Energy-Efficient Method to Improve TCP Performance for MANETs

Chaoyue Xiong, Jaegeol Yim, Jason Leigh and Tadao Murata
Computer Science Department, University of Illinois at Chicago
Chicago, IL 60607, USA

ABSTRACT

The current implementation of TCP for the Internet is not efficient when used for Mobile Ad hoc Networks (MANETs). This is because TCP assumes that all packet losses are caused by congestion, whereas transmission errors are a main reason for packet losses in wireless networks. To ameliorate this situation and increase performance, we propose a method of using multi-metric parameters to distinguish the causes for packet losses and use Colored Petri Net to analyze the revised protocol. We call this TCP-MEDX (Mobile Error Detection eXtension). TCP-MEDX has two characteristics: Firstly, it is energy-efficient because this solution is only initiated when a packet loss is detected. This characteristic is very important for MANETs because of its limited power source; secondly, our approach removes negative effects caused by asymmetry in wireless links, thus improving correctness in determining causes for packet losses. Our simulation results using Design/CPN show that the proposed approach increases throughput and reduces propagation delay compared with standard TCP.

Keywords: Transmission Control Protocol, Colored Petri Nets, Propagation Delay, Throughput, Congestion.

1. INTRODUCTION

Transmission Control Protocol (TCP) has been the dominant transport-layer protocol for reliable data delivery over the Internet, and it has been the main research target for many network researchers. Some of them use simulation to study TCP performance, others use mathematical model to analyze algorithms in TCP. Our prior work with Colored Petri Nets (CPN) [11] has shown them to be a powerful tool for simulating and analyzing TCP protocols [9]. CPNs have a graphical form underlying on mathematical definition and it is the method used in this paper. CPNs together with Design/CPN provide both simulation and formal model to analyze TCP.

TCP assumes that the underlying network is relatively reliable because it was designed for a wired environment. When packet loss occurs, the TCP sender interprets it as evidence of congestion in the network, thus invoking congestion control by drastically reducing its congestion window. This mechanism ensures fairness among connections sharing the same channel, and works well in a wired network, where most of the packet losses are due to congestion. However, it performs poorly in Mobile Ad hoc Networks (MANETs) because communication over MANETs is characterized by high bit-error rates. Packet losses occur more frequently due to transmission errors than due to congestion in MANETs. However, TCP has no mechanism to detect the causes for packet losses but instead interprets all packet losses as those resulting from congestion, thus causing severe performance degradation when used for MANETs. In order to improve TCP performance over MANETs, where wireless links are subjected to high transmission errors, it is necessary to revise TCP, so that it can detect the causes for packet loss. This paper proposes a reactive approach called TCP-MEDX that calculates metrics to detect the causes for packet loss *only* when packets are lost, making the approach energy-efficient.

The rest of the paper is organized as follows: Section 2 describes the proposed approach. Section 3 describes CPN model for TCP-MEDX and simulation environment in Design/CPN, and simulation results are given in section 4. Section 5 introduces related work. Conclusions are given in Section 6.

2. PROPOSED APPROACH

For energy efficiency, our proposed mechanism is activated only when packet loss is detected by assuming that a certain relationship between congestion and propagation delay exists. TCP Vegas [10], for example, similarly adopts this assumption to change its congestion window proactively to avoid serious congestion. In the mechanism, we use the following two parameters to check the occurrence of congestion:

1) Propagation delay

Propagation delay is a better metric for indicating congestion than Round Trip Time (RTT) in MANETs for the following reason. Wireless links are usually asymmetric, the time that a packet spent on a forwarding path may not be equal to the time spent on its corresponding backward path. Thus, propagation delay is not equal to ½RTT. Thus, the use of propagation delay rather than RTT removes this asymmetry and increases accuracy in detecting congestion.

Our mechanism uses average propagation delay as a threshold. Average propagation delay $\overline{P}$ is calculated by the following equation, which is similar to the RTT calculation in TCP:

$$\overline{P} = \alpha \overline{P} + (1-\alpha) P_{new} \quad (1)$$

where α is a parameter, which is chosen between 0 and 1; P_{new} is the most recently received propagation delay.

2) Differences between propagation delays.

