Modelling TCP with Markov chains

Keith Briggs

Keith.Briggs@bt.com

more.btexact.com/people/briggsk2/

2002 June 17

typeset in LaTeX2e on a linux system

- Markov chain basics
- Mini-TCP toy model
- Markov chain perturbations
- Results for mini-TCP
- NewReno analysis

- To build accurate models of various TCP algorithms
- To analyze them and compare mean behaviour such as throughput
- To study short-term behaviour (transient response)
- To generalize to non-Markovian dynamics
- To design new TCPs for multicast, wireless etc.
- Software tool for easy analysis of modified algorithms?

- Sits between IP layer and applications
- Provides rate control and reliable transmission
- Full duplex
- Adjusts send rate to adapt to congestion and receiver buffer
- Adjust rate seeing only ACKs from receiver

BSD early 1980s

Jacobson 1988

(old)Tahoe, (new)Reno, Westwood,

MSWin (proprietary), Linux from BSD

bugs?

TCP terminology

segment (packet): 20 byte header, up to 64K bytes data

window: continuous burst of segments

timeout: indicates lost packet

congestion window: current estimate of maximum window size to avoid congestion

threshold: crossover point between slow-start (exponential) phase and congestion-avoidance phase

TCP segment header

source	destination
sequence number	
acknowledgement number	
flags	window size
checksum	urgent
options	
data	

Markov chain terminology 1

States $\mathbb{R}$

P: matrix of transition probabilities

P P

P P : positive, stochastic matrix

Q $P \Rightarrow Q$

Stationary distribution :

$$Q \Rightarrow P$$

Normalization:

Fundamental matrix: F Q Q

To compute :

find 1d kernel (rank(Q)= , e.g. by LU) of Q

or iterate $\leftarrow P$

My idea: P and therefore everything else depends on a loss probability , so do all calculations in computer algebra as power series in (Even the iteration!)

Toy model to explain Markov chain method:

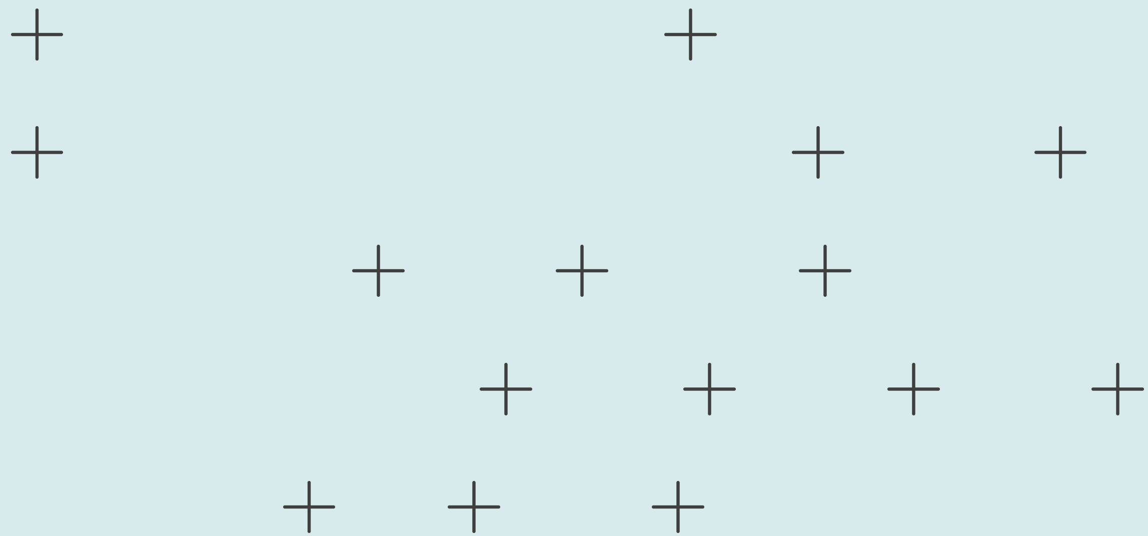
Window

$$\begin{cases} \quad & \text{if packet lost} \\ \quad + \quad & \text{else} \end{cases}$$

(with , loss probability= ,)

$$P \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

My software computes:



My software computes:

