1000-9825/2005/16(05)0878 ©2005 Journal of Software 软件学报 Vol.16, No.5

基于无线自组织网络的TCP Freeze-Probing改进协议*

王 波+, 范平志

(西南交通大学 移动通信研究所,四川 成都 610031)

TCP Freeze-Probing Enhancement for Mobile Ad Hoc Networks

WANG Bo+, FAN Ping-Zhi

(Institute of Mobile Communications, Southwest Jiaotong University, Chengdu 610031, China)

+ Corresponding author: Phn: +1-517-353-9203, E-mail: wangbo1@msu.edu

Received 2003-10-13; Accepted 2004-09-08

Wang B, Fang PZ. TCP freeze-probing enhancement for mobile ad hoc networks. *Journal of Software*, 2005,16(5):878−885. DOI: 10.1360/jos160878

Abstract: Traditional Transmission Control Protocol (TCP) works well in wired network but suffers from performance degradation in mobile ad-hoc networks (MANET) due to the fact that it cannot distinguish packet losses due to congestion from packet losses due to link breakage, channel error and route changes. In this paper, an enhanced TCP, named TCP Freeze-Probing, is proposed to improve the TCP performance in mobile ad-hoc networks. TCP Freeze-Probing is an end-to-end approach that does not need the cooperation of the intermediate nodes in the network. Besides, a throughput model for TCP Freeze-Probing is given, which is validated through simulation. It is shown by analysis and simulation that the proposed approach can greatly improve the TCP performance in MANET.

Key words: TCP; MANET; freeze probing

摘 要: 传统的TCP协议在有线网络中能够良好地工作,但用于无线自组织网络时则性能有所下降.其原因在于,传统的TCP协议无法分辨网络丢包原因,如网络拥塞、链路断开、信道错误或者链路改变.为了提高TCP协议在无线自组织网络中的性能,提出了一种TCP协议的改进方案TCP Freeze-Probing.该方案是一种端到端方法,不需要网络中间节点的反馈合作.同时,提出了一种基于TCP Freeze-Probing的吞吐量模型并利用仿真对模型进行了验证.分析和仿真结果表明,该方案能够有效地改进TCP在无线自组织网络的性能.

关键词: TCP协议;无线自组织网络;冻结探测

中图法分类号: TP393 **文献标识码**: A

---

* Supported by the National Natural Science Foundation of China under Grant No.60272070 (国家自然科学基金); the University Doctorial Research Foundation, Ministry of Education of China under Grant No.20020613020 (高校博士点科研基金)

WANG Bo was born in 1979. He is now a Ph.D. student at the CSE Department of Michigan State University. His current research areas are mobile ad hoc networks and wireless sensor networks. **FAN Ping-Zhi** was born in 1955. He is a professor and doctoral supervisor at the Institute of Mobile Communications, Southwest Jiaotong University. His research areas are CDMA communications, information and coding theory, multimedia wideband radio communications, network and information security, etc.

1 Introduction

Mobile ad hoc network (MANET) is a special peer-to-peer network without fixed infrastructure such as base station. Mobile nodes are connected by wireless links and each node acts as a router in the network. TCP appears to be a de facto transmission protocol in wired and wireless network, but recent researches show that TCP performs poorly in MANET[1−8]. Although there are several reasons for the poor performance of TCP in MANET, the main factor lies in that traditional TCP such as TCP Reno cannot distinguish packet losses due to congestion from packet losses due to other reasons such as channel error, route change and link breakage. However, in MANET environment, the channel errors, route changes and link breakages are inherent, resulting in the TCP performance degradation in MANET due to the incorrect invocation of congestion control mechanism.

According to Refs.[3,4], an effective approach for improving TCP performance in MANET should have the ability to determine the causes of packet losses. TCP should then act properly according to the type of losses.

In this paper, an enhanced TCP for MANET, named TCP Freeze-Probing which is an end node detection approach, is proposed. The main idea of this approach is to divide the causes of the packet losses into two classes: short-term recoverable network anomalies and long-term network anomalies, instead of determining the specific packet loss reasons mentioned above. Analytical and simulation results show that our approach can greatly improve the TCP performance in MANET.

