



HAL
open science

TCP over low-power and lossy networks: tuning the segment size to minimize energy consumption

Ahmed Ayadi, Patrick Maillé, David Ros Sanchez

► **To cite this version:**

Ahmed Ayadi, Patrick Maillé, David Ros Sanchez. TCP over low-power and lossy networks: tuning the segment size to minimize energy consumption. 2010, pp.20. hal-00724802

HAL Id: hal-00724802

<https://hal.science/hal-00724802v1>

Submitted on 22 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Collection des rapports de recherche
de Télécom Bretagne

RR-2010004-RSM



**TCP over low-power and lossy networks:
tuning the segment size to minimize
energy consumption**

*Ahmed Ayadi (Télécom Bretagne - Networks, Security and Multimedia department (RSM))
Patrick Maillé (Télécom Bretagne - Networks, Security and Multimedia department (RSM))
David Ros (Télécom Bretagne - Networks, Security and Multimedia department (RSM))*



Abstract

Low-power and Lossy Networks (LLNs), like wireless networks based upon the IEEE 802.15.4 standard, have strong energy constraints, and are moreover subject to frequent transmission errors, not only due to congestion but also to collisions and to radio channel conditions.

This paper introduces an analytical model to compute the total energy consumption in an LLN due to the TCP protocol. The model allows us to highlight some tradeoffs as regards the choice of the TCP maximum segment size, of the Forward Error Correction (FEC) redundancy ratio, and of the number of link-layer retransmissions, in order to minimize the total energy consumption.

Keywords

Multi-Hop Wireless Networks, Low-Power and Lossy Networks, Energy Efficiency, TCP.

Contents

1	Introduction	5
2	Related work	6
3	TCP energy consumption model	7
3.1	Link layer: one-hop model	7
3.1.1	Link layer mechanisms	7
3.1.2	Performance of one-hop transmissions	9
3.2	Multi-hop model	11
3.3	TCP performance	12
4	Results and discussion	14
4.1	Energy model	14
4.2	Model assessment	14
4.3	FEC redundancy ratio and energy consumption	16
4.4	Selecting the TCP MSS to minimize energy consumption	16
5	Conclusions and Future Work	18
6	Acknowledgements	19

List of Figures

1	Failure and success scenarios for one link-layer transmission attempt. . .	9
2	Failure and success scenarios in a multi-hop transmission.	11
3	Energy consumption with long or short TCP segments, as a function of the BER B	15
4	Number of collisions in a multi-hop scenario.	16
5	Energy consumption with short or long TCP segments, as a function of the number of link layer attempts r (with $B = 5 \times 10^{-4}$).	17
6	Consumed energy using short or long TCP segment, as a function of the redundancy ratio α ($B = 3 \times 10^{-4}, h = 5$).	17
7	Long (MSS=512 bytes) versus short (MSS=64 bytes) in a multi-hop TCP transmission: prefer the short MSS above the curves, the long one below.	18
8	Long (MSS=512 bytes) versus short (MSS=64 bytes) in a multi-hop TCP transmission: prefer the short MSS above the curves, the long one below.	19

List of Tables

1	Notations used in this paper; capital italics letters correspond to probabilities, bold letters to (expected) numbers of bits.	8
2	Energy parameters.	14
3	Default simulation parameters	15

1 Introduction

IP-based Low-power and Lossy Networks (LLNs) are composed of (possibly many) nodes which are subject to strong constraints on power and computing capabilities. Besides, link-layer technologies like IEEE 802.15.4 [10], often used for building wireless LLNs, impose very low limits on link frame sizes.

Many usual applications for LLNs (e.g., wireless sensor networks) do not always require full reliability. However, there exist several use-cases for LLNs where the reliability of data delivery can be critical, like over-the-air software updates [3, 14, 19]. In such cases, the Transmission Control Protocol (TCP) [15] is a common choice for ensuring an end-to-end reliable transport in IP-based LLNs.

TCP is the most used transport protocol in IP networks. It provides a reliable transfer service for all kinds of applications. TCP can manage data that is damaged, duplicated, lost, or delivered out of order. This is achieved by assigning a sequence number to each octet transmitted and requiring a positive acknowledgement (ACK) from the recipient. If the TCP ACK segment is not received within the timeout interval, the data segment is retransmitted.

This paper focuses on the energy cost of reliability when TCP is used in multi-hop wireless LLNs. In what follows, we will illustrate some main issues by considering the case of TCP in IPv6-enabled LLNs based on the 6LoWPAN protocols [11].

Many parameters affect the energy consumption of a reliable transport protocol, like channel conditions (e.g., wireless losses, collisions), the level of link-layer reliability, the number of links/hops, or the maximum size of link-layer frames (e.g., 127 bytes for IEEE 802.15.4). The length of such frames may require the fragmentation of IPv6 datagrams before sending them. The IETF 6LoWPAN working group has defined a new protocol layer between the IPv6 and the medium access control (MAC) layers. The 6LoWPAN layer compresses the IPv6 header and fragments IPv6 packets into short MAC frames. So, if the size of a TCP segment exceeds the maximum allowed length, the 6LoWPAN layer fragments it to make it fit into link-layer frames. Therefore, sending short TCP segments does not require any fragmentation. However, the use of a small maximum segment size¹ (MSS) increases the number of TCP segments and, thus, the number of TCP acknowledgements and the corresponding MAC frames sent. Besides, with small MSS values the overhead due to TCP headers becomes larger; this, coupled with the small size of MAC frames, may result in a very low protocol efficiency.

This paper introduces a mathematical model aimed at predicting the energy consumed by the wireless nodes of an LLN in a bulk-data transfer scenario, with TCP used as a reliable transport layer. The model estimates TCP energy performance based on the bit error rate, the maximum number of retransmissions at the link layer, the number of hops between the sender and the receiver, the amount of FEC, and the TCP maximum segment size.