The degree of difference between propagation delays is also an indication of congestion. If congestion occurs, more and more packets are queued in router buffers that are located along the path to the packets' destination. The propagation delay will continually increase until the congestion is cleared. In the worst case, the buffer will overflow causing the router to discard packets. Thus, when congestion occurs, the difference between propagation delays will be non-zero.

Using the above two metrics, we adopt the following criteria to decide whether congestion occurs or not.

Congestion criteria: TCP-MEDX considers that a network is in a state of congestion if both of the following conditions are satisfied:

I. The current propagation delay is far beyond a threshold, which is equal to $\beta\overline{P}$ where $\overline{P}$ is an average propagation delay and β is specified by the application.

II. Propagation delay keeps increasing for a certain number of packets.

When packet loss is detected, TCP-MEDX initiates the above two criteria to decide whether a congestion indeed occurred or not. If congestion did occur, it halves its congestion window. Otherwise, TCP-MEDX assumes that the packet loss is caused by a transmission error and does not change the congestion window.

To implement the above two criteria, we add the following revisions to standard TCP at the receiver and sender sides, respectively.

At the receiver side: when a packet is received, it calculates the propagation delay which is equal to (T_{recv} - T_{snd}). T_{recv} is the receiving time and T_{snd} is the time when the packet is sent by the source. The propagation delay is attached to the corresponding acknowledgement and sent back to the sender by the receiver. For sender to obtain the most updated network dynamics, the protocol does not adopt delayed acknowledgment and requires the receiver to acknowledge every incoming packet.

At the sender side: Sender obtains propagation delay from acknowledgement and saves it in its propagation buffer. At the same time, the sender updates its average propagation delay. The length of the propagation buffer may be specified by the application[1].

Choice of parameters α and β

α and β have major influence on the performance of TCP-MEDX. For a MANET with high mobility, propagation delay changes frequently, so α should be higher, so that it can immediately reflect the most recent update of propagation delay. However, network dynamics and characteristics inherent in wireless network affects propagation delay and in order to smooth out these noisy influences, α cannot be too high. To balance its tradeoff, α is chosen between 0.2~0.8. In our simulation, we choose α at 0.5. β reflects deviation of a propagation delay away from weighted average propagation delay. It can be chosen from experience from a specific network. We choose β at 1.2.

3. SIMULATION ENVIRONMENT

Our prior work with Colored Petri Nets (CPN) [11] has shown them to be a powerful tool for simulating and analyzing TCP protocols [9]. A simulation environment called Design/CPN [12] was used in that work. CPNs can conveniently express non-determinism, concurrency and different levels of abstraction that are inherent in protocols. CPNs have a graphical form underlying on mathematical definition. Design/CPN is a suite of tools for editing, simulating and analyzing CPNs, and it has a graphical editor that allows users to create and layout different net components. One of its nice features is that Design/CPN uses pages to divide the model into smaller components, enhancing the maintainability and readability without affecting the execution or analysis of the model. A CPN on a subpage, which provides a more precise and detailed description of an activity, is represented by a *substitution transition* on a higher-level page. A non-hierarchical view of the Design/CPN net would involve replacing the substitution transitions with their corresponding subpages. Another important feature of Design/CPN is its supports for timed modeling and simulation, which allows us to easily implement the fuzzy time functions [13].

The TCP-MEDX protocol modeled here is an extension of our prior TCP model with two modifications to the channel module to simulate packet loss. These include:

- modeling transmission errors with a probability (which is fixed during all the simulation time;)
- designing a congestion pattern that can be easily modified to model different congestion situations.

The revised channel module is shown in Fig. 1. It is well known that the transmission error rate of a MANET is substantial, and hence it is set at 0.2 in this model. The transmission error is modeled by the function attached to the arc from the transition: '*Transmit Packet*' to the place: '*arrive*'. A bottleneck router is also imposed on the

[1] The threshold will be determined empirically for simulation results or real test bed experiments

