidelines.

We adopt the two previous approaches in our study ...

End-to-end approach

- **Objective:** Find simple expressions for the throughput of a TCP connection.
- **Motivations:**
  - Heterogeneity of the Internet and simplicity of existing models for TCP.
  - Importance of TCP throughput expressions for understanding TCP performance and for designing Internet applications and routers.
- **Outline:**
  - Measurement-based justification for the need for a general approach.
  - Markovian approach: For paths that change their state according to a Markov chain (e.g., wireless link).
  - General approach: For a general path.
  - Modeling of some other mechanisms of the protocol (Timeout, packet nature of TCP, limitation due to receiver window).

Network-specific approach

- **Objective:** Improve the performance of TCP in some challenging environments.
- **Motivation:** Problems we identified with existing mechanisms.
- **Outline:**
  - Large bandwidth-delay product networks and short transfers:
    Optimization of the window increase policy during slow start.
  - Networks with a slow reverse path:
    Optimization of the ACK filtering strategy.
  - Networks with non-congestion losses at the link level (e.g., wireless links):
    Optimization of the amount of FEC to be added to the noisy part.

One can always consider other environments.

End-to-end modeling of TCP

End-to-end TCP modeling requires:

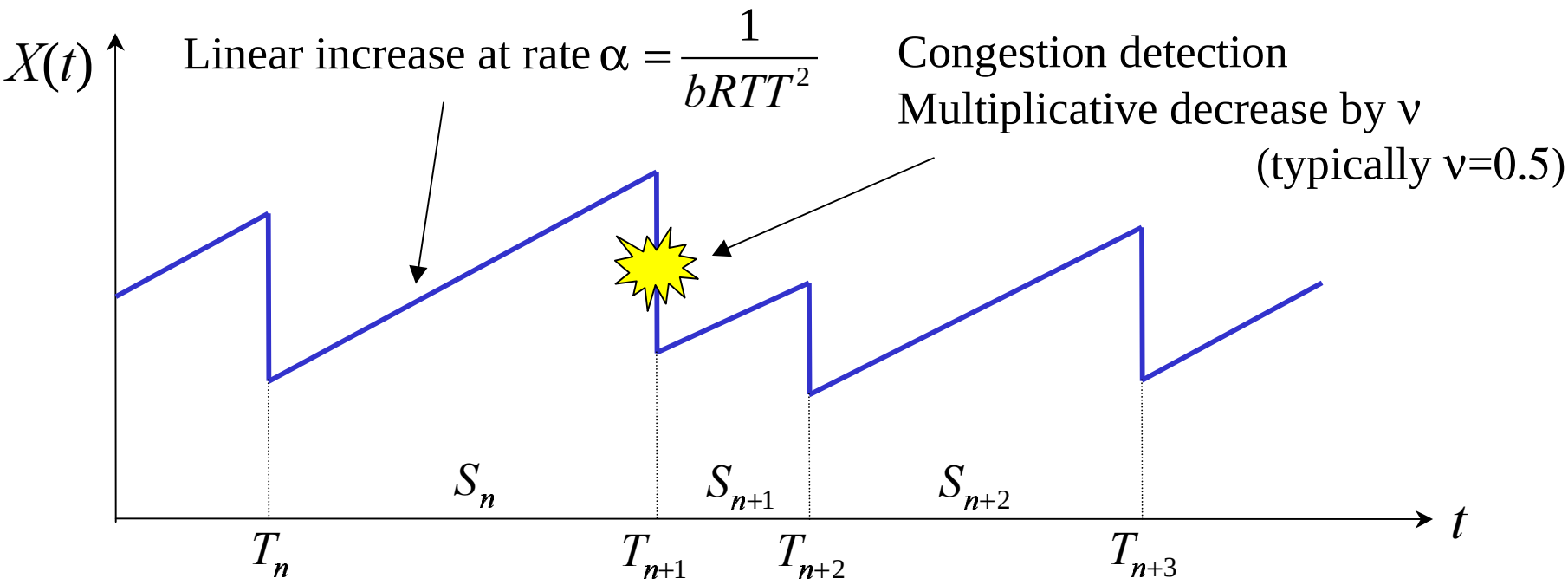
- A model for window (*or transmission rate*) evolution:
  - How the window increases between congestion events?
  - How the window decreases during congestion?
- A stochastic characterization of congestion events (e.g., Poisson).

Using the theory of stochastic processes, one can find the throughput. E.g.,

$$\overline{X} = \lim_{t \to \infty} \frac{1}{t} \int_0^t X(\tau) d\tau$$

$X(t)$: the transmission rate of the connection.

Fluid model for TCP dynamics

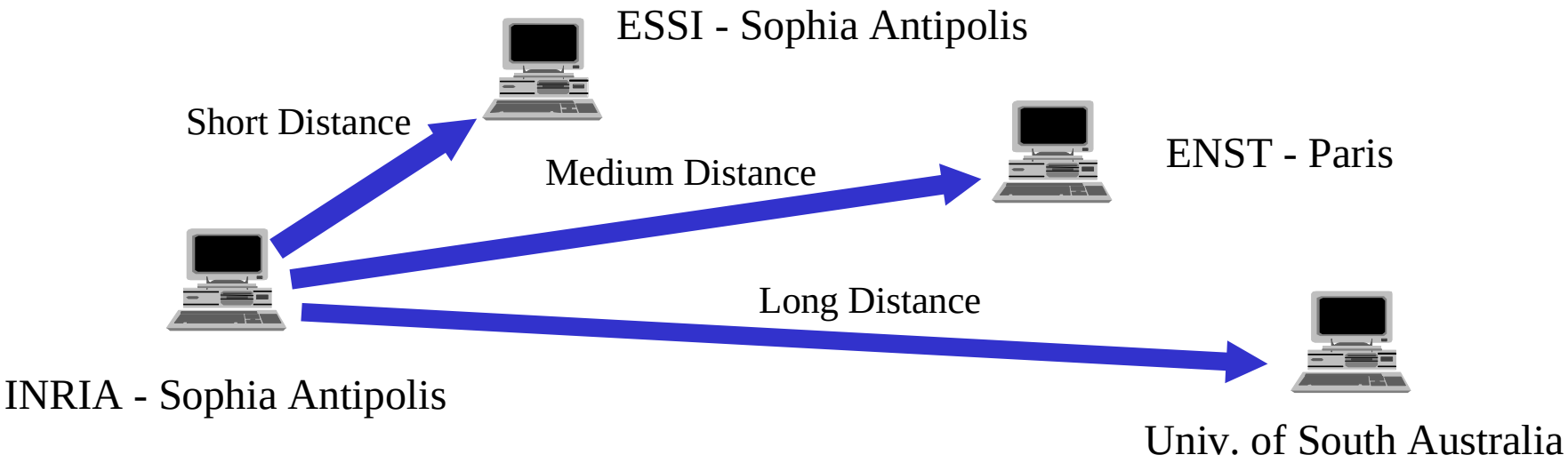


- Use the average round-trip time (*RTT*) in the analysis.
- Ignore the slow start phase.
- For the moment, ignore the timeouts, the discrete nature of TCP, and the limitation caused by the receiver window.

