Energy Efficiency of TCP in a Local Wireless Environment

Presented by Mansoor Jafry

Overview

- Motivation
- Quick Recap of TCP Versions
- System Model
- Analysis
- Performance Results
- Summary

Motivation

- Increasing trend of portable and mobile devices
- Reliance of efficient usage of limited battery power
- Battery technology is a slowly improving field
- Need to exploit other avenues for saving power

Internet Applications on Mobile Devices

- The Transport Control Protocol (TCP) lies at the heart of internet services
- It was designed for wireline networks with low error rates
- Various modifications proposed since original deployment (Tahoe, Reno, Vegas etc.)

Peculiarities of the Wireless Environment

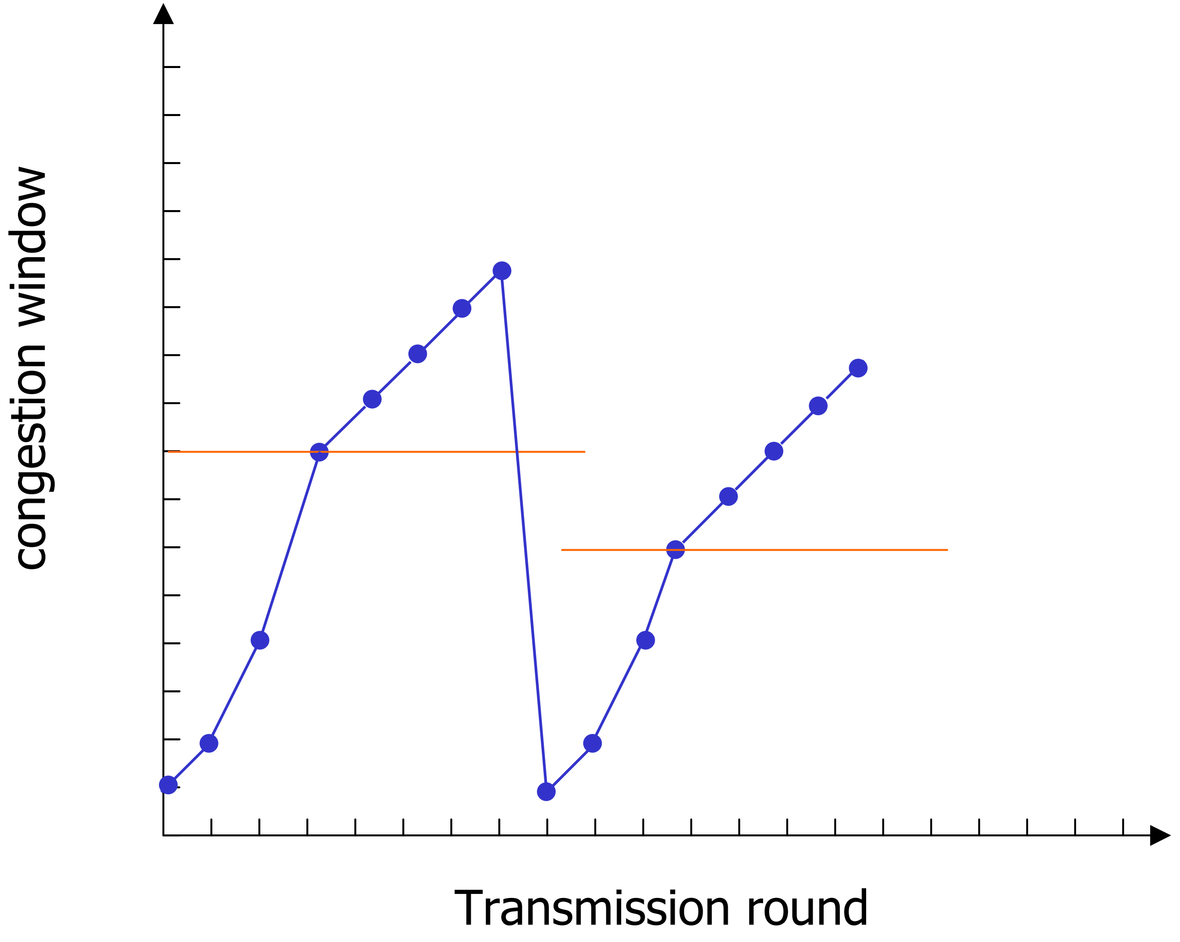
- Mismatch between assumptions and true causes of data loss
- Packet errors are usually correlated
- Should not fight a bad channel, rather save power for better conditions

Which is exactly what TCP does anyway!