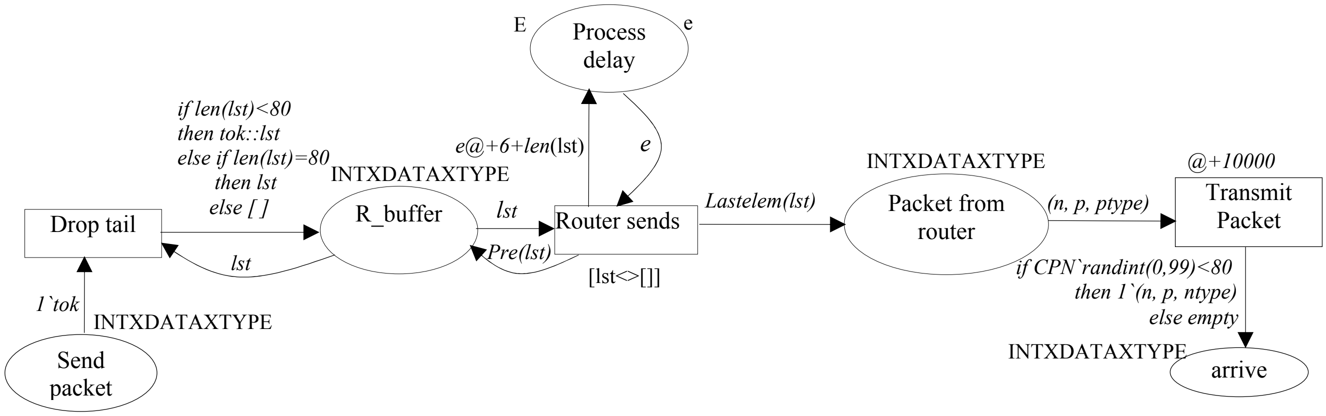

Fig. 1 Channel module of TCP-MEDX

channel. The router has a First-In-First-Out (FIFO) buffer for saving segments temporarily. The maximum length of the buffer is 80 segments. We call the list of segments of the buffer, '*lst*'. When the buffer length is less than the maximum length, an arrived segment is added to the end of '*lst*', but if the buffer is full, a newly arrived segment is discarded. This is done by the function attached to the arc from the transition: '*Drop tail*' to the place: '*R_buffer'*.

Fig. 2 shows our data generation module. In this module, tokens represent segments in TCP protocol. Based on characteristics of an application, segments may be sent in a burst mode or in a continuous mode. In a burst mode, many segments are sent in a small interval and may result in congestion over the network. Corresponding to these two modes, tokens can be of two types and are generated by transition: "*Gen. Tokens with traffic type*". Type "*O*" represents ordinary traffic and type "*C*" represents congestion traffic. The proportion of these two types is specified by the arc inscription from transition: "*Gen tokens with traffic type*" to place: "*with traffic type*":

if CPN`randint(0, 999) < 900 then 1`(e, O) else 1`(e, C).

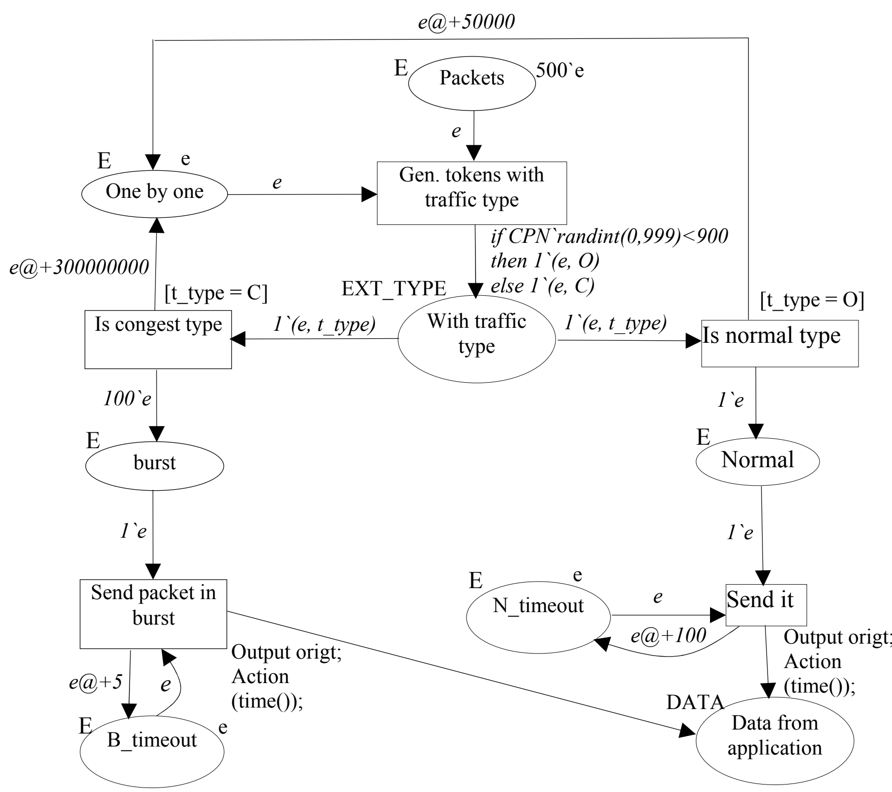

Fig. 2 Data generation module of TCP-MEDX

Our congestion algorithm is shown in Fig. 3.