Model for congestion events

- **Objective:** Find a stochastic characterization for the process $\{T_n\}_{n=-\infty}^{+\infty}$.
- **Literature:** Simplistic models …
  - Deterministic, e.g., [Mathis, Semke, Mahdavi, 1997].
  - Homogenous Poisson, e.g., [Misra, Gong, Towsly, 1999].
  - Poisson with rate dependent on TCP window, e.g., [Savari, Telatar, 1999].
- **Reality:** The process of congestion events changes of law from one Internet path to another and with time on the same path:
  - How much the process of congestion events changes in reality?
  - Find a general model that covers most if not all real processes.
  - Solve the general model for the throughput of TCP and find how much a wrong assumption on congestion events impacts the approximation.

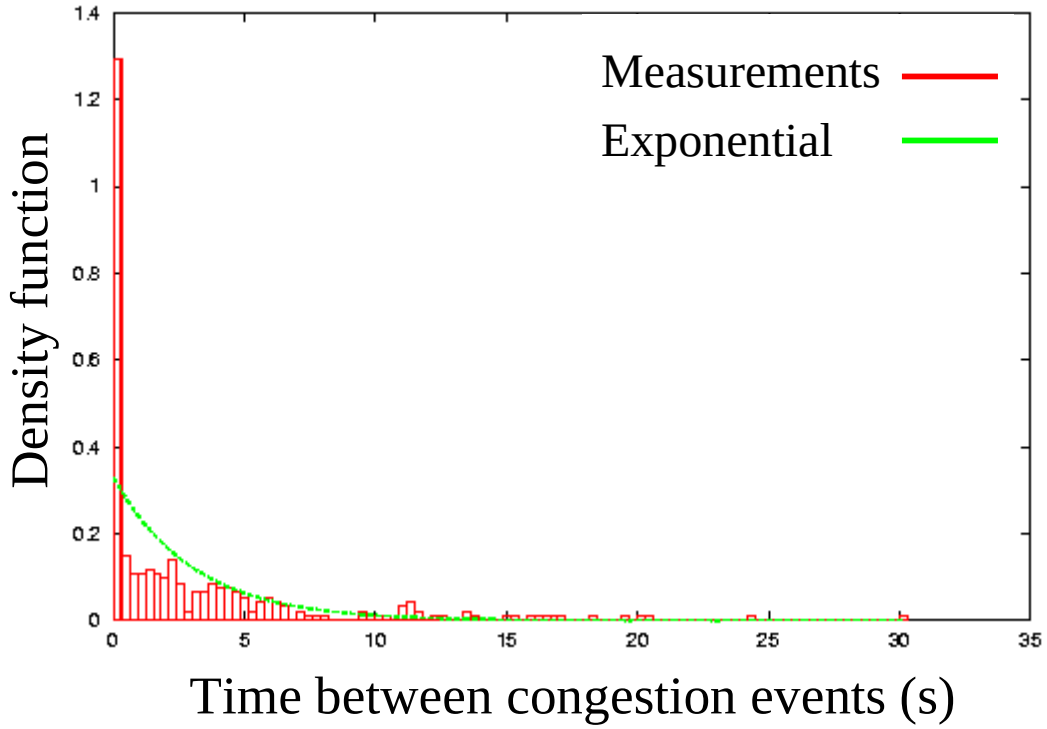
Measurement testbed



- Three long-life unlimited-data TCP transfers (NewReno version of TCP).
- Develop and run a tool at INRIA that detects congestion events.
- Store traces in separate files at fixed intervals (20, 40, and 60 min).

