International Journal of Parallel, Emergent and Distributed Systems
Vol. 23, No. 4, August 2008, 309–324

Exploiting energy-saving potential in heterogeneous networks

Lefteris Mamatas* and Vassilis Tsaoussidis

Department of Electrical & Computer Engineering, Democritus University of Thrace, Vas Sofias 12, 68100 Xanthi, Greece

(*Received 23 October 2006; final version received 29 May 2007*)

We investigate the energy-saving potential of transport protocols. We seek an answer to strategic issues of maximising energy and bandwidth exploitation, without damaging the dynamics of multiple-flow equilibrium. We claim that (i) an energy-saving strategy of the transport level needs to be associated with some energy potential index which, unlike energy expenditure, is not device-specific and (ii) system-wise an energy-efficient system of flows is not always a better choice: we show that a less energy-efficient system may be more reliable in terms of packet multiplexing and, in turn, may reduce the probability that some flows may expend their energy with zero gain. We perform experiments using a real testbed and ns-2 based simulations.

Keywords: energy efficiency; extra energy efficiency; risk index; UAR; energy potential

1. Introduction

Energy consumption is becoming a crucial factor for wireless, *ad hoc* and sensor networks, which affects system connectivity and lifetime. Standard transmission control protocol (TCP), originally designed for wired network infrastructure, does not cope with wireless conditions such as fading channels, shadowing effects and handoffs, which influence energy consumption.

We investigate energy efficiency from two perspectives:

(i) The energy-saving potential of the communication mechanism.
(ii) The risk of a flow to expend its energy for minor gains due to the multiplexing limitations. In particular, we investigate whether increased energy-saving capabilities may result in further unfair behaviour. Since we associate energy expenditure not only with data transmission but also with time, unfair behaviour translates into energy expenditure with minor performance gains.

Wireless network interface cards usually have four basic states of operation and each of these states has different power requirements. The most power-demanding states are the active states where transmission and reception of data take place. The standby/listen state, is the state where a network interface card is simply waiting. The extended period of idle state may lead to a sleep state, which is the least power-demanding state, where the radio subsystem of the wireless interface is turned off. Note that the transition mechanism itself is also energy consuming. Regardless of the states, their number and the frequency of transition, energy consumption is itself device-specific.

---

*Corresponding author. Email: emamatas@ee.duth.gr

ISSN 1744-5760 print/ISSN 1744-5779 online

DOI: 10.1080/17445760801930922
http://www.informaworld.com

Due to the complexity of energy management and the fact that the state transition is device-specific, each transmission or reception attempt by a higher-layer protocol does not necessarily correspond to a similar power transition. That is, we cannot accept *a priori* that the measured energy expenditure reflects the ability of a protocol to administer energy resources. Therefore, we distinguish protocol energy potential from actual device expenditure. The former approaches the latter when the sophistication of devices increases in a manner that all network layers collaborate ideally. Otherwise, if higher-layer protocol operation is suspended but the power module does not adjust, the protocol potential cannot translate into energy efficiency.

Since the network interface is a significant consumer of power, considerable research has been devoted to energy efficient design of the entire network protocol stack of wireless networks [5]. Several attempts have been made to measure the energy efficiency of transport protocols, e.g. [1,2,6,8,12], as well as their potential for energy efficiency [10]. Energy efficiency is clearly device-specific while energy potential is not clearly defined. We attempt to define the latter by introducing a corresponding index; we also attempt to measure actual expenditure using specific device characteristics.

Furthermore, we noticed at this stage of our investigation some interesting results. While protocol *Goodput* is an important factor for energy efficiency (as we have also shown in Ref. [10]), protocol fairness is another key factor for usability, which in turn determines the amount of flows that receive bad or zero service. In this context, fairness also associates with energy: bad or zero service does not translate into minor or zero energy expenditure.

Consider a scenario where a system exhibits unfair behaviour. Practically, some flows are favoured, while some others are not. We show experimentally that a system with increased energy efficiency does not guarantee better results for its users, but instead, the potential risk for a flow to receive bad or zero service is increased. We introduce an experimental metric, named risk index (RI), which captures this behaviour.

The structure of this paper is the following: In section 2, we discuss protocol strategies. In section 3, we choose metrics for experimental analysis. Additionally, we introduce and discuss the energy potential (EP) and RI. In section 4, we detail our experimental methodology and evaluation plan. In sections 5 and 6, we present our experimental results and we conclude the paper.

2. On protocol strategies

Energy cost due to communication relates with:

(i) The effort that the protocol expends (in terms of data transmission rate).
(ii) The amount of time required for the completion of communication.

In general, energy-consumption is the outcome of the transmission strategy that a transport protocol implements. An aggressive protocol, for example, may generate more overhead and hence, expends some extra energy due to that overhead. By the same token, a conservative protocol may expend more energy due to unexploited opportunities for successful transmission. Clearly, a sophisticated (energy-wise) protocol should alternate aggressive and conservative strategies that minimise overhead and maximise efficiency. Such sophistication requires enhanced mechanisms for detecting network dynamics.

Additive increase multiplicative decrease (AIMD) Ref. [3] allows for blind congestion control. According to AIMD, all senders keep increasing their transmission rate additively (i.e. the congestion window W increases by α packets per round-trip time), until a packet loss. When congestion is taking place (i.e. there is a packet loss), a multiplicative decrease ratio is used to avoid a congestive collapse. So, the congestion window W decreases to βW upon congestion. The standard TCP uses the values $\alpha = 1$ and $\beta = 0.5$. TCP-friendly TCP(α, β) protocols parameterise the congestion window increase value α and decrease ratio β in order to trade responsiveness for smoothness. This tradeoff guarantees friendliness to traditional TCP.

Authors of Ref. [13] introduced a simple relationship for α and β:

$$\alpha = \frac{4(1-\beta^2)}{3}. \tag{1}$$

Based on experiments, they propose $\beta = 7/8$ as the appropriate multiplicative decrease value (i.e. Less rapidly than TCP does). For $\beta = 7/8$, Equation (1) gives an increase value $\alpha = 0.31$.

At a first glance, one may think that conservativeness and aggressiveness of the window adjustment strategy can be regulated by the increase/decrease parameters α and β. However, the adjustment of parameters α, β cannot really regulate some conservative or aggressive behaviour. For example, a protocol with an increased α parameter is not always more aggressive than one with a smaller α value. An aggressive sender may trigger the timeout mechanism more times. If bursts of packets are being lost, the retransmission time-out (RTO) mechanism can suspend transmission, which indicates a conservative behaviour. We investigate when a protocol should be aggressive as well as the cost of this behaviour in terms of energy-efficiency and fairness. Since the timeout may be a conflicting factor for scheduling an aggressive behaviour[1], our adjustments of α and β are coupled with a small fixed timeout value. Practically, the trading of α for β parameter regulate the level of smoothness/responsiveness. Smoothness and responsiveness constitute a tradeoff Ref. [15]. Authors in Ref. [11] discuss the dynamics of this behaviour.