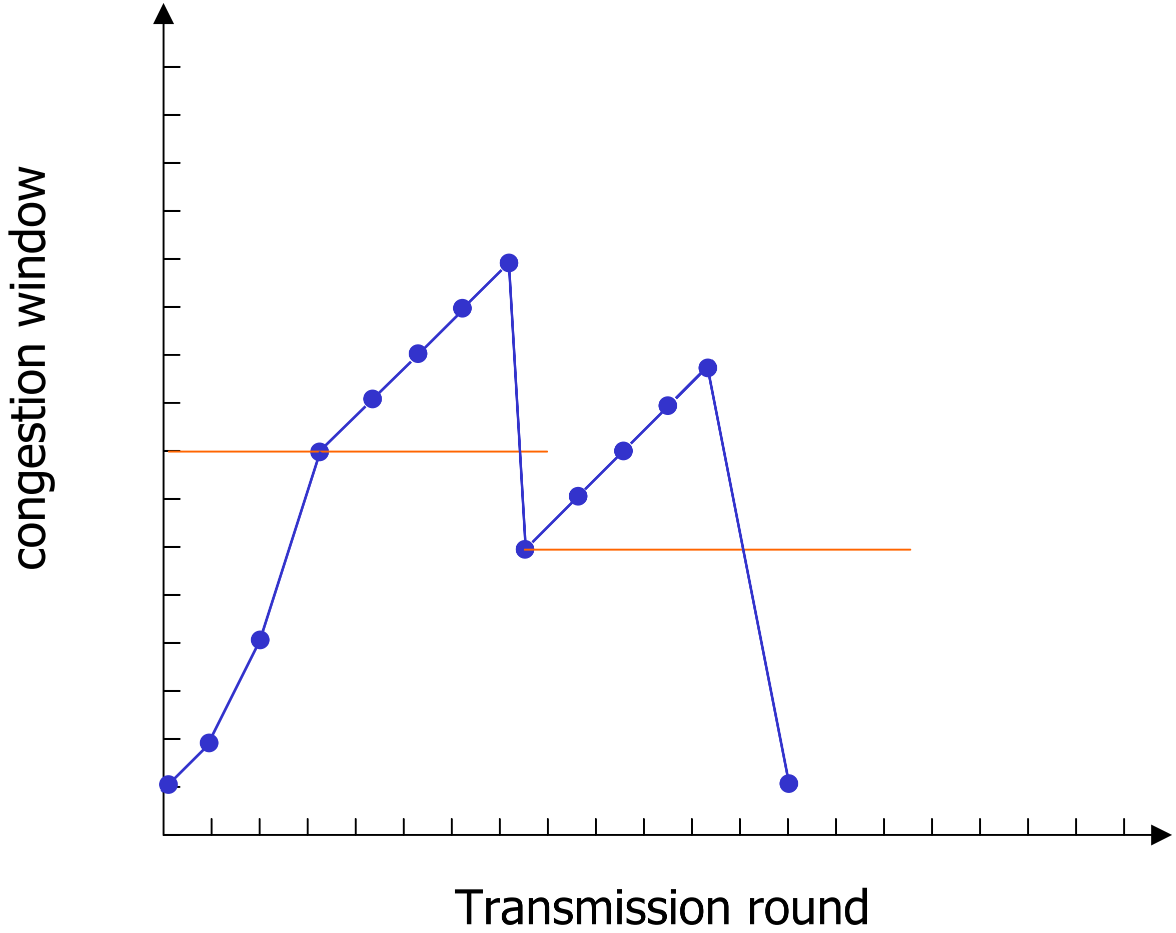
TCP Quick Reference

- TCP can accept packets out of sequence but will deliver them only in sequence
- Receiver advertises W_{max} , which limits the number of unacknowledged outstanding packets
- Correctly received packets are acknowledged with cumulative ACKs
- ACKs carry the next packet sequence number expected from the sender
- Timeouts and duplicate ACKs used to guess occurrence of packet loss