Some time distributions

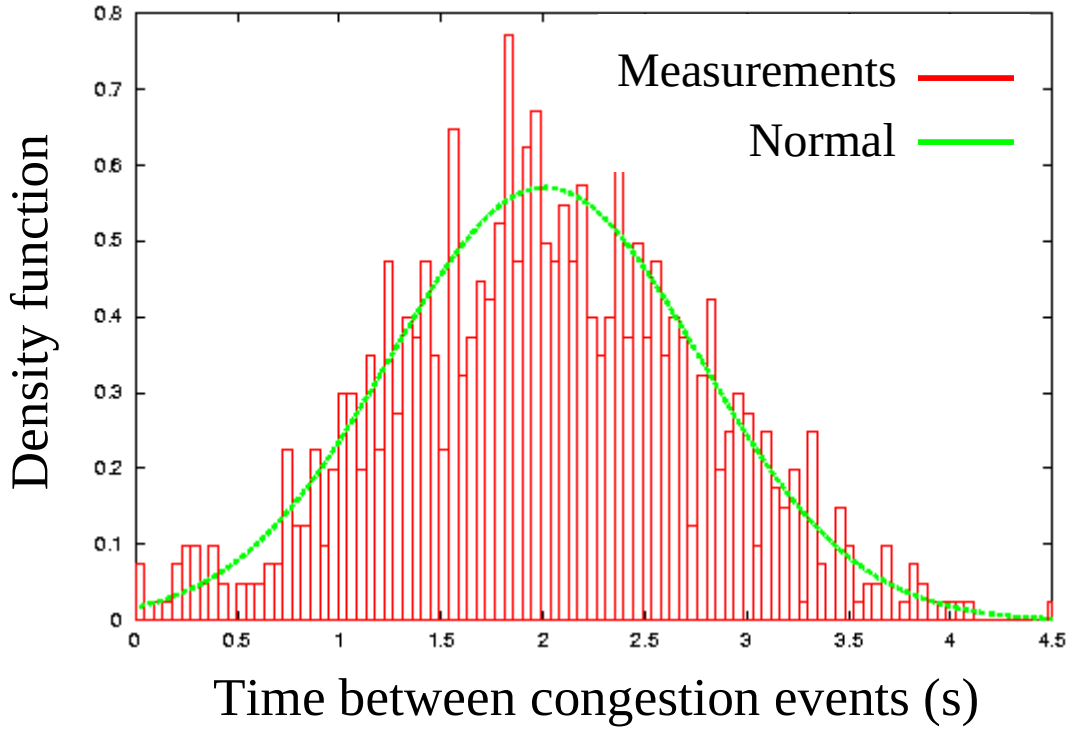
Short-Distance Connection



Highly bursty

Some time distributions

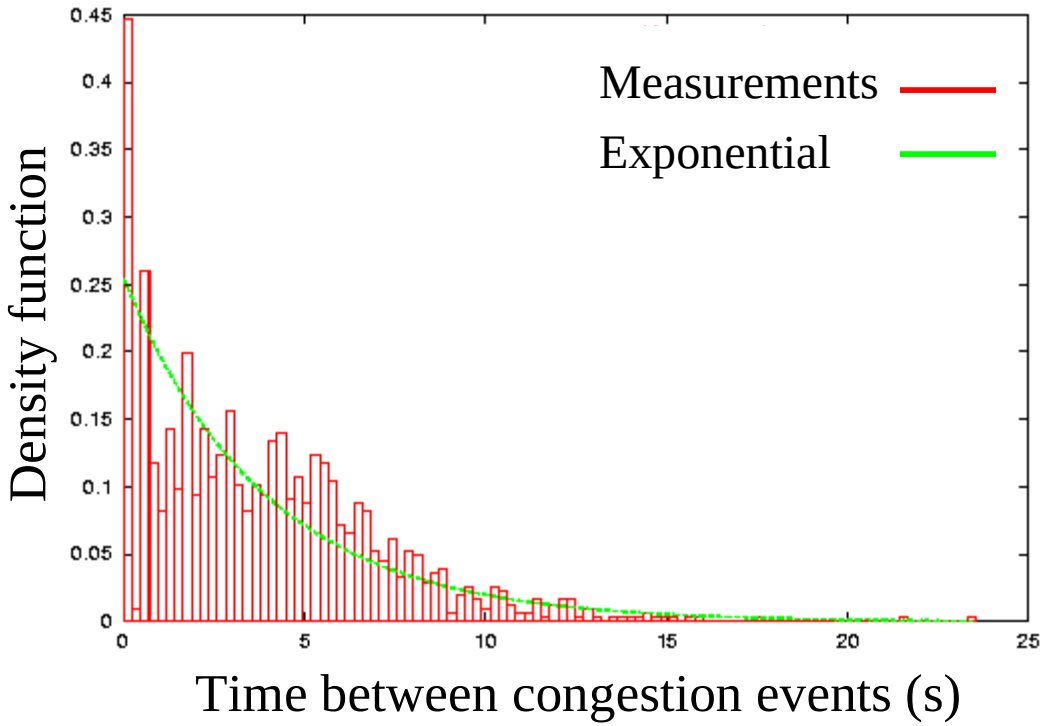
Medium-Distance Connection



Close to Normal

Some time distributions

Long-Distance Connection



As expected:

Close to Exponential

Some correlation coefficients

Short-Distance Connection

Hour (Traces of 20 min)	Covariance coefficient $Cov(S_n, S_{n+1})/Var(S_n)$
11:00	+ 0.034
12:00	+ 0.041
12:30	+ 0.113
13:00	+ 0.001
13:30	- 0.191
14:00	- 0.078

Long-Distance Connection

Hour (Traces of 60 min)	Covariance coefficient $Cov(S_n, S_{n+1})/Var(S_n)$
11:00	- 0.197
12:00	- 0.001
14:00	- 0.102
16:00	- 0.107
20:00	+ 0.023
22:00	- 0.09

-1 < Covariance coefficient < +1

☞ Higher correlation is expected on other paths with more significant memory (e.g., wireless links, satellite links, paths with long-range dependent traffic).

Our models for congestion events

- Consider models that converge to a stationary ergodic regime.
- A Markovian model:
  - The path of the connection changes between different states according to a Markov chain.
  - The rate of congestion events changes with the state of the path.
  - Similar to a Markov Arrival Process.
- A general model:
  - Times between congestion events can follow any distribution (general average and variance).
  - All types of correlation between congestion events are allowed (infinite number of general covariance functions).

Analysis: Stochastic Difference Equation

- X_n = transmission rate at the instant of congestion.
- **Stochastic Difference Equation:**

$$X_{n+1} = \nu \; X_n + \alpha \; S_n$$

- **Theorem:** Assuming that the process of congestion events is stationary ergodic, the rate of TCP converges to the same stationary regime for any initial state, and this regime is ergodic as well.

  - For T_0 in the stationary regime: $$X_0 = \alpha \sum_{j=0}^{\infty} \nu^{\,j} S_{-j-1}$$

  - Due to ergodicity: $$\overline{X} = E[X(t)]$$

    Expectation of the rate at time t in the stationary regime.

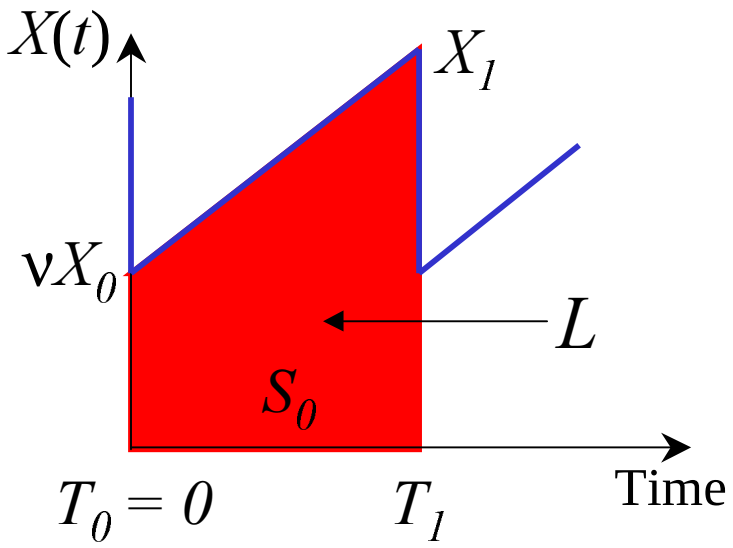
Analysis: Calculation of the throughput

- Use the following inversion formula of Palm theory:

$$\overline{X} = E[X(t)] = \lambda E^0[L]$$

λ: Rate of congestion events

- **Calculation of the throughput:**
  - $$L = \nu X_0 S_0 + \frac{1}{2}\alpha\ S_0^2$$
  - Find the expectation of $X_0 S_0$ using the Stochastic Difference Equation and its solution in the stationary regime.

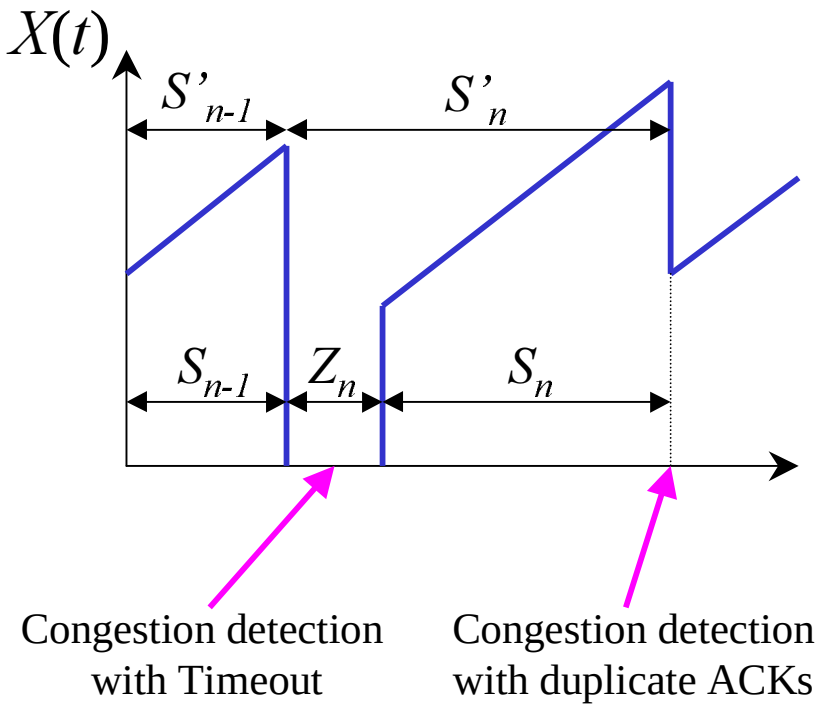


Stationary regime

- The other moments of X(t) can also be found in the same way.