The proposed model allows us to study the tradeoffs involved in sending short versus long TCP segments. We assume that the energy consumed in a data transfer depends mainly on the number of bits sent. Thus, our analytical model estimates the number of bits sent by all nodes in the network, taking into account the cumulated cost of all

¹The TCP MSS option refers to the maximum amount of data a TCP segment can carry, i.e., excluding the TCP header and any options that might be present.

link-layer transmissions. Indeed, the number of MAC frames sent can be large with respect to the initial amount of data to transmit, because of link-layer retransmissions, the redundancy added for Forward Error Correction (FEC), and also due to end-to-end (i.e., transport layer) retransmissions when a TCP segment is lost before reaching the receiving node.

We apply the model to study the energy efficiency of TCP over an LLN using 6LoWPAN and 802.15.4 protocols, and study the effect of the TCP segment size, of the FEC redundancy ratio, and of the maximum link layer retransmission on the total energy consumption.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of the related work in the area of TCP performance in LLNs, and in multihop wireless networks in general. Section 3 presents the derivation of our analytical model. In Section 4, we apply the analytical results to derive the best TCP segment size strategy in terms of energy consumption. Finally, in Section 5, we give our conclusions and provide directions for future work.

2 Related work

There exist many studies of the performance of TCP in multi-hop wireless networks (see for example [8] and references therein), though many of them are concerned mostly with parameters like TCP throughput, and with protocol enhancements aimed at improving such performance metrics. Another amount of work focuses on the computational cost of TCP (e.g., [20]), or on the comparison of the energy efficiency for different TCP congestion control algorithms (like SACK, Tahoe, Reno and NewReno) over wireless networks (e.g., [9, 17, 21]). In [6], Fu et al. study in the impact of the MAC layer protocol on the TCP throughput over wireless multi-hop networks. The authors show that, in contrast to wired networks, the TCP throughput does not increase by increasing the window size, and that (under the conditions of the study) better performance is achieved with a window size equal to the number of hops divided by four.

In [12], Lilakiatsakum and Senevirane propose an energy-efficiency metric to compare the performance of different versions of TCP. The metric is the ratio between the amount of bits sent by all nodes and the size of the application data. In this work, we adopt a variant of that metric (the total amount of bits sent); we use a fixed value for the application data size in all our numerical and simulation scenarios.

Bansal et al. [1] propose an analytical model to compute the energy consumed for carrying a TCP flow over a multi-hop wireless network. The authors assume that all wireless links have the same packet error rate P and the same transmission power. They compute the energy spent by all nodes for sending a single TCP segment. The energy consumption is then a function of the packet error rate, the number of hops, and the maximum number of link-layer retransmissions. However, [1] does not take into account the cost of sending link-layer acknowledgements, nor the cost of transport-layer acknowledgements, which we add in the model introduced here. Besides, we explicitly consider the impact of lower-layer fragmentation and of error correction, as well as different per-link error rates on the energy-efficiency of TCP.

Barman et al. [2] and Gallucio et al. [7] present an analytical study of a TCP optimiza-

tion problem in an hybrid wired/wireless network where the last hop is a wireless link. The authors of both papers define an utility function which is the ratio of the throughput to the cost of a TCP connection. Our work completes these studies by introducing a multi-hop model for computing the energy consumption. Note that in this paper we are not interested in TCP throughput, because data rates are a secondary concern in LLNs; instead, we focus mainly on the energy costs of a TCP connection over multiple lossy links.

3 TCP energy consumption model

This section introduces a mathematical model to estimate the energy consumed by a TCP transmission in a wireless LLN. We assume that such energy mainly corresponds to data emission and reception, and thus directly depends on the number of bits sent by all nodes.

We therefore compute the expected total amount of bits sent for a successful end-to-end TCP data transmission, as a function of several network parameters, namely: the bit error rate, the link-layer maximum number of attempts, the FEC redundancy ratio, the number of hops between the source and receiver TCP hosts, and the TCP maximum segment size.

For the convenience of the reader, Table 1 lists most of the variables used in this paper.

3.1 Link layer: one-hop model

We first focus on modeling the link-layer energy consumption, considering a CSMA-CA network with error correction control techniques combining Automatic Repeat Request (ARQ) [5] and Forward Error Correction (FEC).

3.1.1 Link layer mechanisms

Standard ARQ uses the cyclic redundancy check (CRC) error-detecting code that is added to the data: The receiver uses the error-detecting code number to check the integrity of the received data. After receiving a correct frame, the receiver replies by an ACK. If the sender does not receive any ACK before a timeout²—because either the original message or the ACK is lost, or they contain errors—, the sender retransmits again the same message. If the receiver sees that the frame is damaged the receiver discards it and does not send an ACK. The ARQ algorithm continues until the sender receives an ACK or exceeds a predefined number of attempts r .

ARQ is the algorithm most frequently used by link-layer protocols to reduce the packet error rate. However, if the wireless network becomes very lossy, ARQ would increase the transmission delay between the source and the receiver. Abrupt increases of the end-to-end delay increase the round-trip time and may lead to a spurious TCP timeout. This can deteriorate the TCP performance.

²Remark that we do not consider time issues in this model, thus, only losses can lead to retransmissions.

Table 1: Notations used in this paper; capital *italics* letters correspond to probabilities, bold letters to (expected) numbers of bits.