Smooth protocols may be more aggressive (since they consume temporarily more bandwidth) in the presence of transient errors, while they may behave more conservatively, due to their low increasing rate, when multiple drops force the multiplicative decrease factor to adjust the congestion window back to its initial value Ref. [11]. Consider packet drops at the end of a congestion epoch; the window decreases by a factor of $(1-\beta)$. However, multiple packet drops could cause the window size to be decreased multiple times, or they could also cause the retransmission timer to expire. At the end, it is possible for the window size and the *ssthresh* to be decreased down to two segments, even with smooth backward adjustments. Under such scenarios, the performance of applications (including real-time applications) is not affected by the rate at which the sender reduces its transmission, but rather by its capability to recover from the error and restore its sending rate. Note that our scenario is not unrealistic. For example, in mobile networks, burst correlated errors and handoffs generate this kind of error pattern.

3. Metrics for evaluating energy performance

Energy dynamics in association with protocol strategy cannot be characterised accurately based only on traditional metrics. For example Goodput captures protocol performance but not protocol effort. Goodput is defined as:

$$\text{Goodput} = \frac{\text{Original\_Data}}{\text{Connection\_Time}}, \tag{2}$$

where, Original_Data is the number of bytes delivered to the high-level protocol at the receiver (i.e. excluding retransmitted packets and overhead) and Connection_Time is the amount of time required for the corresponding data delivery.

Therefore, we complement this metric with the extra energy expenditure (EEE) metric. EEE, Ref. [6] attempts to capture the extra energy expended due to protocol operation - not just the expended energy. That is, a protocol may transmit when there are windows of opportunities for error-free transmission, without expending extra energy, or vice versa. In contrast, it may waste opportunities for transmission expending energy (even in an idle state) and extending communication time. EEE attempts to capture extra energy expenditure as an associated result of Goodput, Throughput and maximum Throughput, each one represented as a moving point on a line. The index EEE takes into account the difference of achieved Throughput from maximum Throughput ($\text{Throughput}_{\text{max}}$) for the given channel conditions along with the difference of Goodput from Throughput, attempting to locate the Goodput as a point within a line that starts from 0 and ends at $\text{Throughput}_{\text{max}}$. The metric EEE takes values from 0 to 1, attempting to capture both distances.

$$\text{EEE} = a\frac{\text{Throughput} - \text{Goodput}}{\text{Throughput}_{\text{max}}} + b\frac{\text{Throughput}_{\text{max}} - \text{Throughput}}{\text{Throughput}_{\text{max}}}. \tag{3}$$

The first term of the EEE metric represents the overhead of network communication, normalised by resource availability (i.e. $\text{Throughput}_{\text{max}}$). Protocol overhead has a different impact on energy consumption depending each time on the particular device. Consequently, for every network card a different a value should be assigned. More precisely, the coefficient a is a function of the network card transmission power (P_{tran}) value and can be estimated experimentally.

The second term of the EEE metric captures the amount of available resources that have been exploited. When the available resources are exhausted, Throughput reaches $\text{Throughput}_{\text{max}}$. This term reflects energy consumption due to unexploited resources (e.g. time passes without any transmission). The b coefficient is a function of the network card idle power (P_{idle}) value. This term is bounded by the maximum energy consumption due to protocol inactivity. Consequently, the b coefficient is a function of idle power (P_{idle}) and not sleep power (P_{sleep}).

To summarise, the a and b parameters follow the behaviour of a specific network device. In many cases, a sophisticated energy efficient protocol consumes more energy than it is designed to, due to lack of sophistication of the network device. However, the energy potential of a network protocol is not device dependant.

The *ideal EEE*, is the EEE produced by an ideal device. We assume that an ideal network device is energy efficient and sophisticated in the sense that its states correspond always to the states of the transport protocol (i.e. when the protocol suspend transmission the device remains on an idle state). Therefore, this device allows the transport protocol to operate on its maximum energy efficiency. According our assumption, such a network card has a $P_{\text{idle}}/P_{\text{tran}}$ ratio of 0.3[2] and consumes the 30% of its energy in the idle state. Note that we did not find any network card with lower ratio. For example, according to [14], the Wavelan 2.4 GHz wireless network card has a $P_{\text{idle}}/P_{\text{tran}}$ ratio of 0.78. In this context, the EEE metric normalised with the parameters $a = 1$ and $b = 0.3$ behaves almost ideally.

When Goodput approaches Throughput, which approaches 0, the extra energy expenditure is only due to time waiting (probably in an idle state). We assume that the extra energy expenditure at this stage is 0.3 (the first term is 0). Instead, when $\text{Goodput} = \text{Throughput} = \text{Throughput}_{\text{max}}$ the extra expenditure is 0, since all the expended energy has been invested into efficient transmissions. Also, when $\text{Throughput}_{\text{max}} = 100$, $\text{Throughput} = 99$, $\text{Goodput} = 1$, the extra energy expenditure due to unsuccessful retransmission grows to an almost maximum value (0.993).

In the same context, Fairness is derived from the formula given in Ref. [3] and defined as:

$$F(x) = \frac{\sum_0^{n-1} (\mathrm{Throughput}_i)^2}{n\sum_0^{n-1} \left(\mathrm{Throughput}_i^2\right)}, \tag{4}$$

where $\mathrm{Throughput}_i$ is the Throughput of the ith flow and n the flow number.

Fairness captures overall multiplexing capabilities, but does not indicate clearly whether flows exist that expend significant energy for zero return. Therefore, we complement this metric with the RI defined as:

$$\text{Risk Index} = \frac{\text{Number of unfavored flows}}{\text{Total number of flows}}. \tag{5}$$

We regard as unfavored flows, the flows that have less Goodput than a specific threshold. In our case, the threshold is the 50% of the average Goodput.

EP can be defined as:

$$\begin{aligned} \mathrm{EP} &= 1 - \mathrm{EEE}_{\mathrm{ideal}} \\ &= 1 - \frac{\mathrm{Throughput} - \mathrm{Goodput}}{\mathrm{Throughput}_{\max}} + 0.3\,\frac{\mathrm{Throughput}_{\max} - \mathrm{Throughput}}{\mathrm{Throughput}_{\max}}. \end{aligned} \tag{6}$$

An ideal energy efficient protocol should have *EP* with value 1, which means zero extra energy expenditure.