TCP Window Evolution - Timeout



TCP Window Evolution – Duplicate ACKs



TCP Window Parameters

- Let
  - W(t): sender's congestion window width at time t
  - $W_{th}(t)$: slow start threshold at time t
- The evolution of W(t) and $W_{th}(t)$ is triggered by ACKs and timeouts

TCP Basic Algorithm

- If $W(t) < W_{th}(t)$, each ACK $W(t)$ to increase by 1 (slow start)
- If $W(t) \geq W_{th}(t)$, each ACK causes $W(t)$ to increase by $1/W(t)$ (congestion avoidance)
- If a timeout occurs at time t, and t^+ is the next timeslot, then
  - $W(t^+) = 1$
  - $W_{th}(t^+) = \lceil W(t)/2 \rceil$

TCP Versions

- OldTahoe:
  - loss detection: timeout
  - loss recovery: retransmission
  - window adaptation: $W(t^+) = 1$,
    $W_{th}(t^+) = \lceil W(t)/2 \rceil$

- Tahoe:
  - loss detection: timeout or duplicate ACKs ☞ **fast retransmit**
  - loss recovery: retransmission
  - window adaptation: $W(t^+) = 1$,
    $W_{th}(t^+) = \lceil W(t)/2 \rceil$

TCP Versions (2)

- Reno:
  - loss detection: timeout or duplicate ACKs ☞ **fast retransmit**
  - loss recovery and window adaptation:
    - on timeout, similar to Tahoe
    - on duplicate ACKs: ☞ **fast recovery**
      - $W_{th}(t^+) = \lceil W(t)/2 \rceil$
      - $W(t^+) = W_{th}(t^+)$
      - Transmits only the first lost packet
- NewReno
  - loss detection: as in Reno
  - recovery and adaptation: as in Tahoe

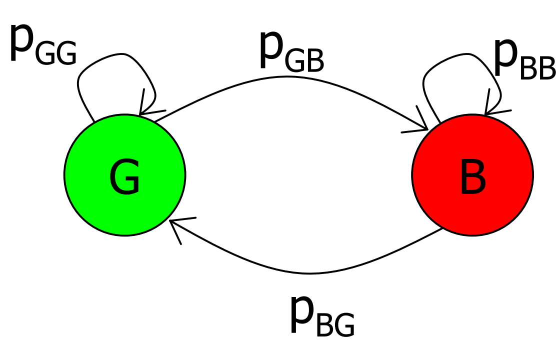
System Model

Goal

- We want to model the performance of TCP…
  - during bulk data transfer (we can ignore connection setup and teardown overheads)
  - over a *single* TCP connection
  - where the transmitter always has some data to send

Error Model

- Fading channels are generally difficult to model accurately
- This approach adopts a two-state Markov model because of sufficient accuracy and analytical simplicity



$$\text{transition matrix} = \begin{bmatrix} p_{BB} & p_{BG} \\ p_{GB} & p_{GG} \end{bmatrix}$$

Error Model (2)

- Avg. probability of a packet loss, ε, depend on physical characteristics of the channel
  - Fading margin, *F*
  - Normalized Doppler frequency $f_D T$

Approach

- Analytical approach
  - ☺ Completely parameterized and fast
  - ☹ Relies on several inaccurate assumptions
- Simulation
  - ☺ Better approximation of reality
  - ☹ Excessive runtime
  - ☹ Is affected by parameters like size of packet
- Hybrid Approach
  - Obtain packet error trace by simulating fading process
  - Use it to estimate avg. packet error rate and avg. length of burst