Variable	Definition
h	Number of hops between source and destination
r	Maximum number of transmission attempts at the link layer
m	Number of fragments corresponding to a single TCP segment (due to link layer fragmentation)
α	FEC redundancy ratio
B	Bit error rate
P_{fail}	Probability of a failure in a link-layer transmission attempt of a data frame
P_{partial}	Probability of a link-layer data frame being correctly received and the corresponding acknowledgement being lost
P_{succ}	Probability of a successful link-layer transmission attempt (data+acknowledgement are correctly received)
F	Probability that a destination node does not receive a link layer data frame after r attempts
Q_s	Probability of an end-to-end packet transmission success
Q_f	Probability of an end-to-end packet transmission failure
D	Link-layer data frame size
A	Link-layer acknowledgement frame size
H_f	Expected number of bits sent after r attempts knowing that the (one-hop) transmission has failed
H_s	Expected number of bits sent within r attempts knowing that the (one-hop) transmission has succeeded
E_f	Expected number of bits sent for an end-to-end packet transmission knowing that it has failed
E_s	Expected number of bits sent for a successful end-to-end packet transmission knowing that it has succeeded
S	Average number of bits sent for successfully transmitting a TCP segment

An orthogonal approach consists in applying Forward Error Correction (FEC). FEC is a good solution for decreasing the packet error rate. The main idea of FEC is to add redundancy to the original frame, to allow the destination node to detect and correct some bit errors. In our case, if the size of a network datagram is greater than the maximum transmission unit (MTU) of the link layer, the datagram is divided into fragments of length \mathbf{K} bits, and the FEC algorithm adds $(\alpha \times \mathbf{K})$ redundancy bits to form a frame of length \mathbf{D} . The ratio $\alpha = \frac{\mathbf{D}-\mathbf{K}}{\mathbf{K}}$ between the amount of redundancy added by FEC and the original frame length is called the *redundancy ratio*.

In what follows we will consider an error-correction method like the well-known Reed-Solomon [16] (RS) algorithm. By adding $\mathbf{D} - \mathbf{K}$ bits to the \mathbf{K} data, the RS algorithm can correct up to $(\mathbf{D} - \mathbf{K})/2$ bits.

3.1.2 Performance of one-hop transmissions

First, we consider here only a transmission between two immediate neighbors, without intermediate nodes. In CSMA-CA, two types of messages are used during data transmission: the data message (i.e., the message that contains the useful data), and the acknowledgement message sent by the receiving node. Figure 1 shows the three possible cases:

- a. *Failure*: A failure due to the loss of the data frame. The sender will retransmit the frame.
- b. *Partial failure*: The data frame is correctly received by the receiver, while the acknowledgement frame is lost³. Therefore, the sender will (needlessly) retransmit the data frame.
- c. *Success*: A successful transmission of both the data and the acknowledgement frames.

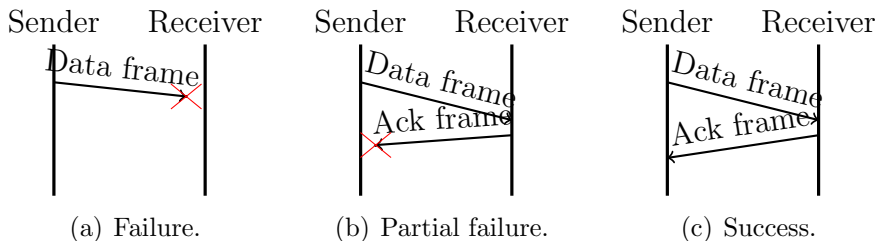


Figure 1: Failure and success scenarios for one link-layer transmission attempt.

The probability of a successful transmission of a link-layer frame depends on the bit error rate, the size of the message, and the redundancy ratio. Here we assume that all bits undergo independent random errors with a given bit error rate B , each bit being correctly received with a probability $1 - B$. Moreover, we assume that a frame transmission fails as soon as the number of bits erroneously received exceeds the number of correctable bits. The probability P_{fail} of having a *failure* (case (a) in Fig. 1) is then the error probability for a data frame, i.e.,

$$P_{\text{fail}} := 1 - \sum_{i=0}^c \binom{\mathbf{D}}{i} B^i (1 - B)^{\mathbf{D}-i}$$

where \mathbf{D} is the number of bits of a data frame, $c = \frac{\mathbf{D}-\mathbf{K}}{2} = \frac{\alpha\mathbf{K}}{2}$ is the number of correctable bits, and for two integers i and j , $\binom{j}{i}$ is the number of possibilities of choosing i elements among j , i.e., $\binom{j}{i} = \frac{j!}{i!(j-i)!}$.

Likewise, denoting by \mathbf{A} the size of an acknowledgement frame, the probability of a *partial failure* is the probability that the data frame is correctly received but the acknowledgement frame contains errors:

³In a multi-hop setting, the receiver would nonetheless relay the frame to the next hop.

$$P_{\text{partial}} := (1 - P_{\text{fail}})(1 - (1 - B)^{\mathbf{A}}),$$

whereas the probability of a *success* is

$$P_{\text{succ}} := (1 - P_{\text{fail}})(1 - B)^{\mathbf{A}}.$$

Note that we assume that acknowledgement frames are sent *without* adding redundancy. This is a reasonable assumption, given the current practices in most wireless technologies.

In order to estimate the energy consumption, we compute the expected number of bits sent at the link layer. Remark that in cases of success or partial failure, $\mathbf{D} + \mathbf{A}$ bits are sent, whereas only \mathbf{D} bits are sent in cases of failure. Recall however that the redundancy ratio determines the number of useful bits per frame, and thus the number of frames to send.

We now take into account the link-layer sending attempts, following the ARQ technique implemented by the MAC protocol. We denote by r the maximum number of attempts, after which the sender of a data frame considers the receiver is unreachable.

The *reception* failure probability of the data frame after r attempts is simply the probability F of having r successive failures, i.e.:

$$F := P_{\text{fail}}^r. \quad (1)$$

In that case, the total number of bits sent only comes from the sender (no acknowledgement is sent), and thus equals

$$\mathbf{H}_f := r \times \mathbf{D}. \quad (2)$$

With probability $1 - F$, the receiver gets the data frame within the r link-layer attempts. In that case, the total number of bits sent depends on the number of failures and partial failures before a success, if any (we only know here that at least one attempt led to a success or a partial failure). There are two possibilities:

- either all r attempts were failures or partial failures, but at least one was a partial failure (since we are in the case where the receiver got the data),
- or the last of the $k \leq r$ attempts was a success (in the sense of case (c) in Fig. 1).