update_prop_list()

1 **when** (receives an acknowledgment)
2 update propagation delay list;

predict_loss_cause()

3 **if** (packet is lost)
4 bool congest ← false;
/* true means a network is in congestion state*/
5 **if** (the weighted sum of propagation delay in the list> $\beta \overline{P}$)
6 congest ← true;
7 **for** (all the propagation delay in the list)
8 **if** (propagation delay < previous propagation delay)
9 congest ← false;
10 break;

Fig. 3 Congestion control algorithm of TCP-MEDX at the sender

When an acknowledgment arrives at the sender, the sender removes the oldest propagation delay from head and appends the most recent propagation delay at the tail (line 1,2). When a packet loss is detected, the sender checks if the congestion occurs or not (line 3~10). If it predicts the occurrence of congestion, the function returns true.

4. SIMULATION RESULTS

In this simulation, the data generation module has options to work in two modes, *normal* and *congested*. The mode thst the module is in depends on a probability. In our simulation, we chose 90% for *congested* and 10% for *normal*, so that we can test the degree of correctness in predicting loss type. In the *normal* mode, the data generation module generates a packet every 50,000 time units. In a *congested* mode, the data generation module generates burst packet every 500 time units.

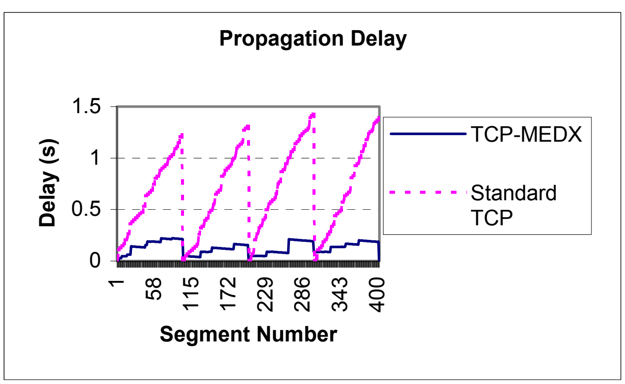

Fig. 4. Comparison of propagation delay for standard TCP and TCP-MEDX

Model Criteria	Revised	Original
Number of Segments	1069	201

Table 1. Throughput comparison

Fig. 4 shows that propagation delays in TCP-MEDX are indeed much lower than those in standard TCP. The reduced propagation delay shows that when packet loss is caused by congestion, TCP-MEDX is able to detect it with a high degree of accuracy. Let us assume for a moment that our mechanism *misjudges* packet loss caused by congestion, as loss caused by transmission error. When this happens the sender will not decrease its congestion window as expected. As a result, congestion will increase dramatically and many successive packets will be discarded, leading to many retransmissions, and thus increasing propagation delay. Fig. 4 shows that this is not the case with TCP-MEDX.

Table 1. compares the throughput between TCP-MEDX and standard TCP. We used the same data generation model and a fixed simulation duration for the two TCPs. After running 4 seconds, we found the total throughput from the original TCP is: 201 segments, while the total throughput from the TCP-MEDX is: 1,069 segments. The increased throughput of TCP-MEDX is further evidence that the scheme is correctly identifying the cause of packet losses. If we were to assume for a moment that TCP-MEDX was mistaking the cause of packet loss from transmission error, as loss from congestion, the sender will always half its congestion window, thus lowering the throughput, which is contrary to the results in Table 1.

Both results strongly suggest that TCP-MEDX can indeed identify the cause of packet loss accurately.