Heuristic for timeouts

☞**Timeout** = A long idle time before the resumption of the transmission.



☞What we calculated until now:

$\overline{X}$ = Throughput when excluding timeouts.

☞When including timeouts:

$$\overline{X}_f = \frac{\overline{X}}{1+\lambda \times Q \times Z}$$

$$Q = \mathbf{P}\{Z_n > 0\}$$

$$Z = \boldsymbol{E}[Z_n \mid Z_n > 0]$$

We use for these two functions the values calculated by the people of UMASS in their SIGCOMM paper 98.

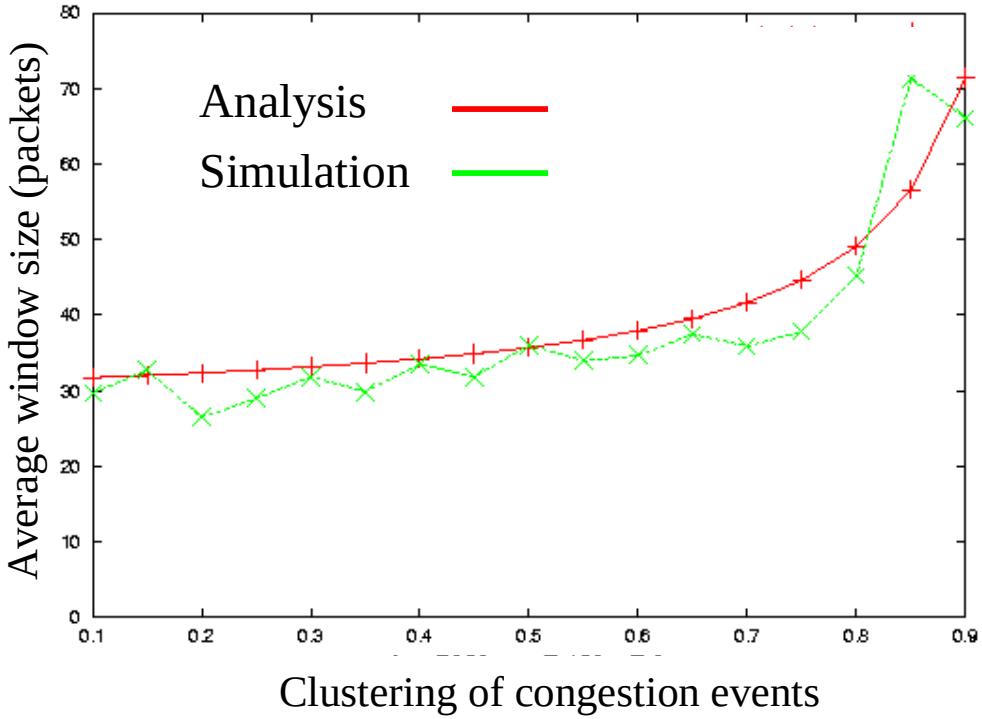
Results: Markovian model

- Simple expression of TCP throughput for a multi-state Markovian path:
  - Rate of congestion events in each state of the path.
  - Transition probabilities of the Markov chain.

- For a path of two states, the throughput of TCP increases when the Markov chain transits less frequently.

Result:

TCP realizes more throughput when congestion events are clustered than when they are uniformly distributed.

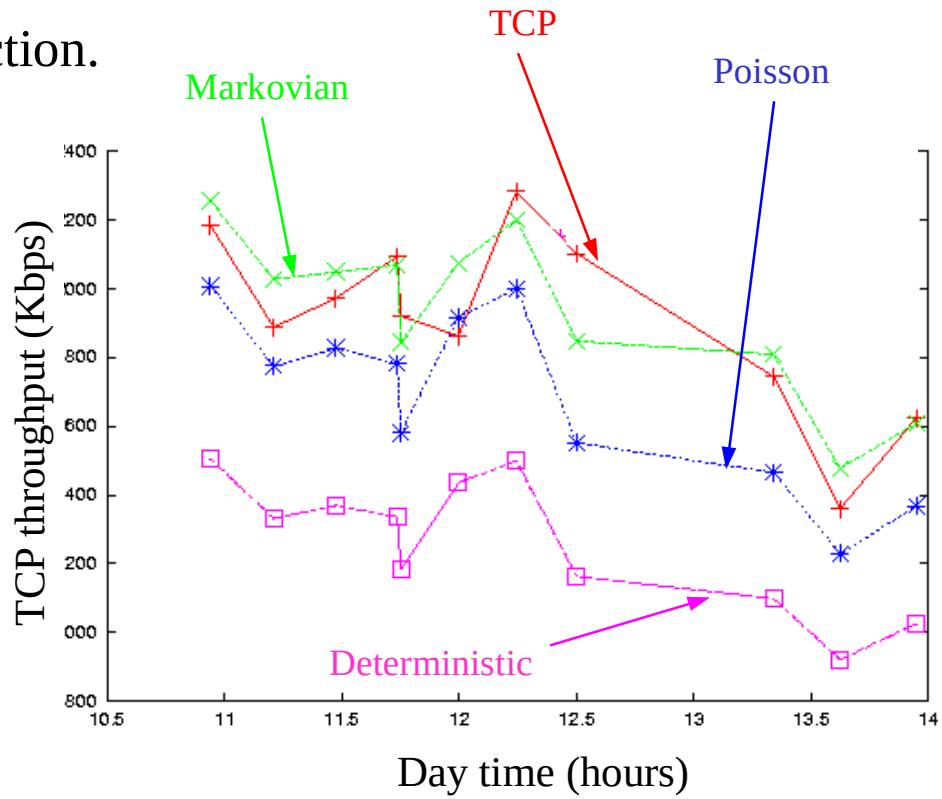


Results: Markovian model

- Proposition of a technique to infer the parameters of the Markovian model,
  - Transition probabilities of the Markov chain,
  - Rates of congestion events,

  from the real traces of a TCP connection.

- Using our inference technique and a Markov chain of two states, we validate the model on the short-distance connection where the congestion events are highly clustered.



Results: General model

Generalization of the famous *square root formula* found under the assumption that congestion events are *deterministic*.