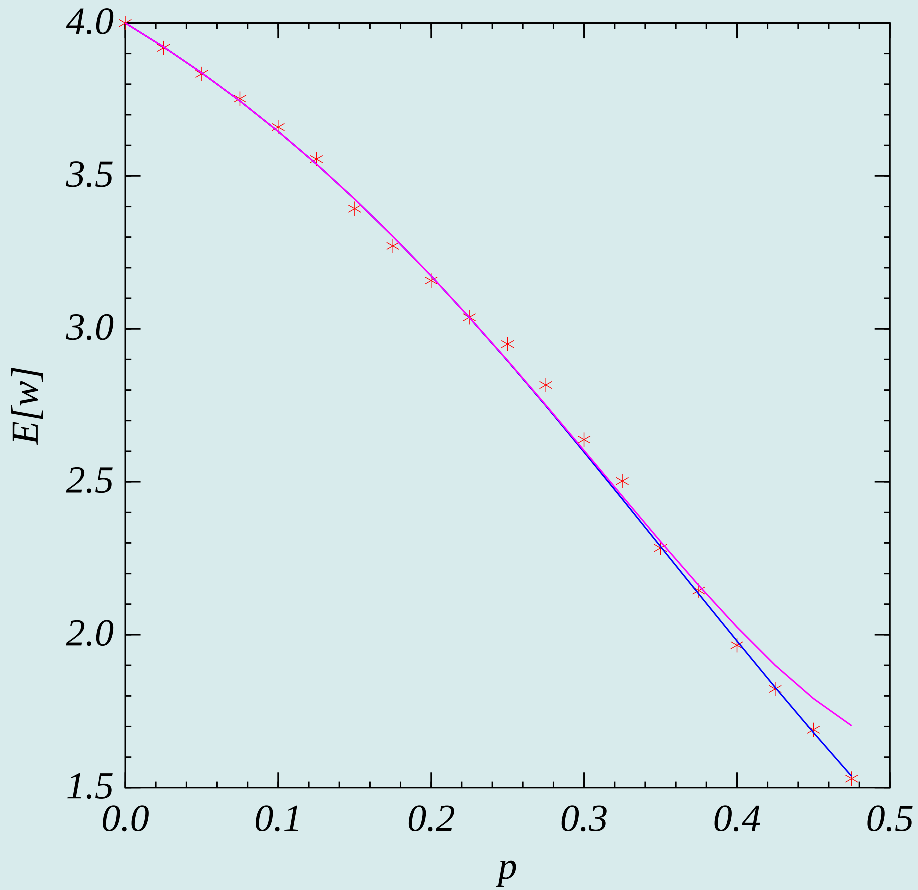
Expectation value of window size:

E

$+ \quad + \quad + \quad +$

Compare exact formula: $\overline{\qquad\qquad}$

mini-TCP 5



Expected window size:

exact, series, simulation *.

Do not store matrix; just compute elements as needed

P becomes a function P

Recall fundamental matrix: $F \quad Q$

Define *group inverse* $Q \quad$ of Q to be F

Then $\quad PQ$

Thus can compute all required quantities

Markov chain perturbation theory 1

$P \to P + \Delta \quad P$, how is stationary distribution changed?

$\to$

$$\Delta F$$

$$\leqslant \Delta \quad F$$

Entrywise:

$$\frac{}{} \Rightarrow \frac{}{} + \mathcal{O}$$

Markov chain perturbation theory 2

$$\leqslant \quad \Delta \qquad \text{tr} \;\; Q$$

$$\leqslant \quad \Delta \qquad\qquad\qquad Q$$

F is dense! E.g. row 0:

$$+ \quad + \qquad\qquad + \mathcal{O}$$

$$+ \qquad\qquad\qquad + \mathcal{O}$$

$$+ \qquad + \mathcal{O}$$

$$+ \qquad + \mathcal{O}$$

$$+ \quad + \qquad + \qquad + \mathcal{O}$$

Can avoid density in practice by computing products F

For mini-TCP: derivative of wrt :

$$+ \quad + \mathcal{O}$$

$$+ \quad + \mathcal{O}$$

$$+ \mathcal{O}$$

$$+ \quad + \mathcal{O}$$

$$+ \quad + \quad + \quad + \mathcal{O}$$

State variables:

- $\leqslant \quad \leqslant$: window size
- : $\Rightarrow$ first of two rounds; $\Rightarrow$ second
- $\leqslant$: lost packets in previous round
- $\leqslant \quad \leqslant$: backoff factor in previous round was

No packets lost:

P $\leqslant$

P $+$ $\leqslant$

P

One or more packets lost:

P

$\leqslant \quad \leqslant \quad \leqslant$

$P \quad \leqslant \quad \leqslant$

$P \quad \leqslant$

$P \quad \sum \quad \leqslant$

Exponential backoff (for $\leqslant$ $\leqslant$):

$$P \qquad + $$

$$P$$

With , this gives 71 allowed states out of a possible states, giving a sparsity of about 5%

NewReno TCP 5 - typical output

Let M be the matrix whose element is the mean time for first transition from state to state . Then

$$M \quad Q \quad + \quad \text{diag} \quad Q \quad \text{diag} \quad QQ$$

Mini-TCP:

$$M \quad \begin{bmatrix} + \\ + \\ + \\ + \\ + \end{bmatrix}$$

Can now analyze more-or-less automatically any TCP version

Try new algorithms specialized for multicast or peer-to-peer?

S L Campbell and C D Meyer, *Generalized inverses of linear transformations*, Dover 1991.

J Padhye et al., *A stochastic model of TCP Reno congestion avoidance and control*, University of Massachusetts. CMPSCI Technical report 99-02.