For the sake of our analysis, and in particular, in order to be able to classify the cause of energy loss we specifically introduce the UAR index, defined as:

$$\mathrm{UAR} = 1 - \left[k\frac{\mathrm{Throughput}}{\mathrm{Throughput}_{\max}} + l\frac{\mathrm{Goodput}}{\mathrm{Throughput}}\right], \tag{7}$$

where, typically, $k = 0.5$ and $l = 0.5$ (the k, l parameters may be adjusted according to a specific hardware). The unexploited available resources (UAR) index ranges also from 0 to 1, expressing a negative performance aspect.

UAR, [6] captures how well did the protocol exploit the windows of opportunities for successful transmissions. More precisely, holding transmission when conditions call for transmission, will perhaps result in minor energy expenditure but have a great cost on protocol Goodput. Reasonably, the case of Goodput = Throughput = 0 should not give us, at this point, a minor (as with the EEE metric) but a major penalty.

UAR metric captures the behaviour of the protocol in terms of available resources exploitation. A smooth protocol, which has a small α value, cannot exploit available bandwidth very fast. So, it has a high UAR value in the beginning. After some time, the protocol (due to the increased β value) is more aggressive. Consequently, the protocol may exploit available bandwidth efficiently further on.

The choice of metrics is very important for the experimental analysis. Each metric captures a different view of the protocol behaviour. Additionally, each application type calls for specific metrics. Table 1 summarises the metrics we used to highlight the different aspects of system performance.

Table 1. Metrics for evaluating energy performance.

Metric	Description
Goodput	Captures protocol performance
Extra energy expenditure	Captures extra energy expended due to protocol operation
UAR	Captures how well the protocol exploits the windows of opportunities for successful transmission
Fairness	Captures the multiplexing capabilities of the system
RI	Indicates whether flows exist that expend significant energy for minor gains
EP	Indicates the energy potential of a protocol

4. Experimental methodology

4.1 Evaluation plan

We developed a real testbed in order to perform measurements. Our testbed consists of a laptop, a desktop PC and a switch. We used the advanced configuration performances interface (ACPI) to sample current voltage level, current drawn and available energy (in mAh) from the laptop battery. ACPI is integrated in the Linux kernel and maps to the proc filesystem. ACPI takes measurements directly from the battery when an application accesses the corresponding file of the proc filesystem (*/proc/acpi/battery/BAT1/state*). Authors in Ref. [7] use similar methodology to measure energy consumption of 'basic' application-level tasks, such as processing, input/output (disk, display, etc.) and communication (transmission and reception over the network).

We used an Acer Aspire 1692WLMI with Debian Linux OS, equipped with a Sanyo 65W Li-Ion battery, an Intel PRO/Wireless 2200BG 802.11b/g network card and a Broadcom BCM5700 network card for wired network.

We developed a tool for analysing protocol performance which is focused on energy consumption. Our tool is based on almost TCP over UDP (*atou*; Ref. [4]), an application-level implementation of TCP. We integrated our protocols and performance metrics into *atou* and evaluated the impact of different transport mechanisms on the energy consumption. Every experiment started with a full battery. We repeated our experiments several times in order to have statistically accurate results. Each experiment lasted 600 s, a time-period deemed appropriate to allow all protocols to demonstrate their potential. We used standard New-Reno TCP(1, 0.5), an extreme aggressive TCP(1.2, 1) with a small fixed timeout (50 ms) and a conservative TCP (0.3, 0.2) with a large fixed timeout (1 s), in order to explore the limits of the energy consumption due to the network communication and to adjust our metrics. We used the adjusted metrics to evaluate three classes of TCP(α,β) protocols: (i) Standard New Reno TCP(1, 1/2); (ii) Responsive TCP(α,β), with relatively low β value and high α value; and (iii) Smooth TCP(α,β), with relatively high β value and low α value. We used the same testbed for the two-node scenario.

For a more extensive experimental analysis, we complemented our results by using ns-2 [9] based simulations. We used the same protocols and performance metrics. The network topology used is the typical single-bottleneck *dumbbell*, as shown in Figure 1. The bw_1 link is 10 Mbps, the bw_2 link is 10 Mbps and the bw_3 is 1 Mbps. We used equal number of source and sink nodes. We simulated a heterogeneous (wired and wireless) network with ns-2 error models, which were inserted into the access links at the sink nodes. The Bernoulli model was used to simulate packet-level errors with configurable packet error rate (PER). The simulation time was fixed at 120 s. Due to the deterministic nature of the experiments, statistical validity is not an issue.

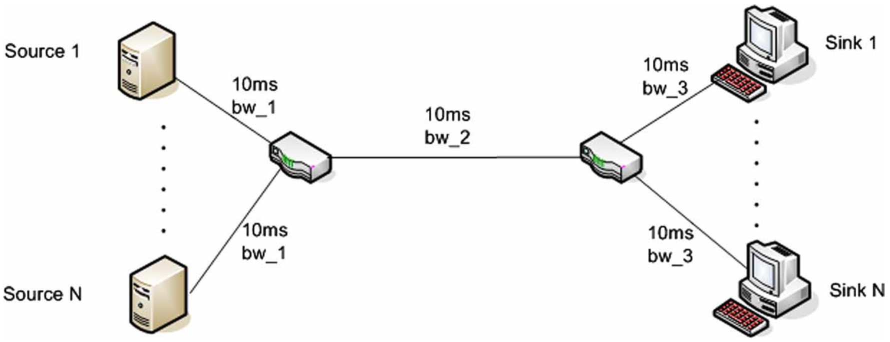

Figure 1. Dumbbell topology.

5. Results and discussion

5.1 Energy efficiency results and adjustments of metrics

In Figure 2(a), we observe the energy that three different transport mechanisms expend. The Idle curve depicts the energy consumption of our laptop battery when no communication takes place. When the TCP connection is on an Idle state (i.e. it does not transmit or receive any packet), the energy consumption slightly increases (Idle TCP curve). The actual communication-related energy consumption of a mechanism is therefore represented by the area between the corresponding energy-consumption curveand the Idle TCP one.

We assume that the aggressiveness of the transport mechanism ranges from the Idle TCP (which is zero) to the aggressive TCP and we adjust the EEE metric accordingly. We also assume that the Throughput of the aggressive TCP approaches the maximum Throughput that can be achieved under the specific network conditions. So, in the case of the aggressive TCP, the value of EEE should be close to 1 and the value of UAR approaches 0. In contrast, for the extremely conservative TCP, the UAR index should approach 1 and the EEE should approach the value $1.34/1.86 = 0.72$, where 1.34 is the average Idle TCP's power and 1.86 is the maximum power in the Figure 2(a).

Based on the Equation 3 and on the results depicted from Figures 2(a) and 3(a), we get:

$$a = \frac{\text{Throughput}_{\text{max}}}{\text{Throughput}_{\text{max}} - \text{Goodput}_{\text{aggressive}}} = 5, \tag{8}$$

$$b = 0.72. \tag{9}$$

In Figure 2(b), we show the impact of the different transport mechanisms on the available energy of the system. In the case of aggressive TCP the battery is drained faster. The conservative TCP is more energy efficient than TCP NewReno. The aggressive TCP consumes 4 mAh more energy than the conservative TCP and the NewReno 3 mAh.