Conditioned on the receiver getting the data frame within the r link-layer attempts, the expected total number of bits sent \mathbf{H}_s can therefore be computed as follows:

$$\mathbf{H}_s = \frac{1}{1 - F} \left(\sum_{i=1}^r \binom{r}{i} P_{\text{partial}}^i P_{\text{fail}}^{r-i} (r\mathbf{D} + i\mathbf{A}) + \sum_{k=1}^r P_{\text{succ}} \sum_{i=0}^{k-1} \binom{k-1}{i} P_{\text{partial}}^i P_{\text{fail}}^{k-1-i} (k\mathbf{D} + (i+1)\mathbf{A}) \right). \quad (3)$$

In (3), k stands for the index of the first success in r attempts (if any), and i is the number of partial failures among all attempts.

3.2 Multi-hop model

Let us now focus on the multi-hop case. An end-to-end transmission succeeds if the message reaches the destination after a certain number h of hops.

In this paper, we assume that link layer transmissions on each hop are independent. We denote B_i the bit error rate and F_i the frame error rate of the i^{th} hop. F_i is computed as per (1), taking $B = B_i$. Therefore, the probability that a frame is correctly received by a destination node is simply the probability Q_s that all h one-hop transmissions succeed, i.e.,

$$Q_s = \prod_{i=1}^h (1 - F_i) \quad (4)$$

where h is the number of hops from the sender to the receiver.

We assume here that the MAC layer is able to detect duplicate frames, such as those that are produced in case of a partial failure (case (b) in Fig. 1). This can be implemented, for instance, by using a sequence number in the MAC frame headers; when a node receives from one neighbor two successive frames with the same sequence number, it assumes that the corresponding Ack frame was lost and deletes the second frame⁴. This avoids the propagation of several copies of the same data frame.

Figure 2 shows the two possibilities for the outcome of the end-to-end transmission of a frame.

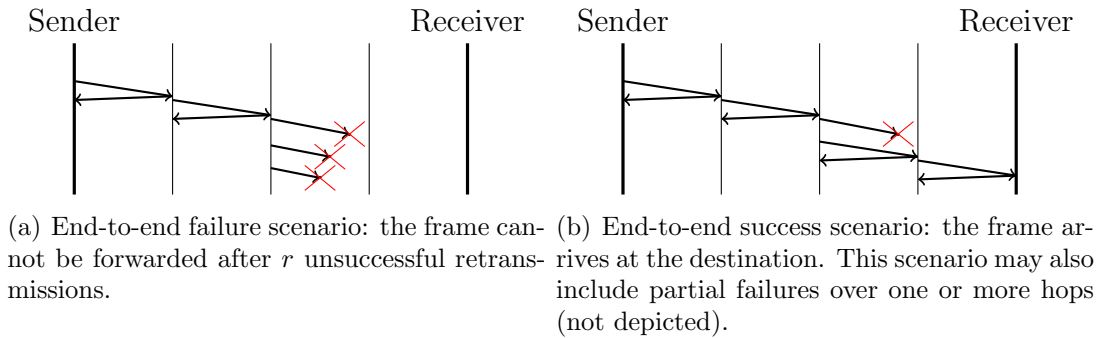


Figure 2: Failure and success scenarios in a multi-hop transmission.

Again, we express the expected number of bits sent, conditioned on the success of the end-to-end transmission (encompassing possible link-layer retransmissions, within the limit of r total attempts per hop).

- Knowing that the destination node correctly receives the message, the expected number of bits sent by all network nodes is simply

$$\mathbf{E}_s := \sum_{i=1}^h \mathbf{H}_{s_i}. \quad (5)$$

⁴Extending the current model to consider the case where there is no duplicate frame detection is left as future work. Note that the current 802.15.4 specification does not explicitly mandate such a procedure, though it is possible to implement it.

- Knowing that the message was lost in one of the h hops, the number of bits sent depends on the hop where the loss (i.e., the failure of all r attempts) occurs. The expected value then equals

$$\mathbf{E}_f := \frac{\sum_{k=1}^h (\sum_{i=1}^{k-1} \mathbf{H}_{s_i} + \mathbf{H}_{f_k}) \prod_{j=1}^{k-1} (1 - F_j) F_k}{1 - Q_s} \quad (6)$$

where \mathbf{H}_{s_i} and \mathbf{H}_{f_i} are computed using (3) and (2), respectively.

3.3 TCP performance

We now study the energy consumption of TCP in a multi-hop wireless network, taking into account the fact that TCP segments may be fragmented by lower layers if the total size of data frames (that depends on the TCP maximum segment size (MSS)) exceeds the MTU of those layers.

We can intuitively think of several opposite effects of the MSS, which suggest that a trade-off has to be found:

i) If TCP segments are fragmented, the loss of one fragment leads to the loss of the original TCP segment and therefore a new TCP end-to-end retransmission. Further, using short TCP segments (i.e., a small MSS) saves CPU power associated to fragmentation and reassembly. Moreover, sending several fragments (treated independently by lower layers) increases the probability of collision between two TCP data fragments, due to the hidden-node problem. As an illustration of that phenomenon, one can refer to Fig. 4, provided later on, where the number of collisions with long TCP segments is much larger than when short TCP segments are used.

ii) On the other hand, the use of long TCP segments reduces the number of TCP segments sent by the source, and thus, the TCP header overhead (which can be very large, for small MTU values) and the number of TCP acknowledgements.

To analyze such trade-off, we now apply the model presented in Section 3.2. Our objective here is to determine an MSS choice that optimizes TCP performance, in terms of energy consumption.

Let us consider that the MSS of TCP and the MTU of the MAC layer are such that each TCP segment is fragmented into m link-layer frames. This encompasses the particular case when no fragmentation occurs, that simply corresponds to $m = 1$. We will make the assumption that TCP ACKs always fit in a single MAC frame, though it would be easy to consider a more general case in which TCP ACKs are also fragmented. Note finally that we do not take into account the use of the delayed ACKs mechanism by the TCP receiver; including the effect of such a mechanism is left to future work.