Analysis

For the case of TCP Tahoe

Analysis – Reward Renewal Process

- Many stochastic processes have the property of regenerating at certain timeslots
- Behavior after a regeneration epoch is probabilistic replica of the initial behavior
- Long-term behavior can be studied in terms of behavior during a single regeneration cycle

Parameters

- Parameters **W**, **W_{th}** and the **channel state** evolve in cycles between two loss detection events
- We define t_k as the slot immediately following the detection of a packet loss.
- t_k , t_k+1, … $t_{k+1}-1$, t_{k+1}, $t_{k+1}+1$, …

Semi-Markov Process

- We define a random process $X(k) = \big(C(t_k\text{-}1), W(t_k), W_{th}(t_k) \big)$
- $\Omega X = \{(C, W_{th}, 1) \mid C = B, G,\ 1 \leq W_{th} \leq \lceil W_{max}/2 \rceil \}$
- Future evolution of process X(m), m>k, is independent of the past X(m), m<k
- Given a Markov chain it is always possible to define a semi-Markov process which admits the original chain as its embedded Markov chain

Metrics of Interest

- Introduce metrics on transitions to track
  - Number of slots, $\mathbf{N_d}$
  - Number of transmission attempts: $\mathbf{N_t}$
  - Number of successful transmissions: $\mathbf{N_s}$
- We follow the evolution of the embedded Markov chain while cumulating the metrics on each transition

Approach

- We divide the cycle into two phases
  - $t_k+n+1, \ldots, t_{k+1}-1$
- Probability distribution of n is given by
  $\alpha_C(n) = P[\text{first error at } t=t_k+n \mid C(t_k)=C]$

$$= \begin{cases} p_{CB}, & n=1 \\ p_{CG}p^{n-2}p_{GB}, & n>1 \end{cases}$$

Transition Matrix

- Crucial ingredient: size of window at t_k+n
  - $Y(k) = W(t_k+n)$
  - $\Omega_Y = [1, W_{max}]$
- We can study the two phases separately, obtaining two matrix transition functions $\Phi^{(1)}(z)$ and $\Phi^{(2)}(z)$
  - *ij* th entry of $\Phi^{(1)}(z)$?
  - *jk* th entry of $\Phi^{(2)}(z)$?

Transition Matrix (2)

- Matrix transition function
  - $\Phi(z) = \Phi^{(1)}(z)\ \Phi^{(2)}(z)$
  - z is actually a vector of variables being tracked
- Let $\xi_{ij}(N_d, N_t, N_s)$ be the probability the system makes a transition to state j
  - in exactly N_d slots, with N_t transmissions
  - of which N_s are success
- $\Phi_{ij}(z_d,z_t,z_s) = \sum_{N_d, N_t, N_s} \xi_{ij}(N_d, N_t, N_s)\ z_d^{Nd} z_t^{Nt} z_s^{Ns}$

Transition Matrix (3)

- Matrix of average delays (slots) will be

$$D = \frac{\partial \, \Phi(z_d, z_t, z_s)}{\partial \, z_d}$$

- The averages of other quantities, *T* and *S*, can be found similarly

Reward Renewal Theory

- If A(k) and B(k) cumulative metrics during the first k cycles, then

$$\lim_{k\to\infty} \frac{A(k)}{B(k)} = \frac{E[A]}{E[B]} = \frac{\sum_{i\in\Omega x} \pi_i \sum_{i\in\Omega x} P_{ij} A_{ij}}{\sum_{i\in\Omega x} \pi_i \sum_{i\in\Omega x} P_{ij} B_{ij}}$$

  - π_i: steady state probabilities of Markov chain with transition Matrix P
  - A_{ij}, B_{ij}: averages of the corresponding metrics during transition ij

Computation of Metrics

- Various metrics can be computed by setting A and B appropriately
  - A=S, B=D : avg. success per slot *(throughput)*
  - A=T, B=D: avg. transmissions per slot *(system load)*
  - A=S, B=T: average number of successes per transmission *(success probability)*