Loss probability of a TCP packet : $p = \overline{X} / \lambda$

Previous result

Our general result

$$\overline{X} = \frac{1}{RTT\sqrt{bp}} \sqrt{\frac{1+\nu}{2(1-\nu)} + \frac{1}{2}\hat{V} + \sum_{k=1}^{\infty} \nu^k \hat{C}(k)}$$

For $\nu = 0.5$: $\overline{X} = \frac{1}{RTT}\sqrt{\frac{3}{2bp}}$

Due to average rate of congestion events

Due to variance

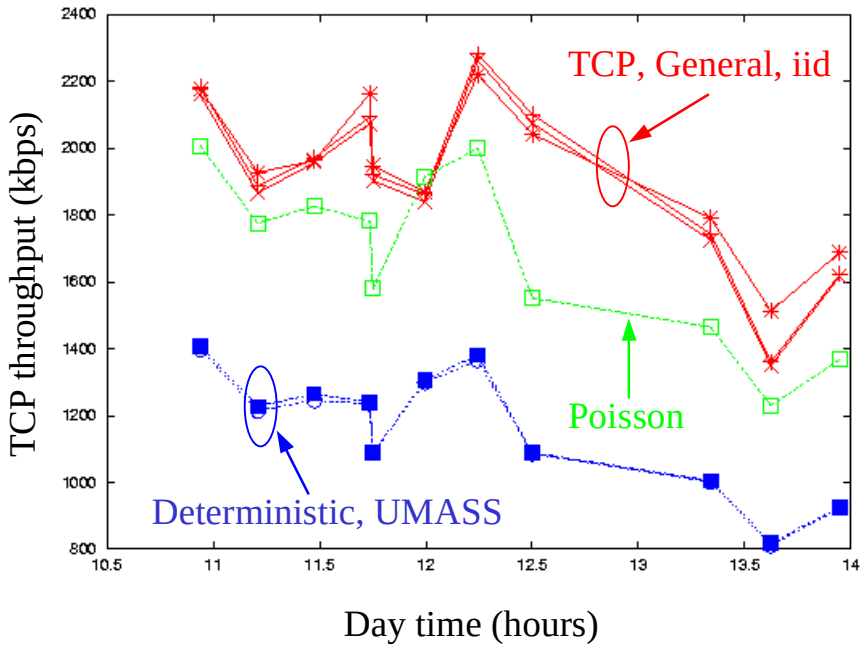
Due to correlation

Specification of the general model

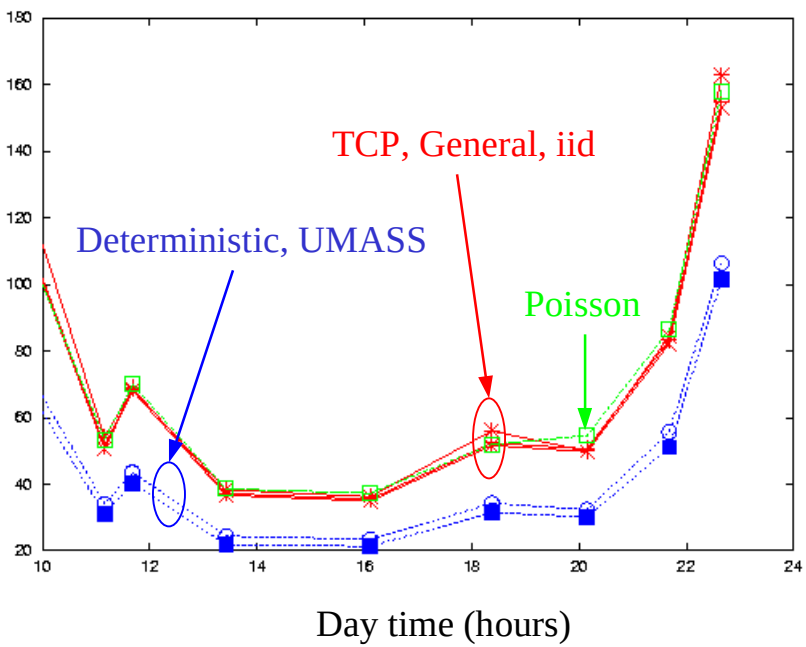
- Deterministic process: $\overline{X} = \frac{1}{RTT}\sqrt{\frac{3}{2bp}}$ (square root formula)
- Poisson process: $\overline{X} = \frac{1}{RTT}\sqrt{\frac{2}{bp}}$
- General renewal process: $\overline{X} = \frac{1}{RTT}\sqrt{\frac{1}{bp}\left(\frac{3}{2} + \frac{1}{2}\frac{Var(S_n)}{E[S_n]^2}\right)}$
  - TCP throughput is an increasing function of the variance of time intervals between (or the clustering of) congestion events.
- Markov Arrival Process (see [Altman, Avrachenkov, Barakat, SIGCOMM'00]):
  - The characteristics of the process of congestion events seen by the TCP connection depend on the state of the path which has a Markovian evolution. Congestion events may occur between and during state transitions.

Choice of the process of cong. events

Short-Distance Connection

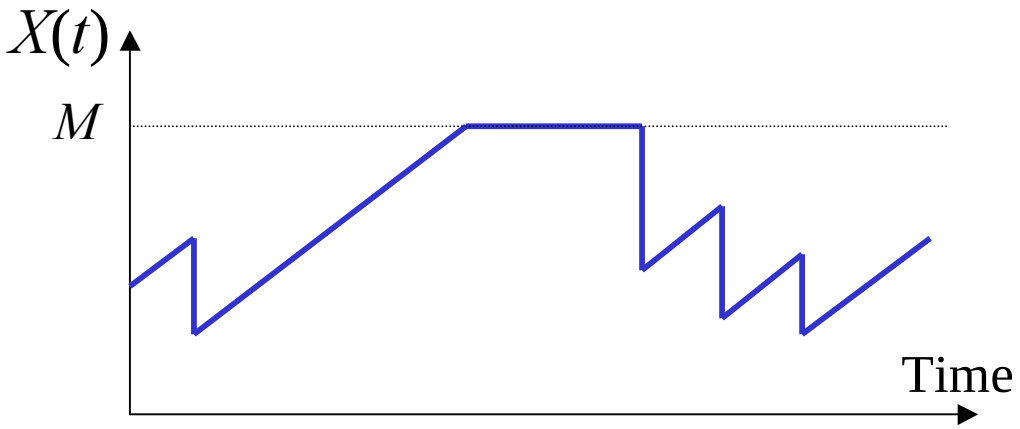


Long-Distance Connection



Modeling of receiver window

- Exact calculation of TCP throughput seems to be impossible for a general process of congestion events.



- **Approximations:**
  - Markovian model: Approximation of the throughput using a fixed-point approach similar to that used by the people of UMASS.
  - General model : Calculation of bounds for the throughput which are also good approximations.
- **Exact calculation:**
  - Particular case of congestion events that follow a Poisson process.