To simplify the analysis, we assume that all fragments of a TCP data segment have the same total size \mathbf{D}_{DATA} , and all TCP acknowledgement frames have the same total size \mathbf{D}_{ACK} . At the destination, the segment is reconstructed from the received fragments. The loss of a single link-layer frame induces the loss of the whole TCP segment, and thus the retransmission of all m fragments. Further, we consider that the number of TCP retransmissions is not limited; that is, the TCP source keeps on sending a segment until it receives a TCP acknowledgement from the TCP receiver.

From the previous analysis, the success probability P_s of a TCP segment transmission attempt is simply the probability that all m data fragments be correctly sent to the

destination, and the TCP ACK be successfully sent back to the source:

$$P_s = Q_s^m \times Q_{s,\text{ack}} ,$$

where Q_s and $Q_{s,\text{ack}}$ denote the success probability of the multi-hop transmission of a TCP data fragment and of a TCP acknowledgement frame, respectively. Q_s and $Q_{s,\text{ack}}$ are simply obtained by applying (4), but replacing \mathbf{D} by respectively \mathbf{D}_{DATA} and \mathbf{D}_{ACK} .

In our model, each TCP data fragment transmission succeeds or fails independently of the others. Knowing that a transmission is successful *at the TCP level* (i.e., the TCP ACK is correctly received by the TCP source, which implies that all m fragments correctly reached the destination), the expected total number of bits sent by all nodes equals:

$$\mathbf{S}_s := \mathbf{E}_s \times m + \mathbf{E}_{s,\text{ack}}$$

As above, \mathbf{E}_s and $\mathbf{E}_{s,\text{ack}}$ are obtained from (5) using \mathbf{D}_{DATA} and \mathbf{D}_{ACK} , respectively, as the size of a frame. Likewise, we also define $\mathbf{E}_{f,\text{ack}}$ from (6).

Knowing that a TCP transmission attempt has failed, the expected number of bits sent end-to-end by all nodes is:

$$\mathbf{S}_f := \frac{1}{1 - P_s} \left[\underbrace{\mathbf{I}_f(1 - Q_s^m)}_{\substack{\text{end-to-end transmission failure} \\ \text{of one or more of the } m \text{ fragments}}} + \underbrace{(\mathbf{E}_s \times m + \mathbf{E}_{f,\text{ack}})Q_s^m(1 - Q_{s,\text{ack}})}_{\substack{\text{end-to-end transmission failure of the TCP ACK}}} \right],$$

where \mathbf{I}_f is the expected total number of bits sent for the m fragments to reach the destination, knowing that they (i.e., at least one) finally fail. Formally,

$$\mathbf{I}_f := \sum_{k=1}^m \binom{m}{k} (k\mathbf{E}_f + (m - k)\mathbf{E}_s)(1 - Q_s)^k(Q_s)^{m-k}.$$

After some algebra, we get

$$\mathbf{I}_f = m(1 - Q_s)\mathbf{E}_f + m\mathbf{E}_sQ_s(1 - Q_s^m).$$

To simplify the analysis, we will assume that the TCP window is equal to one TCP segment. This assumption is justified because a small window is a sensible choice for networks with a moderate number of hops [6]. Moreover, such small windows are typical of current TCP implementations for LLNs (e.g., Contiki OS [4]), since the memory and CPU constraints of wireless embedded devices make it difficult to fully implement TCP's congestion control mechanisms.

We can now compute the total number of bits that have to be sent by all nodes, to successfully transmit both a TCP segment and its corresponding TCP ACK. Since we assumed the number of TCP retransmissions is unbounded (i.e., the TCP sender keeps on retrying until the transmission succeeds), the mean number of transmissions for a given TCP segment equals $1/P_s$. This therefore corresponds to a total number of bits sent (per segment) of

$$\mathbf{S} := \mathbf{S}_f(1/P_s - 1) + \mathbf{S}_s.$$

Finally, to successfully send a given amount M of *application* (“useful”) data, the expected number of bits sent by all wireless nodes is the number of segments $\lceil M/\text{MSS} \rceil$, multiplied by the expected number of bits sent for a successful segment transmission. It is that final value that will be considered as representative of the overall energy consumption of the TCP transmission.

4 Results and discussion

We will now compare the predictions of our analytical model to simulation results, and discuss the tradeoff between sending long or short segments in different scenarios.

We apply our model to study the energy consumption of TCP on IPv6 over Low-Power Wireless Personal Area Networks [11], considering IEEE 802.15.4 [10] as the link-layer technology. The maximum link-layer frame size is thus equal to 127 bytes. The 6LoWPAN layer adapts the IPv6 datagrams to the link-layer MTU, as explained in [11].

We consider two MSS choices (MSS=64 and MSS=512 bytes) for a given TCP session. When the MSS is 64 bytes, no fragmentation is performed by the 6LoWPAN layer, whereas in the case where MSS=512 bytes, this adaptation layer splits each segment into 8 frames.

4.1 Energy model

To compute the energy consumed by wireless nodes to successfully send a data frame, we specify in this section a simple energy model. We denote by n the average number of neighbors of each node⁵. If a frame is sent by a wireless node, it is received by its n neighbors (assumed thus to be in an active listening state). Therefore the energy consumed to send a link-layer frame equals

$$E := (\text{TransmitEnergy} + n \times \text{ReceiveEnergy}) \times \text{FrameSize}$$

Table 2 shows the energy consumption values that we considered in our numerical computations. Those values correspond to a transmit and receive consumption currents of

Table 2: Energy parameters.

Parameter	Value
Transmit Energy	0.24 $\mu\text{J}/\text{bit}$
Receive Energy	0.21 $\mu\text{J}/\text{bit}$

20 mA and 17.7 mA, respectively, a tension of 3V and a transmission rate of 250 kbit/s (thus, a per-bit transmission duration of 4 μs). Such figures are typical of low-power wireless modules currently on the market (e.g., Crossbow TelosB⁶).