*

Accounting for Idle Time

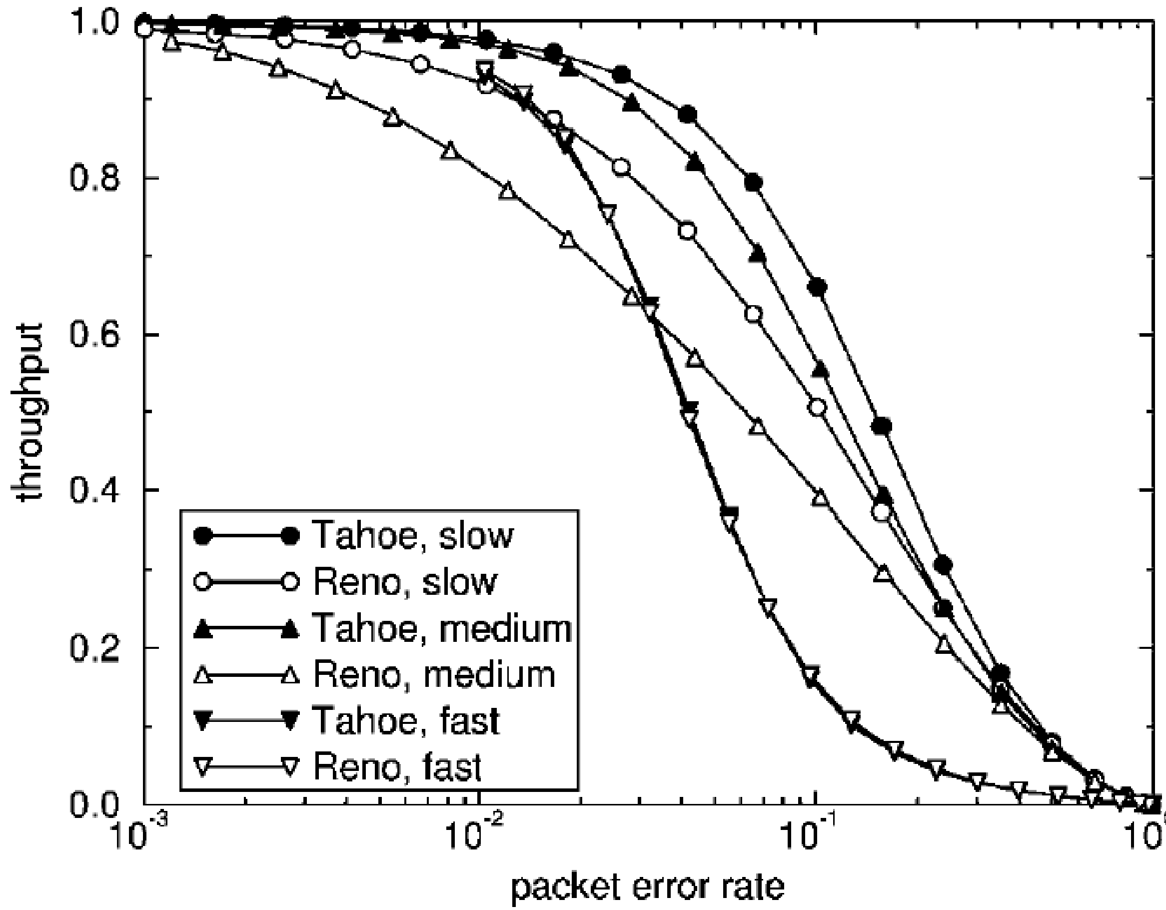
- In reality terminals also consume power when idle
- Let $C = (F+w_a)T + w_i(D-T)$
  - C: matrix of avg. energy consumption per transition
  - F: energy consumed during one transmission
  - w_a,w_i: energy consumed by rest of the circuitry during active and idle slots