Analysis: Recurrent equation for moments

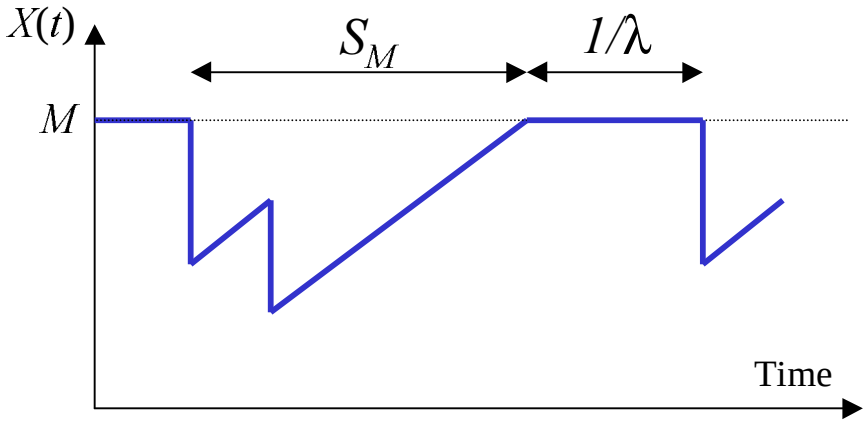
- **Theorem:** For a stationary process of congestion events, the rate of TCP converges to the same stationary regime for any initial state.

- **Analysis:** $F(x)$ = PDF of $X(t)$ in the stationary regime,
  - Write the Kolmogorov equation for $F(x)$,
  - Apply the Laplace Transform to the equation,
  - Differentiate and solve for moments,

$$\text{for } k = 1,2, \ldots, \quad E[X^k(t)] = \frac{k\alpha\left(E[X^{k-1}(t)] - P_M M^{k-1}\right)}{\lambda(1-\nu^k)}$$

Unknown : $P_M = \mathbf{P}\{X(t) = M\}$ ...

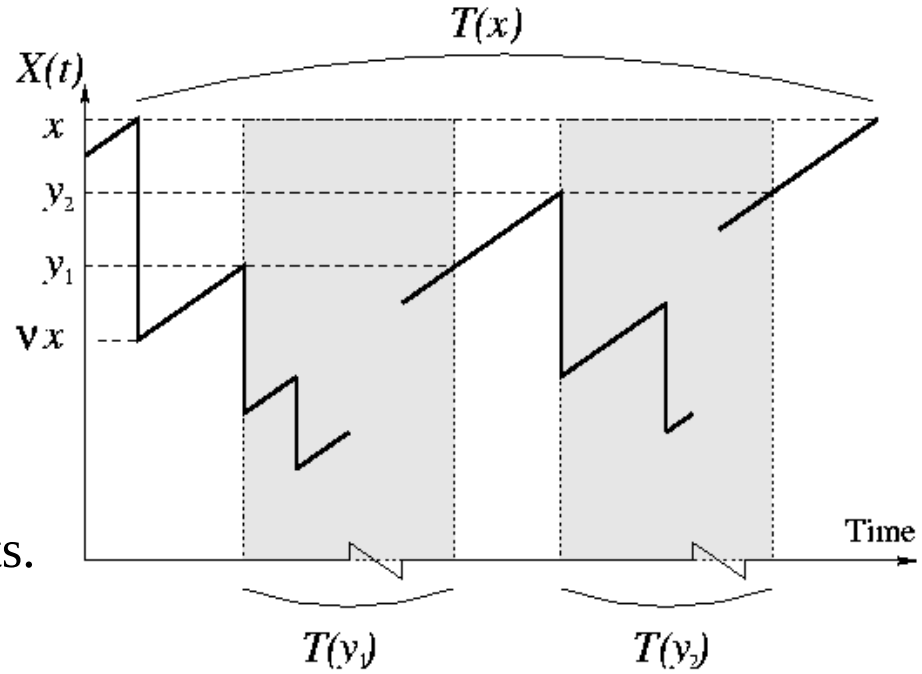
Analysis: Calculation of P_M



$$P_M = \frac{1/\lambda}{1/\lambda + S_M}$$

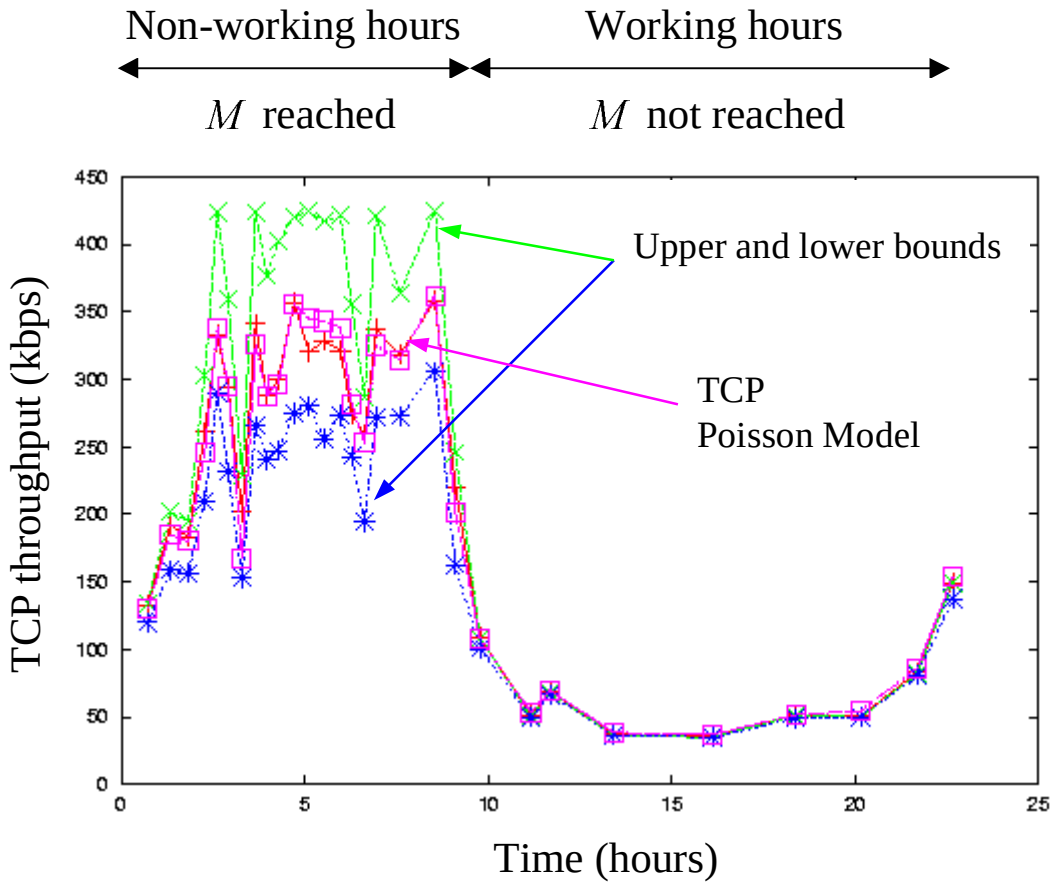
$$S_M = E[T(M)]$$

- Write an implicit equation for $E[T(x)]$,
- Solve it with Laplace Transforms,
- Calculate P_M and hence all the moments.



Modeling of receiver window: Validation

Long-Distance Connection: Receiver window of 32 Kbytes.