5. RELATED WORK

Prior proposals for improving TCP performance in mobile wireless networks [2, 3] require cooperation from an intermediate node to generate a notification message when the node detects congestion. Shagdar et. al [2] proposes Explicit Wireless Loss Notification (EWLN), which uses information from the Medium Access (MAC) layer to distinguish between the two causes of packet losses. Chawla and Nandi [3] propose Freeze and Explicit Congestion Notification (FECN) that uses Explicit Congestion Notification (ECN) [8] to monitor congestion. These techniques improve TCP performance in MANETs under certain situations, but they incur implementation cost and energy consumption at mobile nodes along a transmission path. Thus, they may be difficult to apply in practice because energy-efficiency is an important consideration in MANETs. It is therefore meaningful for MANETs to adopt an end-to-end mechanism which imposes demands only on the sender or the receiver. In this paper, we are primarily interested in an end-to-end solution.

A number of effective end-to-end solutions have been proposed in the past [4] [7] . Barman et. al [4] detects congestion using its correlation with RTT. This approach is sound, but the probability of false detection can sometimes be high, which significantly reduces its performance. Fu [7] applies multi-metric parameters instead of a single metric in [4] to detect the nature of packet losses and hence is able to improve detection accuracy . But it also imposes high computation cost at the receiver because it must calculate at least two metrics when each packet is received. Our reactive scheme is more energy-efficient because it calculates metrics only when a packet is lost.

6. CONCLUSIONS

We have proposed a method to revise TCP to meet the special characteristics of Mobile Ad hoc Networks (MANETs). TCP-MEDX works by properly identifying the cause of packet loss in a TCP stream, and shrinking congestion window only when it is caused by congestion rather than transmission error. This has the effect of increasing throughput and decreasing propagation delay.

7. REFERENCES

[1] S. Hadjiefthymiades, S. Papayiannis, and L. Merakos, “Using path prediction to improve TCP performance in wireless/mobile communications”, *IEEE Communications Magazine*, Vol.40, Issue: 8, August 2002, pp. 54 –61.

[2] O. Shagdar, M. N. Shirazi and B. Zhang, “Improving ECN-based TCP performance over wireless networks using a homogeneous implementation of EWLN”, Telecommunications, 2003. ICT 2003. 10th International Conference on, Vol.1, 2003, pp. 812 –817.

[3] R Chawla, and S. Nandi, “TCP FECN: a unified solution for wireless networks” Communication Systems, 2002. ICCS 2002. The 8th International Conference on, Vol. 2, 2002, pp. 815 –819.

[4] D. Barman and I. Matta, “Effectiveness of loss labeling in improving TCP performance in wired/wireless networks”, 10th IEEE International Conference on Network Protocols (ICNP'02), November 12 - 15, 2002.

[5] B. S. Bakshi, P. Krishna, N.H. Vaidya and D.K.Pradhan, “Improving performance of TCP over wireless networks”, International Conference on Distributed Computing Systems, 1997.

[6] S. B. Moon, “Measurement and analysis of end-to-end delay and loss in the Internet”, Ph.D. thesis, University of Massachussets at Amherst, Feb. 2000.

[7] Z. Fu, B. Greenstein, X. Meng and S. Wu, “design and implementation of a TCP-friendly transport protocol for ad hoc wireless networks”, 10th IEEE International Conference on Network Protocols (ICNP'02).

[8] S. Floyd, “TCP and explicit congestion notification”, ACM SIGCOMM Computer Communication Review, V.25, No. 5, pp. 8-23, 1994.

[9]Y. Zhou, T. Murata, and T. DeFanti, "Modeling and performance analysis using extended fuzzy-timing Petri nets for networked virtual environments," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol.30, No.5, October 2000.

[10] L. Brakmo and L. Peterson. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communication*, Vol 13, No. 8 (October 1995) pp. 1465-1480.

[11] K. Jensen, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concept*s. Monographs in Theoretical Computer Science, Springer-Verlag, 2nd corrected printing, 1997.

[12] K. Jensen, Design/CPN [Online]. Dept. Computer Science, Univ. Aarhus, Denmark. Available: http://www.daimi.au.dk/designCPN/.

[13] T. Murata, T. Suzuki and S. Shatz, "Fuzzy-Timing High-Level Petri Nets (FTHNs) for Time-Critical Systems," in J. Cardoso and H. Camargo (editors) "Fuzziness in Petri Nets" Vol. 22 in the series "Studies in Fuzziness and Soft Computing" by Springer Verlag, New York, pp. 88-114, 1999.