The rest of this paper is organized as follows: in Section 2, we introduce the related work; in Section 3, detailed description on the new TCP Freeze-Probing enhancement is presented; in Section 4, an analytical model for determining the throughput of the TCP Freeze-Probing enhancement is proposed; and finally, in Section 5, simulation results and comparative discussions are given, followed by some concluding remarks in the last section.

2 Related Work

Recently, several approaches have been proposed to improve the TCP performance for MANET. In Ref.[2], Ruy de Oliveira and Torsten Braun classified these approaches into two categories: end node detection approaches and network detection approaches. TCP ELFN[1], TCP-Feedback[5] and ATCP[6], are examples of the network detection approaches relying heavily on the network explicit notification for detecting the packet loss reasons. On the other hand, ADTCP[4], TCP DOOR[7], Fixed RTO[8], are examples of the end node detection approaches relying only on some implemented algorithms in end node to detect the packet loss reasons instead of using network feedback.

In fact, both approaches have their advantages and disadvantages. For network detection approach, the explicit notification from the intermediate nodes in the network can provide TCP with accurate information about the causes of the packet loss and therefore enabling proper action accordingly. However, the implementation of network detection is expensive and sometimes unrealistic in MANET. In addition, the network-based detection may cause security problems. As for end node detection approach, there is no need to request cooperation from the intermediate nodes, but the information provided by end nodes may not be as accurate as network feedback.

Recently, the research on designing new transport layer protocols for MAENT has received an increasing attention. In Ref.[15], the authors showed through detailed arguments and simulations that several design elements in TCP are fundamentally inappropriate for the unique characteristics of ad-hoc networks. They present a new reliable transport layer protocol for ad-hoc networks called ATP (ad-hoc transport protocol). Strictly speaking, ATP is a totally new transport protocol instead of an enhancement of TCP. In Ref.[16], the authors proposed a fair hop-by-hop congestion control algorithm for wireless multi-hop network. Although these approaches are more effective than traditional TCP, due to the wide range of TCP-based applications, their application may not realistic in MANET for now.

TCP ELFN is used as a comparison basis for many TCP enhancements, so we will also use it as a reference to our analysis and simulation. TCP ELFN cooperates with routing protocol DSR[9] to detect the link breakage, thus enables TCP to distinguish the packet losses due to congestion from the packet losses as a result of the link breakage. When an intermediate node in the network detects the link breakage, it sends an Explicit Link Failure Notification (ELFN) to TCP sender. This kind of message is carried by the route error message of DSR. Upon receiving ELFN, TCP sender enters into the stand-by (frozen) mode. TCP retransmission timer is frozen during this mode and normal data transfer is interrupted. Then some probe packets are sent to network to determine the network condition. When an acknowledgement (ACK) of probe packet is received by TCP sender, TCP leaves stand-by mode and resumes normal transfer. By using ELFN, TCP avoids the exponential backoff mechanism when losses take place by factors other than congestion, and greatly improves the TCP performance in MANET. Some limitations of ELFN are given in Ref.[10]. In fact, as to the factors that cause packet losses, the author of TCP ELFN focuses mainly on the link breakage, and the related simulation results in Ref.[1] are conducted only in mobility environment.

Our work is mainly inspired by the TCP-Probing proposed in Ref.[11]. In this scheme, a “Probe Cycle” consists of a structured exchange of “probe” segments between the sender and receiver that monitor network conditions. The sender enters a probe cycle when a segment is detected lost either by a timeout event, or by three duplicate ACKs. At the end of probe cycle, TCP-Probing uses a simple algorithm based on the measured RTTs (Round Trip Time) of the last two consecutive probe packets to determine the TCP restart mechanism (Immediate Recovery or Slow Start). Analysis and simulation results show that TCP-Probing outperforms traditional TCP in channel error conditions.