End-to-end modeling: Open issues

- Refinement of throughput expressions:
  - Paths where RTT cannot be substituted by its average.
  - Introduction of slow start.
  - More complicated processes for the case of window limitation.
  - Specification of our general result to more particular paths (*e.g., paths where the rate of congestion events is an increasing function of the window size*).

.
.

- Application of our results:
  - End-to-end applications that use throughput expressions (*e.g., TCP-Friendly Rate Control mechanism* [Floyd et al., SIGCOMM'00]).
  - Optimization of TCP congestion control parameters (*to improve the fairness of TCP, to reduce the variation of TCP window without adding to the aggressiveness, etc*.).

.
.

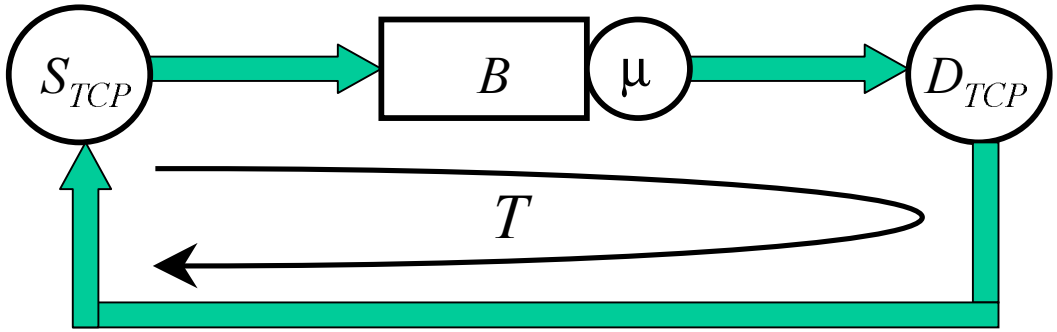
Network-specific models

- Large bandwidth-delay product paths (*e.g., satellite networks*)
- Paths with asymmetric bandwidth (*e.g., hybrid satellite networks*)
- Paths with non-congestion losses at the link-level (*e.g., wireless networks*)

Large BDP paths

- **Main problem:** Long time taken by the slow start phase.
- **Possible solution:** Accelerate the window increase (e.g., Byte Counting).

  But, this increases the aggressiveness of TCP.
- **Question:** What is the optimal window increase policy?
- **Model:**



  - Find the maximum window increase rate that does not overload the router ...

Large BDP paths: Results

- For a given B, we found the maximal window increase rate.

  And for a given window increase rate, we found the optimal B.

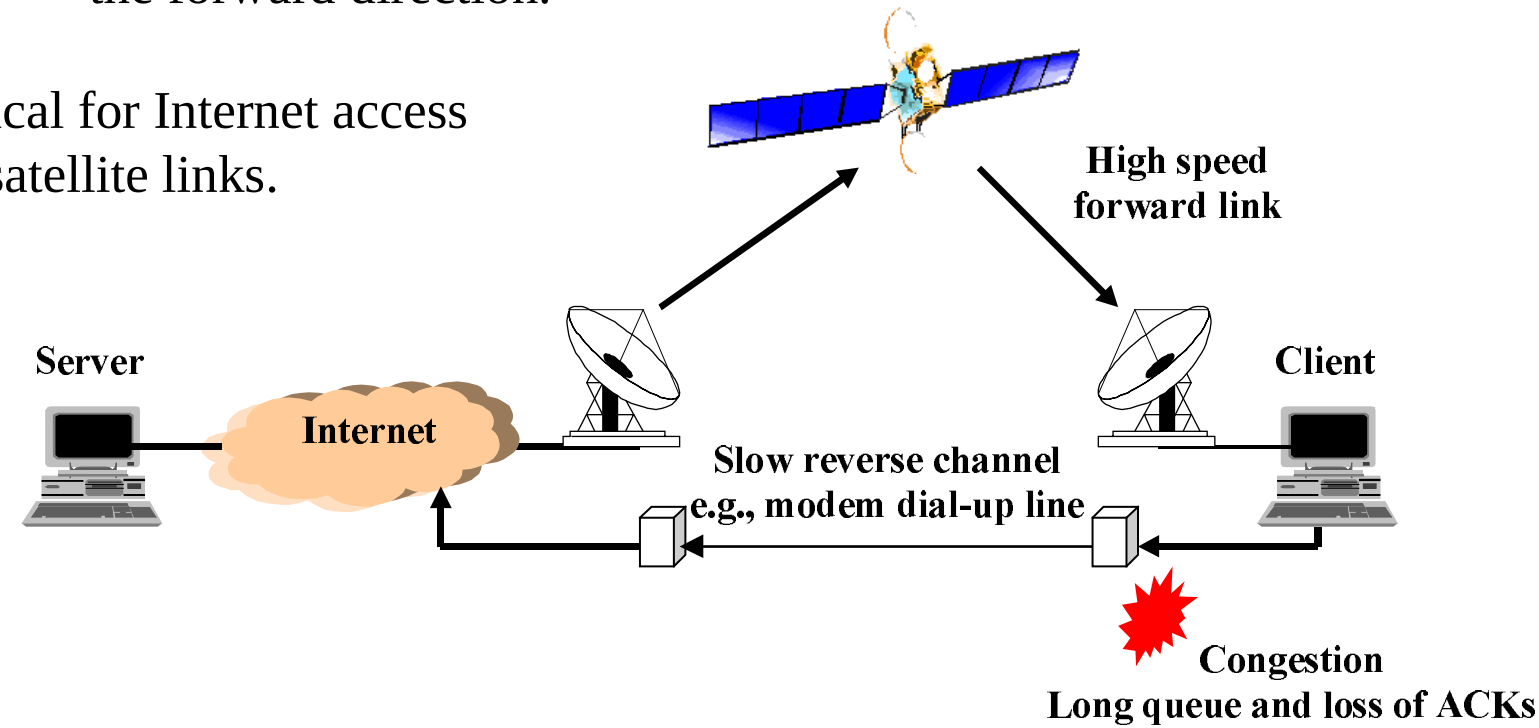
- Current window increase policy is not optimal:
  - Incrementing the window by the same amount for all ACKs results in a conservative behavior at small window sizes.

- **Proposition:** Go faster in slow start at small window sizes.
  - **Case study:** Switch from Byte Counting ($W=W+2.PacketSize$) to Standard TCP ($W=W+PacketSize$) as long as we progress in slow start.
  - **Result:** Better performance without much adding to the aggressiveness.

Asymmetric networks

- **Definition:** When the reverse channel (Client ⇨ Server) is not able to carry the flow of ACKs resulting from a good utilization of the bandwidth in the forward direction.

- Typical for Internet access via satellite links.



- **Problem:** The congestion of the reverse path which results in an increase in the round-trip time and in problems of fairness.

ACK filtering as a solution

- **Idea:** Given that the information carried by ACKs is cumulative, substitute an old ACK in the congested buffer by a new one.
  - **Result:** Reduction to the minimum of the queue length.
  - **But**, ACKs are also used to increase the window, so

New objectives

The maximum number of ACKs must be passed to TCP sources, and

The reverse path must be fairly shared between the different connections.