4.2 Model assessment

We first validate the results of our analytical model, through the simulation of the following scenario. A TCP sender sends a short file (51.2 Kbytes) to a TCP receiver. We consider an average size of 20 bytes for IPv6 headers, thanks to the LOWPAN_IPHC [13] compression. The simulation results plotted are average values after 30 simulation runs. The TCP window size is set to 1, which implies that no congestion losses occur (we only consider one flow here). As a result, retransmissions can only be due to transmission errors, as described in Section 3. For the sake of simplicity, all links have the same bit-error

⁵In the plotted curves, we considered $n = 2$.

⁶http://www.hoskin.qc.ca/uploadpdf/Instrumentation/CrossBow/hoskin.TPR2400CA_42efb73715b8b.pdf

rate $B_i = B, i = 1, \dots, h$. Unless indicated otherwise, the parameters used in simulations and in numerical computations correspond to the default values shown in Table 3. Finally, as in Section 3.3, the maximum number of transport-layer retransmissions is not bounded.

Table 3: Default simulation parameters

Parameter	Value
h	5
r	3
α	0
BER B	3×10^{-4}
Link-layer Ack frame size	40 bits
Link-layer data frame header	120 bits
IP header	160 bits
TCP header	160 bits

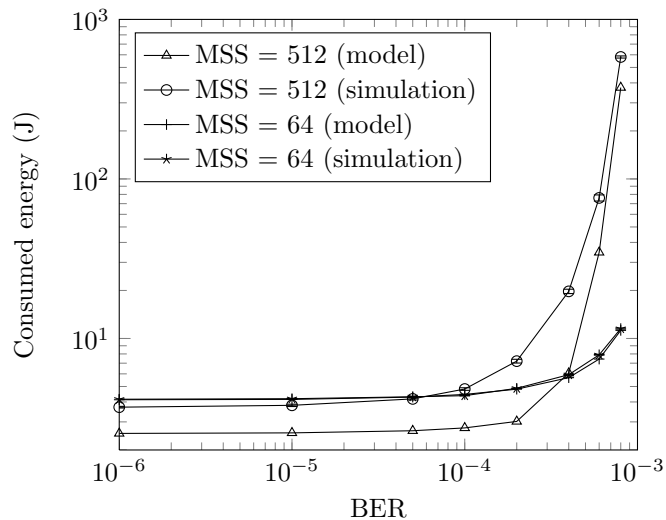


Figure 3: Energy consumption with long or short TCP segments, as a function of the BER B .

Figure 3 shows that analytical results closely match simulations results when a short TCP segment size is chosen. However, we can see that with long TCP segments, the model tends to underestimate the energy consumption. This difference comes from the presence of collisions between fragments of a given TCP data segment, that are not taken into account in the analytical model but do occur in simulation, as illustrated in Fig. 4. In the figure, nodes 1 and 6 are the source and destination TCP nodes, respectively, and node 2 to 5 are intermediate nodes.

Moreover, Fig. 3 shows that using short TCP segments becomes interesting, from an energy point of view, when the bit-error rate is high (above 10^{-4}). For example, for a BER of 4×10^{-4} , the total consumed energy with $MSS = 512$ bytes is three times larger than with $MSS = 64$ bytes. For low values of B , the number of retransmissions is small, hence the energy consumption becomes roughly independent of B .

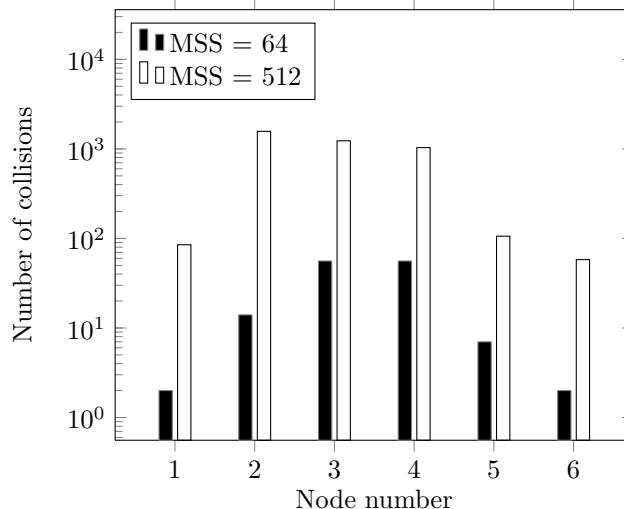


Figure 4: Number of collisions in a multi-hop scenario.

We remark on Figs. 3 and 5 that simulation results fit better our analytical results when no fragmentation is performed by the link layer. This is again due to collisions that occur in the large-MSS case and are not encompassed by our model. Fig. 5 also illustrates that increasing the number of link-layer attempts decreases the energy consumption, especially so when there is fragmentation (i.e., a large MSS). Indeed, giving the link layer more chances to send a data frame to its next-hop node reduces the discard probability of TCP segments, and therefore the number of end-to-end retransmissions.

In the following, we study the impact of the other system parameters on the energy consumption, based on the model only. We should therefore stay aware that the energy consumption may be a little underestimated for large values of the MSS.

4.3 FEC redundancy ratio and energy consumption

We first study the impact of FEC on the energy consumption. Fig. 6 shows that there seems to be an optimal amount of redundancy, in terms of energy consumption. Below the optimal value, adding redundancy reduces the probability of losses, and thus reduces the energy consumption. When α is above such optimal value, however, energy expenditure steadily increases because of the redundancy overhead.

Recall that the MTU being fixed, the number of frames to send depends on the redundancy ratio according to a staircase function, hence the discontinuities in the figure.

4.4 Selecting the TCP MSS to minimize energy consumption

Depending of the numerous parameters of a given scenario, it appears that the same MSS value is not always the most efficient one in terms of energy. We now intend to summarize the effect of all parameters, by focusing on the best MSS strategy to implement. In the following figures, we compare the two values $MSS = 64$ and $MSS = 512$, and we concentrate on the boundary values, that delimitate zones where one MSS value outperforms the other.