The effort/gain dynamics of the system can be observed by Figure 3(a). The conservative TCP has less overhead, less Throughput but more Goodput than NewReno. Although, it expends less effort, it achieves more gains. Consequently, NewReno expends more effort in this specific scenario. Similarly, the aggressive protocol expends significant effort (26% more) for only 8% gain.

In Figure 3(b), we plot the behaviour of the three protocols in terms of EEE and UAR. The EEE_1 curve represents the ideal EEE, while the EEE_2 represents the EEE normalised

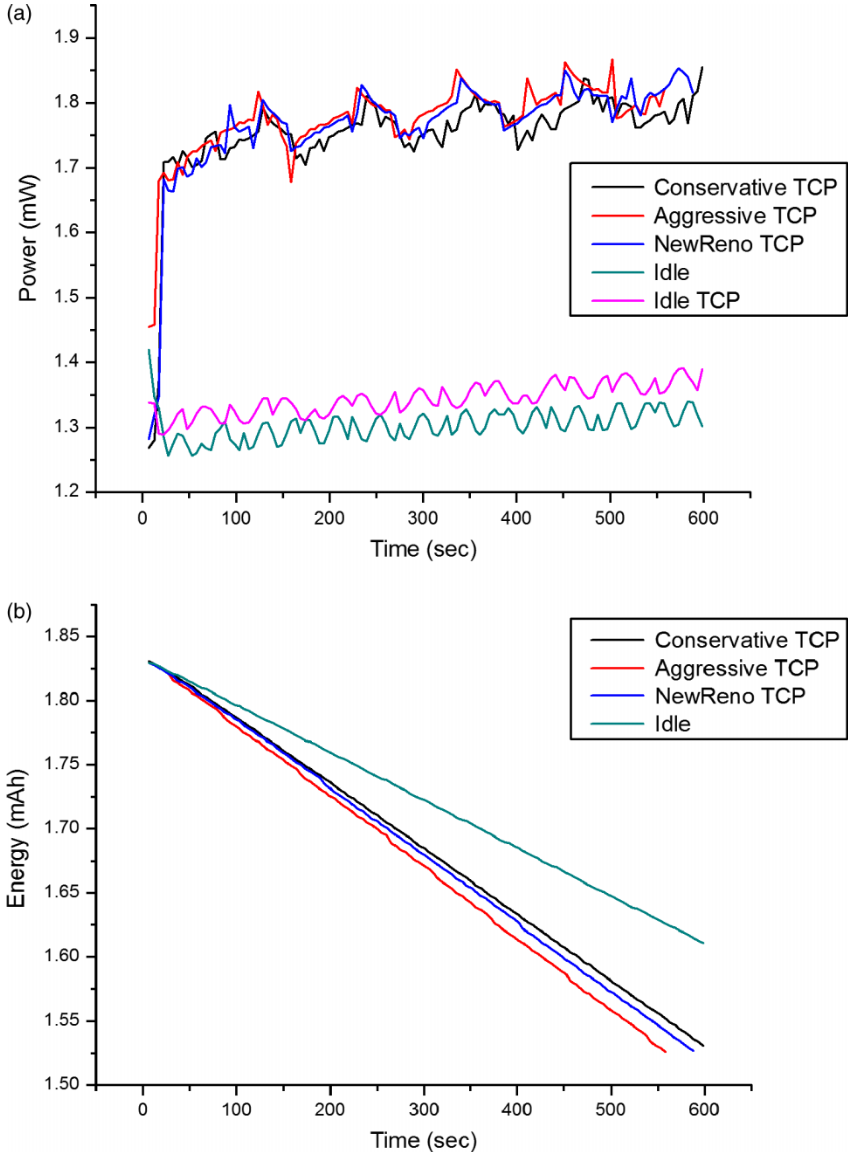

Figure 2. Energy-wise behaviour of different transport mechanisms. (a) Energy consumption of different transport mechanisms and (b) Available energy of the system.

to the particular network device. We can observe that the aggressive protocol consumes more energy and instead the conservative protocol is the most energy efficient. We can also claim, based on the same figure, that the space for improvement is significant for all protocols.

The three protocols transmit data for about 600 s. The conservative TCP transmits 6.3 GB with 174.2 MB overhead. The aggressive TCP transmits 8.4 GB with 1.5 GB overhead and the NewReno TCP 6.1 GB with 543 MB. The system consumes about 300, 304 and 303 mAh energy, respectively.