- **Proposed mechanism:** Delay the filtering of ACKs of a TCP connection until
  - The reverse path is fully utilized.
  - The connection achieves its fair share of the scarce bandwidth.
  - **Result:** Faster window increase without adding to the congestion.

Wireless networks

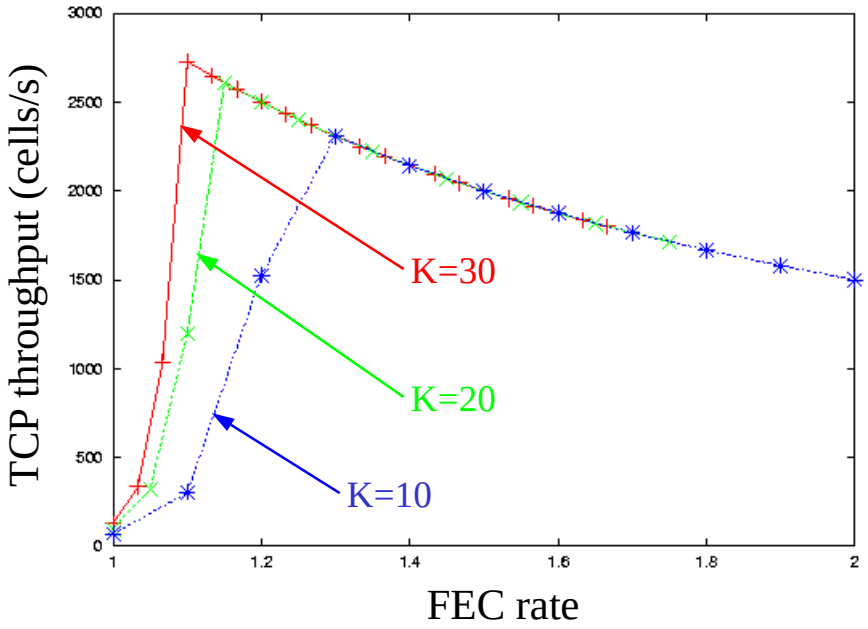
- **Problem:** Transmission errors (*non-congestion losses*) which are interpreted by TCP as congestion signals.
- **Solution:** Improve the quality of the wireless link with Forward Error Correction (*i.e., transmit redundant information on the wireless link*).

  But, much redundancy may steal some of the bandwidth used by TCP.
- **Question:** What is the appropriate amount of redundancy?
- **Model:**
  - A two-state Gilbert model for the wireless link (*to study the impact of burstiness of transmission errors*)
  - A block code for FEC
  - The square root formula for TCP throughput

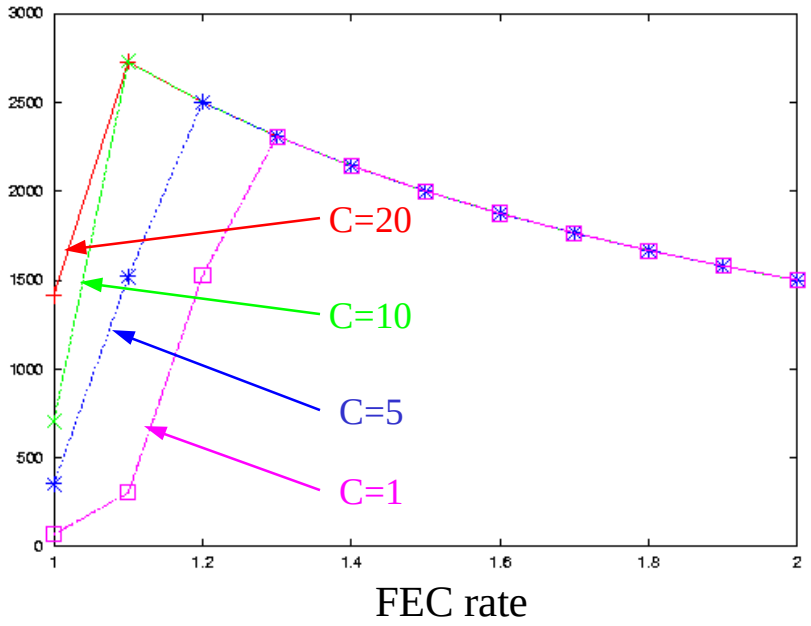
Results: Non-bursty link

Model: Long TCP connections over a 1.5 Mbps ATM wireless link

K = Packet size in ATM cells
Single TCP connection



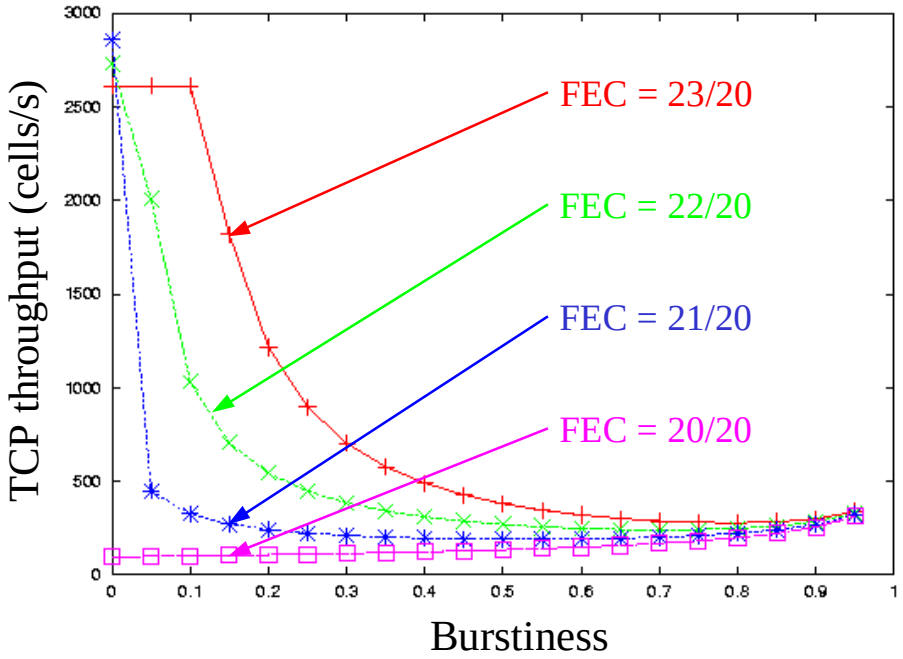
C = Number of TCP connections



Results: Bursty link

For an average loss rate of 1% :

Packet size in ATM cells = 20
Single TCP connection



Network-specific models: Open issues

- In the wait for an efficient TCP version that copes with the heterogeneity of the Internet on end-to-end basis:
  - Propose models and mechanisms to evaluate and improve the performance of TCP over any new transmission medium (e.g., shared media, ADSL).
- Detailed study of the performance of TCP in an Internet that provides different classes of services (e.g., Differentiated Services architecture).

Thank you!