Energy Efficiency Formulation

$$\text{energy efficiency} = \frac{\sum_{i\in\Omega_X} \pi_i \sum_{j\in\Omega_X} P_{ij} S_{ij}}{\sum_{i\in\Omega_X} \pi_i \sum_{j\in\Omega_X} P_{ij} C_{ij}}.$$

Performance Results

Throughput vs. Error Rate

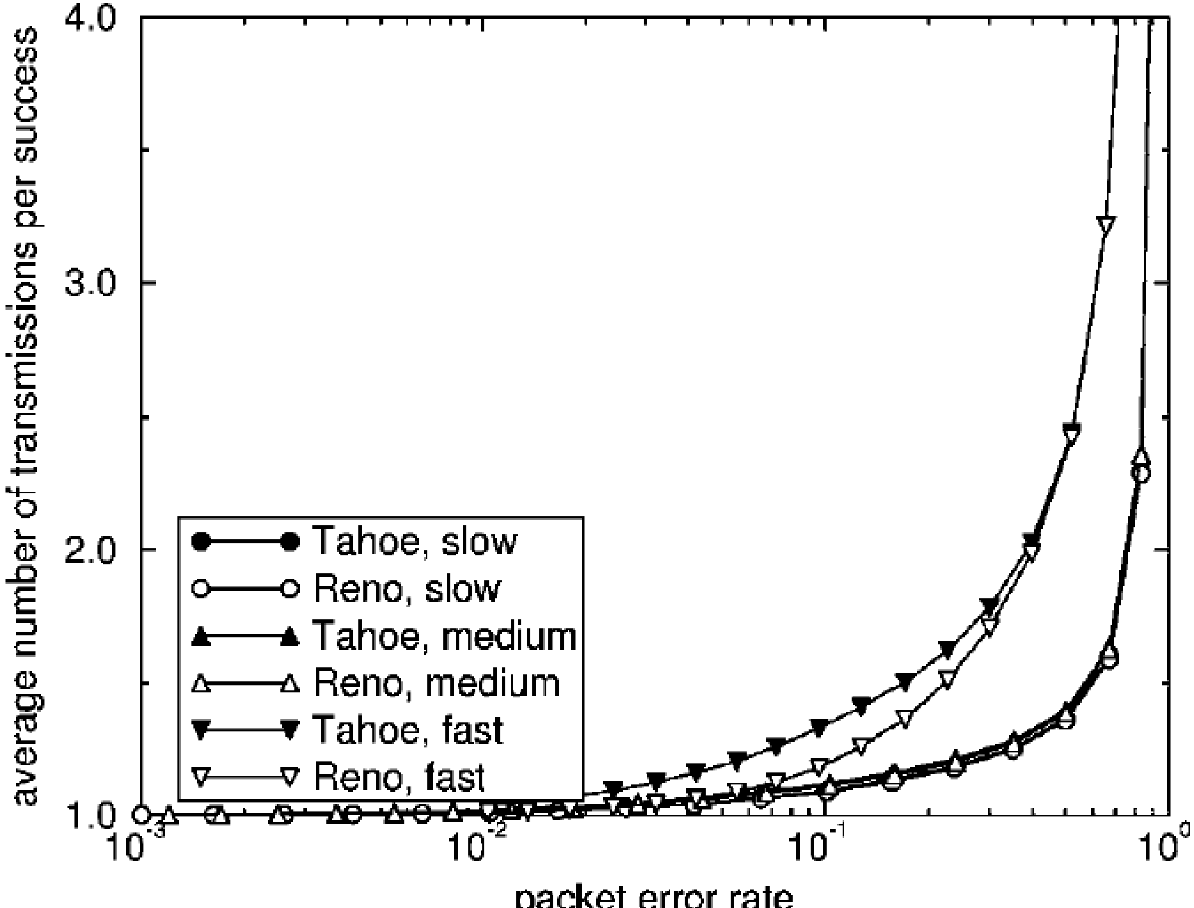


W_{max}=24, Packet size=1024 bytes fdT=0.001 (slow), 0.016 (medium), 0.256 (fast)