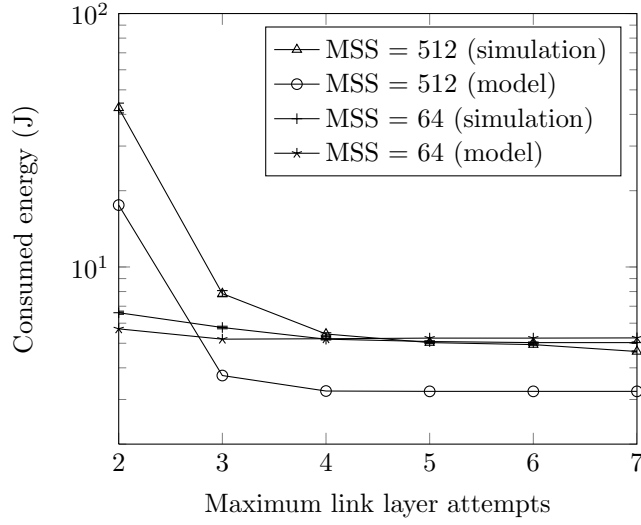


Figure 5: Energy consumption with short or long TCP segments, as a function of the number of link layer attempts r (with $B = 5 \times 10^{-4}$).

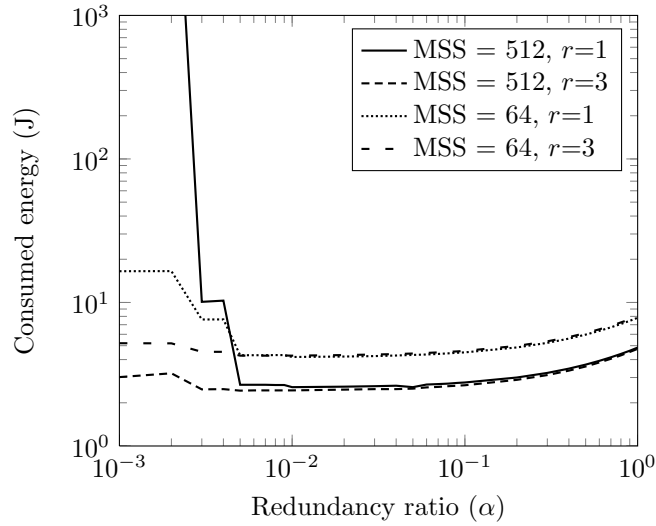


Figure 6: Consumed energy using short or long TCP segment, as a function of the redundancy ratio α ($B = 3 \times 10^{-4}$, $h = 5$).

We plot those frontier curves in Figs. 7 and 8. In each figure, the area above the curve represents the case where a TCP MSS value of 64 bytes consumes less energy than an MSS of 512 bytes, while the opposite holds below the curves.

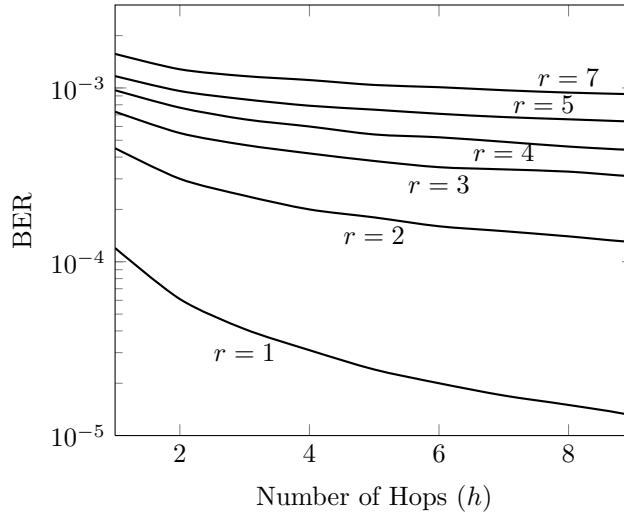


Figure 7: Long (MSS=512 bytes) versus short (MSS=64 bytes) in a multi-hop TCP transmission: prefer the short MSS above the curves, the long one below.

Figure 7 shows the two zones, depending on the transmission distance and the BER. We remark that for a given BER, short MSSs tend to outperform long MSSs when the distance grows: it is more and more interesting to use short MSS values instead of long ones. Indeed, for large networks the cost of end-to-end retransmissions will exceed the potential economy in terms of TCP header overhead. For a given distance, short MSSs are better suited for large BER environments since (energy-spending) segment retransmissions tend to occur more frequently. As previously observed, increasing the maximum number of link layer attempts r reduces the one-hop transmission failures, and thus limit the effects of errors, hence favoring long MSSs.

Finally, Fig. 8 illustrates the effect of FEC mechanisms. Not surprisingly (the FEC reducing the effect of transmission errors), redundancy makes large MSSs outperforms small MSSs due to the overhead reduction they allow.

5 Conclusions and Future Work

In this paper, we have proposed an analytical model to estimate the number of bits sent by all wireless nodes in a TCP session in a Low power and Lossy Network, in order to evaluate the overall energy consumption. The model has been validated through simulations, using the INETMANET framework [18] of the OMNet++ network simulator, in the context of TCP over 6LoWPANs.