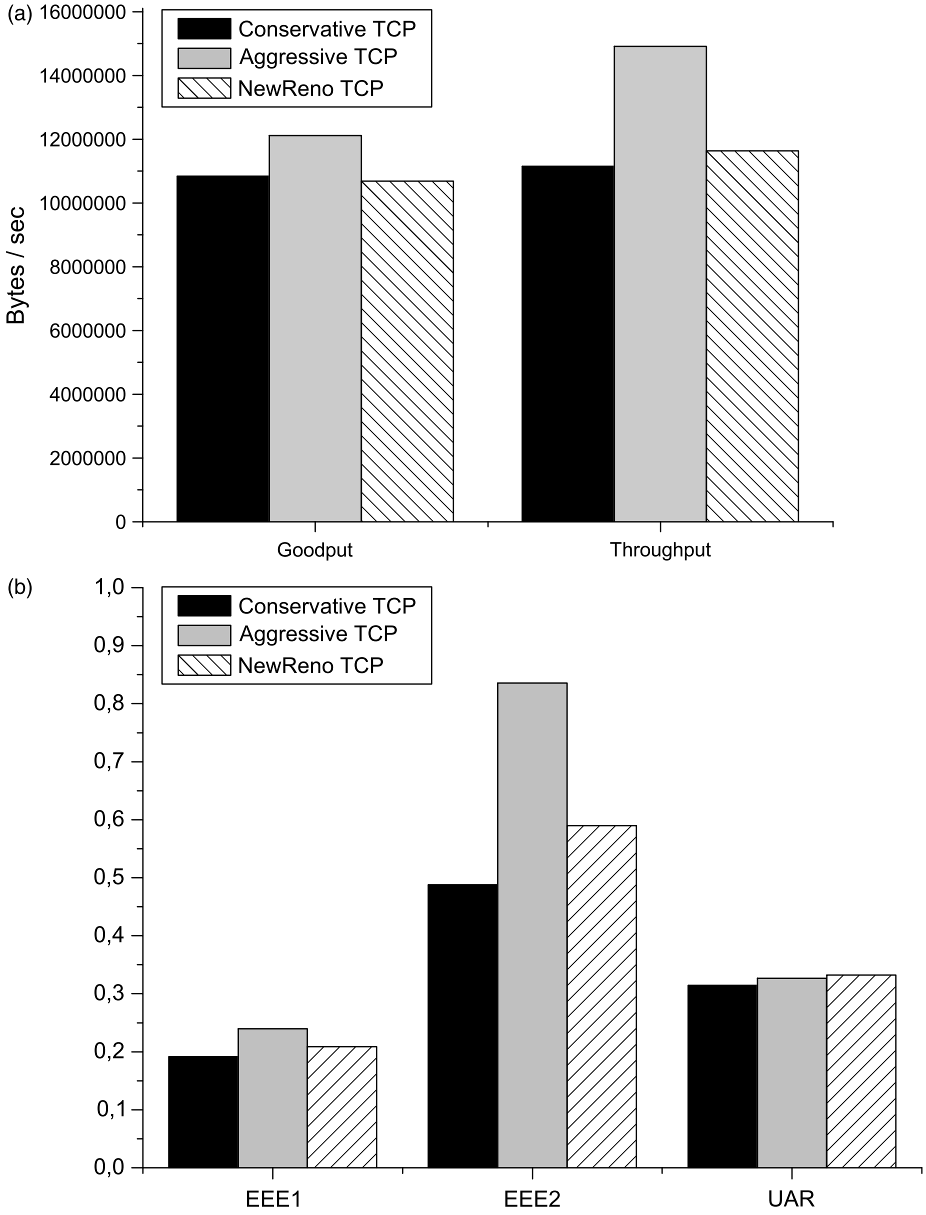

Figure 3. Behaviour of different transport mechanisms. (a) Performance of different transport mechanisms and (b) EEE and UAR of different transport mechanisms.

In contrast to the conservative version of the protocol, the aggressive version expends extra effort for 1.4 GB and consumes 4 mAh more energy in order to transmit 2.1 GB of useful data. However, the conservative version would have required an extra minute of communication in order to transmit the same amount of useful data (2.1 GB); the specific parameters of our experiments, would have caused more energy consumption than the 304 mAh of the aggressive version. However, we note that this conclusion may have been reverse had the network card idle state consumption been different (i.e. less power-demanding).

The NewReno TCP appears less energy efficient and is outperformed by the conservative TCP in terms of Goodput. The additional effort expended by NewReno is not invested in performance gains. This result is quite interesting: 302 MB less effort, which also corresponds to 3 mAh less energy consumption transmits 66 MB more useful data.

5.2 Evaluation of different transport mechanisms using testbed

We evaluate three different versions of TCP: NewReno TCP, Responsive TCP and Smooth TCP. The Responsive TCP is the TCP(1.24, 0.25) and the Smooth TCP is the TCP(0.31, 0.875). We repeat the experiment 10 times in order to investigate the statistical accuracy of the results. In the following Figures 4(a) and 5(d), we plot the average values for the 10 experiments. We did not observe significant deviation between the 10 experiments. For example, in the case of Fairness, the maximum deviation was 0.08, the minimum deviation was 0 and the average was 0.00899 (1.18%).

According to Figure 4(a) and (b), the aggressive behaviour of NewReno TCP is not translated into increased Goodput. Compared with the Responsive TCP, Smooth TCP expends slightly more effort (Figure 4(a)) for a very significant return in Goodput (Figure 4(b)). This behaviour is also captured by the UAR curve (Figure 5(c)). However, this extra effort is not distributed uniformly among participants (Figure 4(c)).

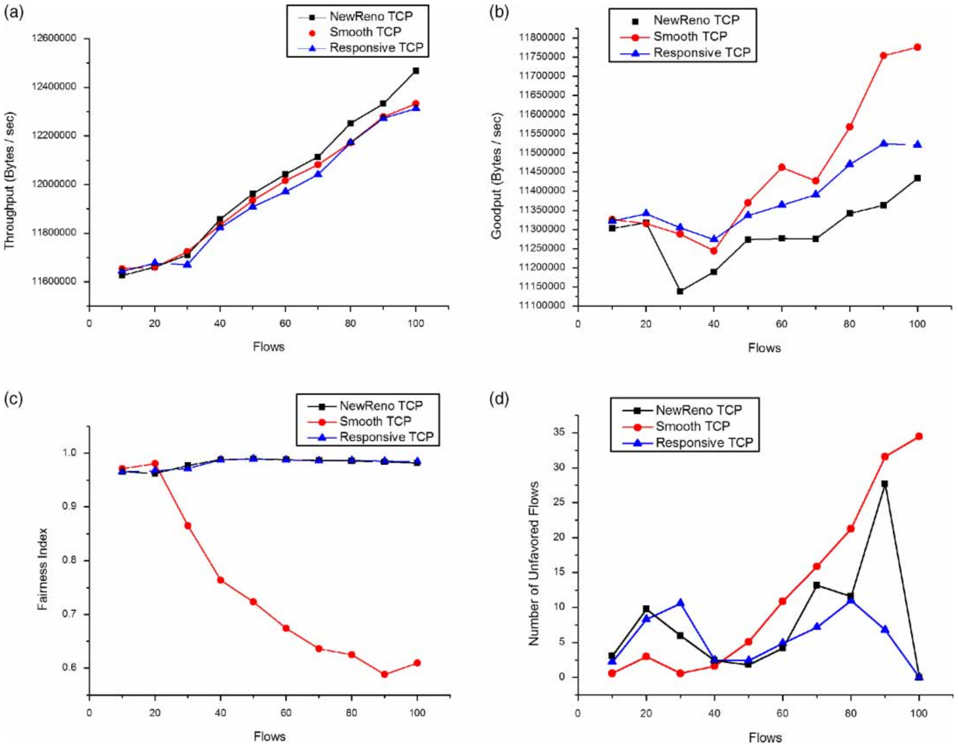

Figure 4. Behaviour of different transport mechanisms. (a) Throughput of different transport mechanisms; (b) Goodput of different transport mechanisms; (c) Fairness of different transport mechanisms and (d) Number of unfavoured flows of different transport mechanisms.