Observations

- For large error rates, throughput is higher for slower fading (error clustering)
- Reno performs worse than Tahoe in virtually all cases
- Information that can't be seen here:
  - Results of Tahoe improve upon increasing W_{max} (not true for Reno)

Transmission attempts per success

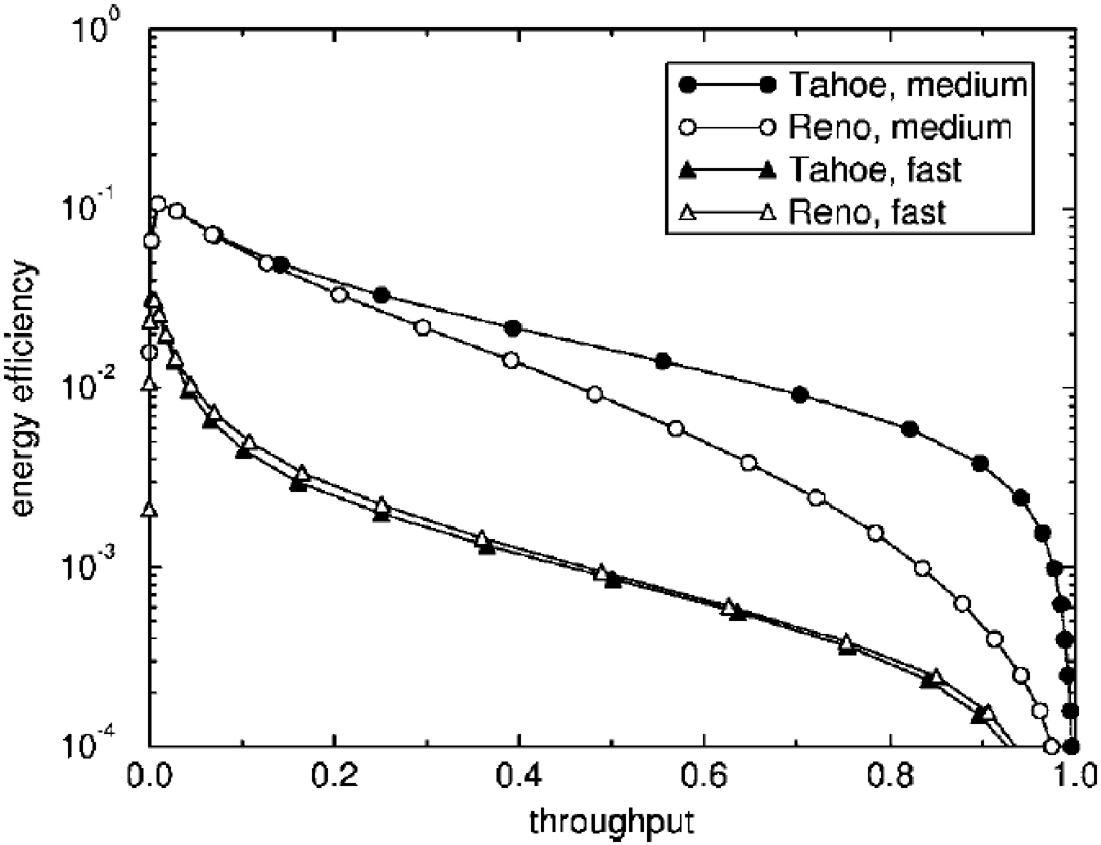


W_{max}=24, Packet size=1024 bytes fdT=0.001 (slow), 0.016 (medium), 0.256 (fast)

Observations (2)

- Tells us how much energy sent to get a packet across
- Unlike throughput, this metric is not very sensitive to error correlation
- Reno (although worse in throughput) totals fewer transmission attempts per packet
- Interestingly, this metric is also insensitive to W_{max}