Our main outcomes regard the choice of an energy-saving Maximum Segment Size (MSS) for TCP. We have shown that using a large TCP segment size is less energy-consuming in small, low-error networks, while it becomes interesting to reduce the MSS

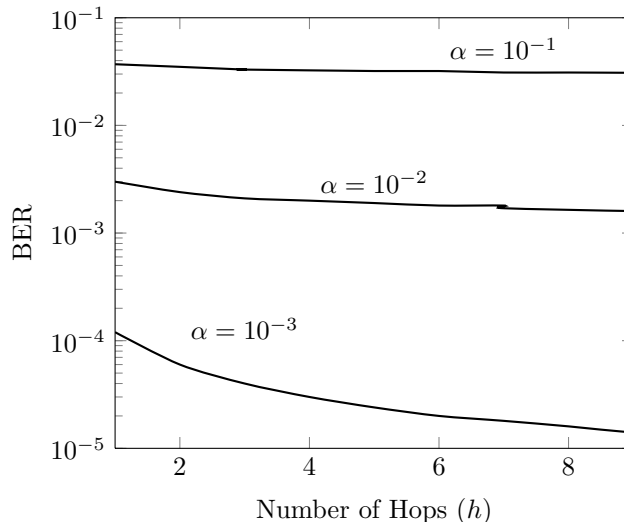


Figure 8: Long (MSS=512 bytes) versus short (MSS=64 bytes) in a multi-hop TCP transmission: prefer the short MSS above the curves, the long one below.

when the network is large or very lossy. The impact of the number of attempts at the link layer, as well as the use of FEC, has also been studied.

The work presented here could be extended in several ways. A first interesting direction would be to model the collision process, that has been observed in our simulations for large MSS values. We would also like to consider the case when duplicate frames are not detected at the link layer; likewise, the transport layer modeling could be extended to encompass TCP's delayed acknowledgement mechanism, and also larger TCP windows. Finally, we are currently investigating an adaptation algorithm which dynamically adjusts the TCP segment size to the optimal MSS value.

6 Acknowledgements

The work of A. Ayadi has been funded by the Pôle de Recherche Avancée en Communications (PRACom). We would like to thank Ahmed Triki for his help with the simulations.

References

- [1] S. Bansal, R. Shorey, R. Gupta, and A. Misra. Energy efficiency and capacity for TCP traffic in multi-hop wireless networks. *Wireless Networks*, 12(1):5–21, 2006.
- [2] D. Barman, I. Matta, E. Altman, and R. El Azouzi. TCP Optimization through FEC, ARQ, and Transmission Power Tradeoffs. In *Proceedings of WWIC*, pages 87–98, 2004.
- [3] A. Dunkels, J. Alonso, and T. Voigt. Distributed TCP Caching for Wireless Sensor Networks. In *Proceedings of 3rd Annual Mediterranean Ad-Hoc Networks Workshop*, 2004.

- [4] A. Dunkels, B. Grönvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of First IEEE Workshop on Embedded Networked Sensors (Emnets-I)*, Tampa, Florida, USA, November 2004.
- [5] G. Fairhurst and L. Wood. Advice to link designers on link Automatic Repeat reQuest (ARQ). RFC 3366, IETF, 2002.
- [6] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla. The Impact of Multihop Wireless Channel on TCP Performance. *IEEE Transactions on Mobile Computing*, 4(2):209–221, 2005.
- [7] L. Galluccio, G. Morabito, and S. Palazzo. An Analytical Study of a Tradeoff Between Transmission Power and FEC for TCP Optimization in Wireless Networks. In *Proceedings IEEE INFOCOM*, pages 1765–1773, 2003.
- [8] A. Al Hanbali and E. Altman. A Survey of TCP over Ad Hoc Networks. *IEEE Communications Surveys & Tutorials*, 7:22–36, 2005.
- [9] L. Heng and G. Ai-huang. Energy Efficiency Comparison of SACK, Tahoe, Reno and NewReno Over Ad Hoc Network Based on Analytic Hierarchy Process. In *Proceedings of International Conference on Wireless Communications Networking and Mobile Computing*, pages 1–4, sept. 2006.
- [10] IEEE Computer Society. IEEE Std. 802.15.4-2006, October 2006.
- [11] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919, IETF, 2007.
- [12] W. Lilakiatsakun and A. Seneviratne. Enhancing TCP energy efficiency for mobile hosts. In *Proceedings of 10th IEEE International Conference on Networks*, pages 235 – 239, 2002.
- [13] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, IETF, September 2007.
- [14] S. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz. GARUDA: Achieving Effective Reliability for Downstream Communication in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 7(2):214–230, 2008.
- [15] J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. Updated by RFC 3168.
- [16] I. Reed and S. Golomb. Polynomial codes over certain finite fields. *Joint Society of Industrial and Applied Mathematics Journal*, 8(2):300–304, June 1960.
- [17] A. Seddik-Ghaleb, Y. Ghamri-Doudane, and S.-M. Senouci. A performance study of TCP variants in terms of energy consumption and average goodput within a static ad hoc environment. In *Proceedings of 2006 international conference on Wireless communications and mobile computing*, pages 503–508, USA, 2006.

- [18] A. Vagas. INETMANET framework, May 2010.
- [19] C.-Y. Wan, A.T. Campbell, and L. Krishnamurthy. Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):862–872, April 2005.
- [20] B. Wang and S. Singh. Computational Energy Cost of TCP. In *Proceedings of IEEE INFOCOM*, pages 206–216, 2004.
- [21] M. Zorzi and Ramesh R. Rao. Energy efficiency of TCP in a local wireless environment. *Mobile Networks and Applications*, 6(3):265–278, 2001.

www.telecom-bretagne.eu

Campus de Brest

Technopôle Brest-Iroise

CS 83818

29238 Brest Cedex 3

France

Tél. : + 33 (0)2 29 00 11 11

Fax : + 33 (0)2 29 00 10 00

Campus de Rennes

2, rue de la Châtaigneraie

CS 17607

35576 Cesson Sévigné Cedex

France

Tél. : + 33 (0)2 99 12 70 00

Fax : + 33 (0)2 99 12 70 19

Campus de Toulouse

10, avenue Edouard Belin

BP 44004

31028 Toulouse Cedex 04

France

Tél. : +33 (0)5 61 33 83 65

Fax : +33 (0)5 61 33 83 75

© Télécom Bretagne, 2010

Imprimé à Télécom Bretagne

Dépôt légal : novembre 2010

ISSN : 1255-2275