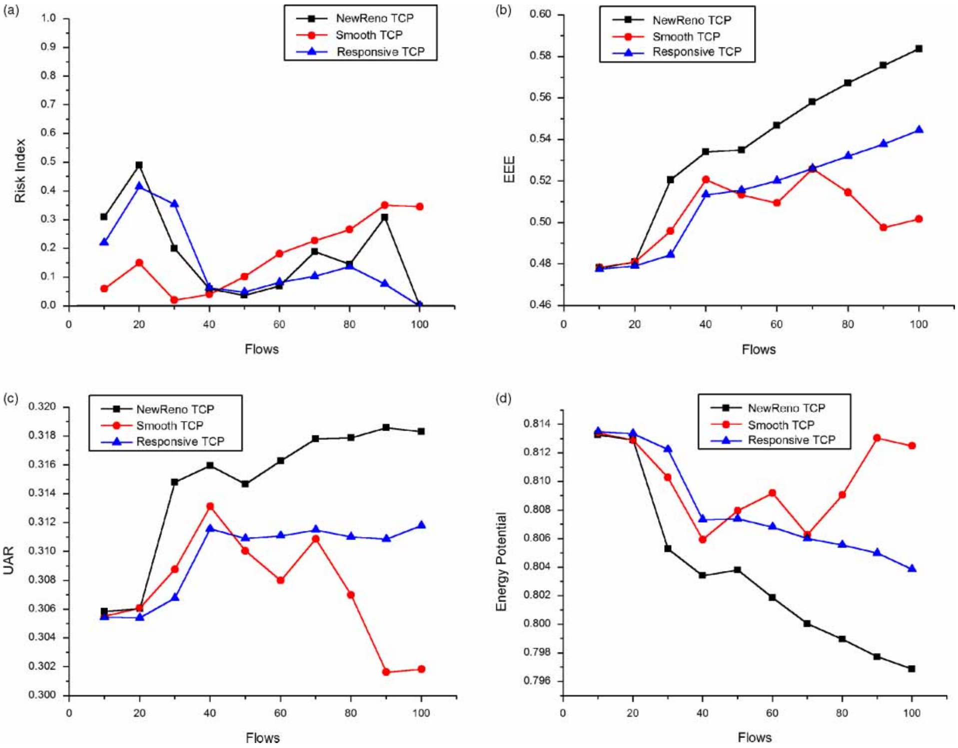

Figure 5. Behaviour of different transport mechanisms. (a) RI of different transport mechanisms; (b) EEE of different transport mechanisms; (c) UAR of different transport mechanisms and (d) EP of different transport mechanisms.

In Figure 4(d), we plot the amount of flows that receive bad service due to the unfair system behaviour. We defined as bad service the situation where a flow does not achieve at least the 50% of the average Goodput. While, NewReno and Responsive TCP exhibit similar behaviour in terms of Fairness, the Smooth TCP is not fair (Figure 4(c),(d)).

According to the RI (Figure 5(a)), the Smooth TCP appears unfair indeed. It causes several flows to receive bad service, which in turn causes great uncertainty to users of such system, especially when contention is high. There, the probability to expend significant energy for minor return is higher, even if the system is in general, more energy efficient.

Furthermore, Smooth TCP appears more energy efficient (Figure 5(b)). The situation uncovers a very interesting tradeoff. At least occasionally, in order to achieve better energy efficiency system-wise, we may let the RI grow. In Figure 5(b), we show the ideal EEE curve. In Figure 5(d), we plot the behaviour of the three protocols in terms of EP. We can see that, independently of the network device, the Smooth TCP has the best EP in this particular case. Additionally, the NewReno TCP is outperformed by Responsive TCP in terms of energy efficiency (Figure 5(d)).

5.3 Evaluation of different transport mechanisms using simulations

We evaluated the same protocols with ns-2 based simulations. We used the same dumbbell topology and different levels of heterogeneity.

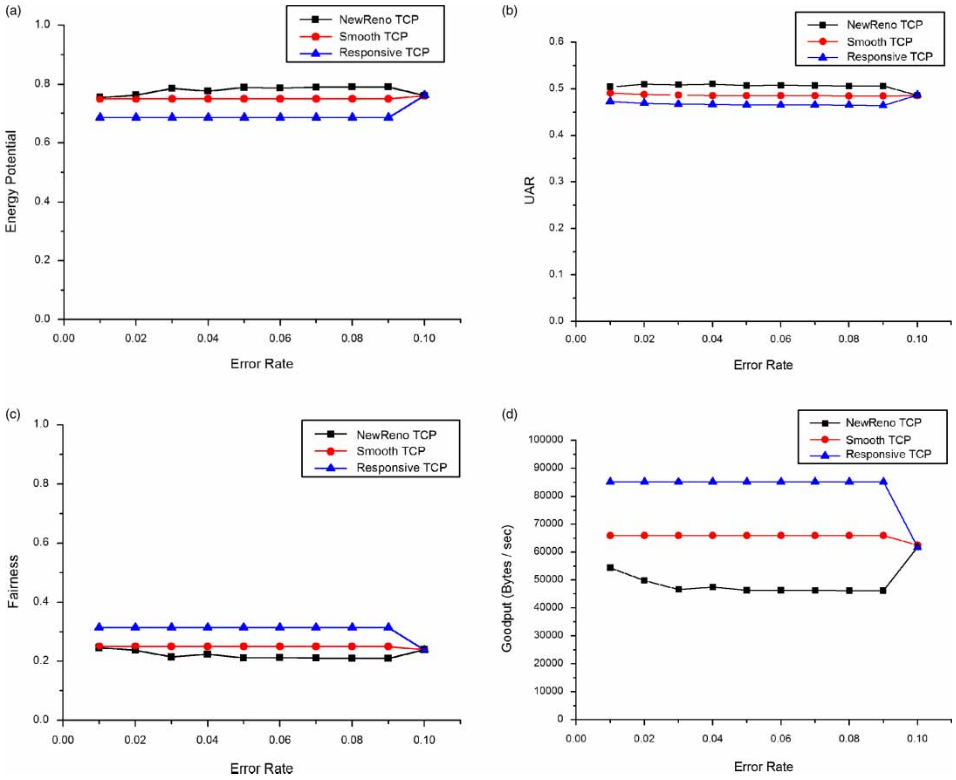

Figure 6. Low error-rate favors responsive protocols. (a) EP and low error-rate; (b) UAR and low error-rate; (c) Fairness and low error-rate and (d) Goodput and low error-rate.

5.3.1 *Low error-rate favours responsive protocols*

In the first scenario we simulated a heterogeneous environment with random transient errors increasing from 0.01 to 0.1 PER. We used 30 flows and a 10 Mbps bottleneck, a relatively low-contention environment. The responsive protocol outperforms the smooth one in terms of energy efficiency (Figure 6(a)) and performance in terms of Goodput (Figure 6(d)) because it exploits the available resources better (Figure 6(b)). In this case, the responsive protocol deals with the transient errors sooner due to the setting of parameter α, without any negative impact on the system's fairness (Figure 6(c)).