Energy efficiency vs. throughput

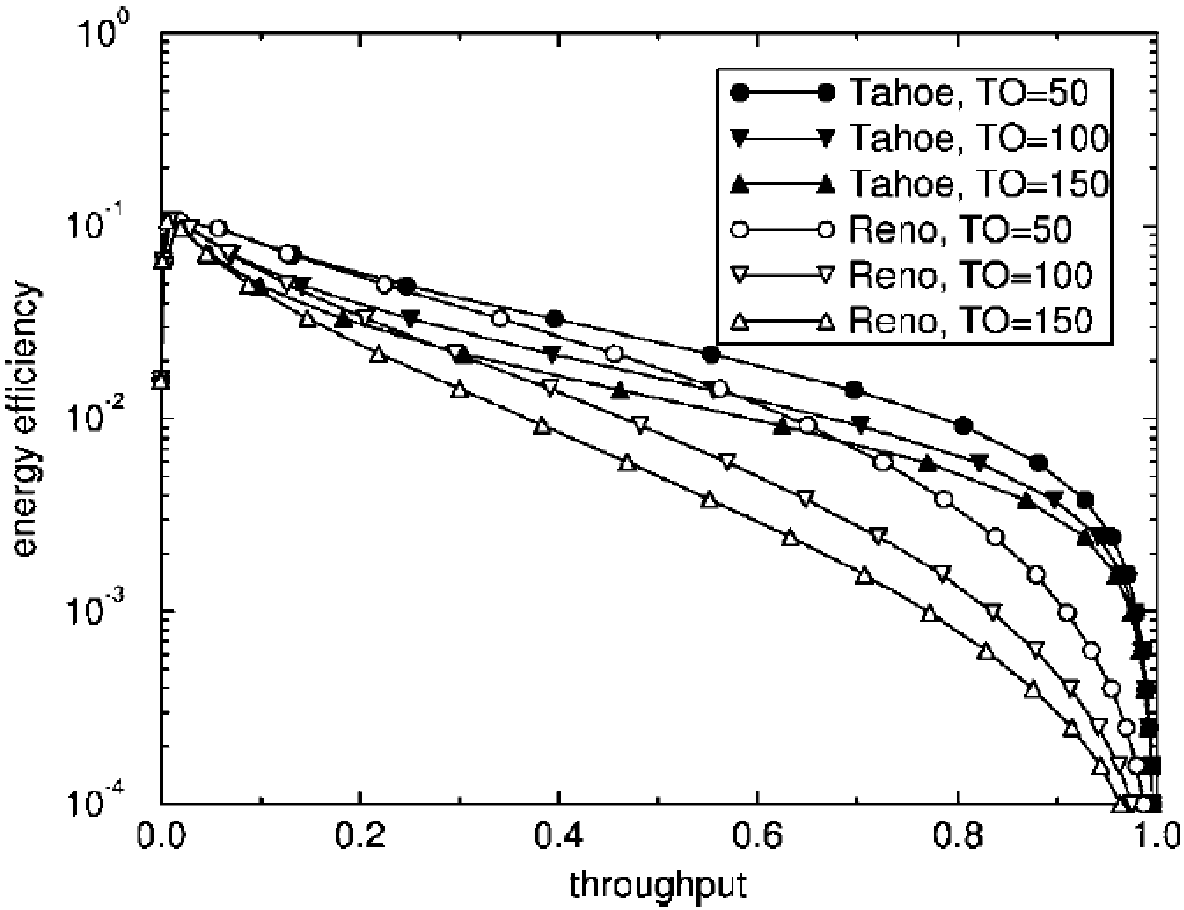


W_{max}=24, Packet size=1024 bytes fdT=0.016 (medium), 0.256 (fast)

Main Findings and Summary

- TCP Reno performs the poorest, while Tahoe and NewReno exhibit similar performance (for fast fading NewReno is better)
- As fading rate increases, energy efficiency suffers
- Shorter timeouts result in better performance in general. Reno is more sensitive to timeouts than Tahoe
- A larger W_{max} allows to fully exploit advantages of correlated errors

Timeouts



W_{max}=24, Packet size=1024 bytes fdT=0.001 (slow), 0.016 (medium), 0.256 (fast)

Questions & Answers