5.3.2 *A macroscopic view of the effort/gain dynamics*

In this scenario, we used handoffs with duration 0.2 sec in a 10 Mbps bottleneck. We measured performance in terms of Goodput, ranging the number of flows from 10 to 100. We can observe that, better resource (Figure 7(d)) and energy exploitation (Figure 7(a)) may have a positive impact on protocol Goodput, although, the reverse is also possible. See, for example the contrasting outcome with less and more effort, in Figures 6(a),(b),(d) and 7(a),(b),(d), respectively. Although, smooth TCP appears fair (Figure 7(c)), it is less energy efficient (Figure 7(a)) due to worse resources exploitation (Figure 7(d), (b)). A sophisticated protocol should have gains in terms of energy

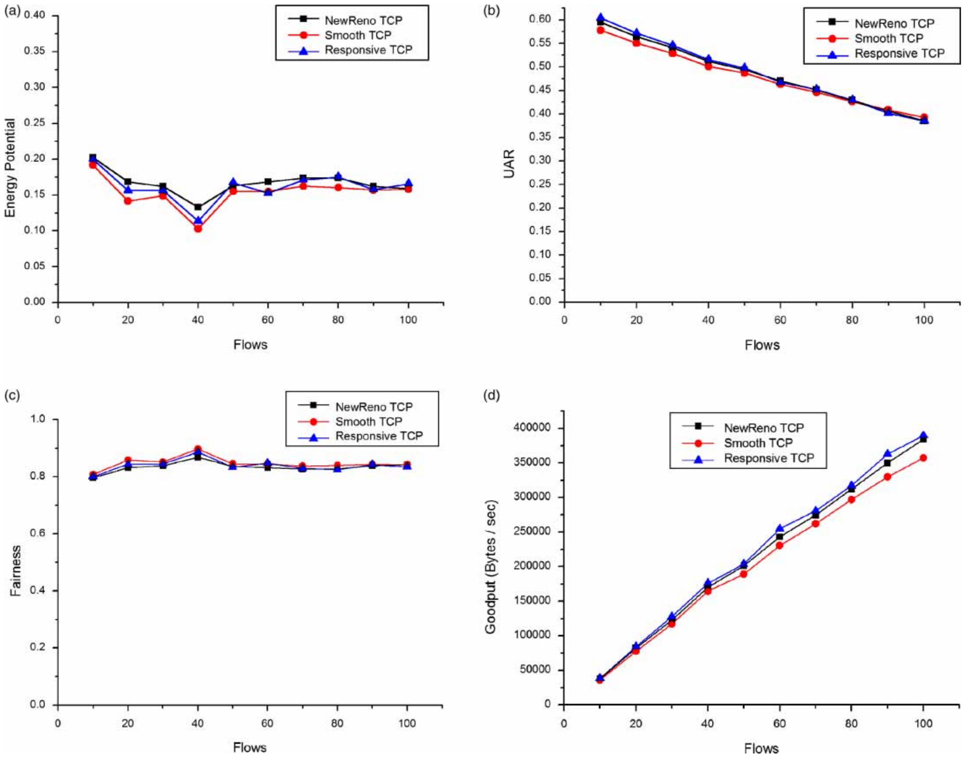

Figure 7. A macroscopic view of the effort/gain dynamics. (a) EP and handoffs; (b) UAR and handoffs; (c) Fairness and handoffs and (d) Goodput and handoffs.

consumption and performance but without being unfair. Otherwise, some flows may drain their resources for minor data transmission.

5.3.3 Observations with contention decrease

The next scenario presented here intends to provide a framework for characterising protocol behaviour when bandwidth becomes available rapidly in heterogeneous networks. We measure EP, Figure 8(a); UAR Index, Figure 8(b); Fairness, Figure 8(c) and Goodput, Figure 8(d) for a range of flows from 10 to 20. We used a 0.2 PER. All flows are entering in the system within the first 2 s. For the rest 118 s, we have a graduated contention decrease, starting from 10 flows and repeating the experiment for 11 to 20 flows. We reduce the number of flows to half every Decrease_Step seconds, where Decrease_Step, is the step needed, in order for the last flow to exit at the 120th second.

The small value of parameter α of Smooth TCP leads to slow resource exploitation (Figure 8(b)) without any gains in terms of energy efficiency (Figure 8(a)). On the other hand, Responsive TCP consumes less energy (Figure 8(a)) but exploits resources (Figure 8(d) in an unfair manner (Figure 8(c)).

5.3.4 Error-rate increase cancels responsive TCP's advantages

In the following scenario, we used 30 flows, a 10 Mbps bottleneck and a variable error-rate from 0.01 to 0.4 PER. During small error rates the responsive protocol has better return for

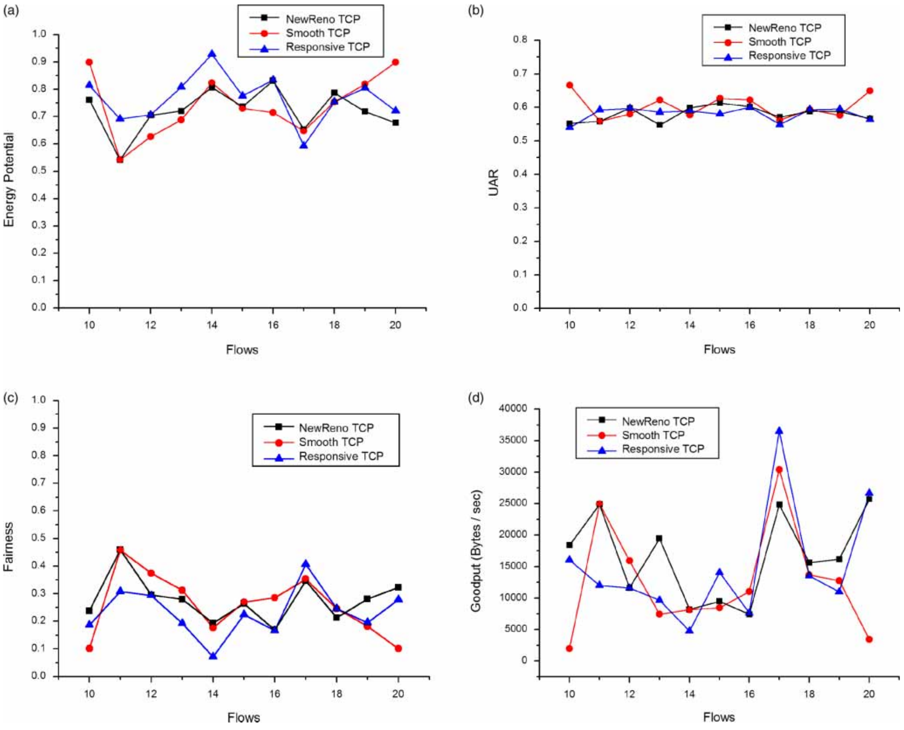

Figure 8. Observations with contention decrease. (a) EP and contention decrease; (b) UAR and contention decrease; (c) Fairness and contention decrease and (d) Goodput and contention decrease.

its effort, however, when error-rate exceeds 0.1, these advantages are cancelled (see Figure 9(a), (b), (c)). In Figure 9(a)–(d), we summarise the difference in EP, UAR Index, Goodput and Fairness.

We can see that the responsive protocol is favoured at the beginning. After a certain point, which is relevant to the specifics of the experiment (which in our case is 0.1), the smooth protocol may even become more efficient (in Goodput) and fair, while it expends less extra energy.

6. Conclusions

We explored the energy-saving potential of different transport protocols using a real testbed. We introduced two new metrics, the EP and RI. EP is a device-independent metric, which captures the energy-saving potential of a protocol. RI refers to a system's behaviour and captures the potential risk for a flow to expend its energy for minor Goodput, due to the multiplexing limitations.

We confirm experimentally that, in general, smoothness and responsiveness constitute a tradeoff; however, we show that this tradeoff does not correspond to a conservative/aggressive behaviour. Energy-wise, existing protocol tactics cannot always be justified; our results suggest that an adaptive congestion control algorithm is needed to integrate the dynamics of heterogeneous networks into protocol behaviour.

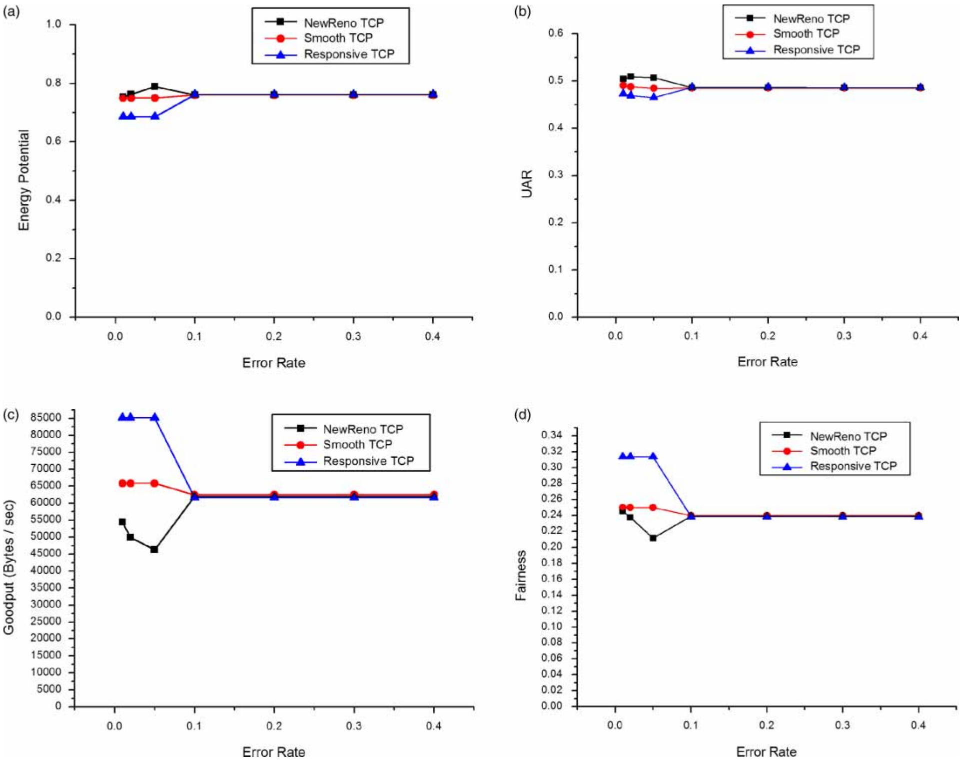

Figure 9. Error-rate increase cancels responsive TCP's advantages. (a) EP and error-rate; (b) UAR and error-rate; (c) Goodput and error-rate and (d) Fairness and error-rate.

Notes

1. That is, an aggressive transmission may result in long periods of suspension.
2. This assumption is subject of further work and may be explored theoretically.

References

[1] S. Agrawal and S. Singh, *An experimental study of TCPs energy consumption over a wireless link*, in *Proceedings of the 4th European Personal Mobile Communications Conference*, 2001.

[2] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, *A power saving architecture for Web access from mobile computers*, in *Proceedings of the 2nd International IFIP Networking Conference (Networking 2002)*, 2345, 2002, pp. 240–251.

[3] D. Chiu and R. Jain, *Analysis of the increase/decrease algorithms for congestion avoidance in computer networks*, J. Comput. Netw. ISDN 17(1) (1989), pp. 1–14.

[4] T. Dunigan, *Almost TCP over UDP (ATOU)*. Available at http://www.csm.ornl.gov/dunigan/netperf/atou.html.

[5] C. Jones, K. Sivalingam, P. Agrawal, and J. Chen, *A survey of energy efficient network protocols for wireless networks*, Wireless Netw. 7(4) (2001), pp. 343–358.

[6] L. Mamatas and V. Tsaoussidis, *Protocol behaviour: More effort, more gains?*, in *Proceedings of PIMRC 2004*, Barcelona, Spain, 2004.

[7] C. B. Margi, K. Obraczka, and R. Manduchi, *Characterizing system level energy consumption in mobile computing platforms*, in *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing*, 2, 2005, pp. 1142–1147.

[8] H. Singh and S. Singh, *Energy consumption of TCP Reno, Newreno, and SACK in multi-hop networks*, In *Proceedings of ACM SIGMETRICS 2002*, Marina del Rey, California, USA, 2002.

[9] The Network Simulator – NS-2 *Web Page: http://www.isi.edu/nsnam/ns*.

[10] V. Tsaoussidis, H. Badr, X. Ge, and K. Pentikousis, *Energy/Throughput Tradeoffs of TCP Error Control Strategies*, in *Proceedings of the 5th IEEE Symposium on Computers and Communications (ISCC)*, Antibes-Juan les Pins, France, July, 2000.

[11] V. Tsaoussidis and C. Zhang, *The dynamics of responsiveness and smoothness in heterogeneous networks*, IEEE J. Selected Areas Commun. 23(6) (2005), pp. 1178–1189.

[12] B. Wang and S. Singh, *Computational energy cost of TCP*, in *Proceedings of the INFOCOM 2004*, Hong Kong, 2004.

[13] Y. Yang and S. Lam, *General AIMD congestion control*, in *Proceedings of the IEEE International Conference on Network Protocols*. Osaka, Japan, November, 2000.

[14] W. Yong, S. Chandra, and S. Bhandarkar, *A statistical prediction-based scheme for energy-aware multimedia data streaming*, in *Wireless Communications and Networking Conference (IEEE WCNC 2004)*, 2004, pp. 2053–2057.

[15] C. Zhang and V. Tsaoussidis, *The interrelation of TCP responsiveness and smoothness*, in *Proceedings of the 7th IEEE ISCC*, July, 2002.