3 TCP Freeze-Probing Enhancements

In this section, the proposed TCP Freeze-Probing enhancement is presented. The main idea of our enhancement is based on the fact that, generally speaking, there are four reasons for packet losses, congestion, channel error, route change and link breakage, which can also be classified into two classes, short-term recoverable network anomalies such as transient channel error and fast route change, and long-term network anomalies such as severe congestion, long burst of channel errors and link breakage (disconnection). Instead of determining the exact reason of packet loss, one can enable TCP performing properly according to the two classes defined above. The state transition diagram of TCP Freeze-Probing is given in Fig.1. Assuming that the sequence number of the last acknowledged packet of a TCP sender is N, when the TCP sender receives three duplicate ACK (DUPACK) of this packet or the TCP retransmission timeout (RTO) expires, it means that packet N+1 is lost due to network anomalies. TCP then resends packet N+1 and enters into frozen state, at the same time a “frozen timer” is set (in our simulation, the value of frozen timer is 2 seconds). If short-term recoverable network anomalies are encountered, network is expected to recover before the frozen timer expires. So TCP can receive the ACK of packet N+1 and then resumes the normal transmission. If however, long-term network anomalies are encountered, the ACK of packet N+1 cannot be received before frozen timer

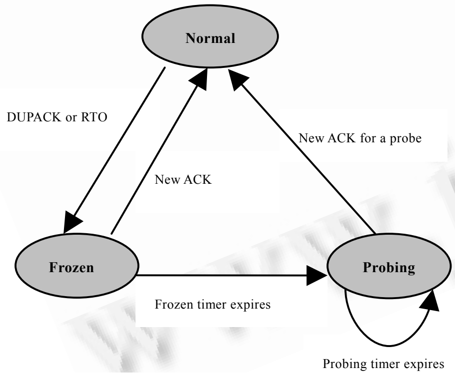

Fig.1 State transition diagram of TCP Freeze-Probing

expires. After frozen timer expires, TCP enters into the probing state, a probe packet with size 40 bytes will be sent into network and a probing timer is triggered at the same time (in our simulation, the value of probing timer is 2 seconds). Probe packet will be resent if probing timer expires. Once the ACK of probe packet is received, TCP enters into the normal state. We reduce the congestion window to its half size when TCP resumes the normal transmission. Besides, it is shown by simulation that the optimal value of the frozen timer and probing timer depends on the network environment.

TCP Freeze-Probing is an end node detection approach, while TCP ELFN is a network detection approach. Their advantages and disadvantages have been discussed in the previous section. We will give the performance comparisons based on simulation results in Section 5.

The main difference between TCP Freeze-Probing and TCP Probing is that we use a “Frozen” state when network anomalies are detected instead of directly sending probe packets. The advantage of our approach lies in: firstly in short-term recoverable network anomalies, TCP Free-Probing can immediately restart the TCP instead of entering the time-wasting probing cycle; while TCP-Probing needs to issue at least two probe packets to restart TCP; secondly since we use a “Frozen” state to cope with the short-term network anomalies, we don’t need to measure the RTTs of two consecutive probe packets to determine the network condition at the end of the probe cycle.

4 Throughput Modeling and Validation

In order to analyze the proposed enhanced TCP Freeze-Probing scheme, a throughput model is developed in this section and will be validated through simulation in the next section. Figures 2 and 3 represent the original and modified versions of window size evolution of our model, where $W_{\max}$ is the maximum congestion window size, P_{short} is the period of short-term recoverable network anomalies, P_{long} is the period of long-term network anomalies, T_0 represents the time of frozen timer and probing timer (in our approach, these two values both equal 2 seconds). Figure 2 gives the original evolution of window size of TCP Freeze-Probing, in which P_{short} and P_{long} compose P_{cycle}: period of one cycle. For the convenience of modeling and calculation, we modify this original version to the modified version as shown in Fig.3. It is obvious that these two versions are identical for calculating the throughput.

Now we give our definitions of *throughput* and *goodput* in this paper. The throughput of a flow is defined as:

$$Throughput = \frac{TotalBytes}{Time}$$

which represents the bandwidth taken up by a flow, but not always related to the efficiency of the running protocol. As opposed to the throughput mentioned above, the so-called *goodput* gives the actual transmission rate perceived on the receiver’s application layer, which is defined as:

$$Goodput = \frac{DataBytes}{Time}$$

where, *DataBytes* is the number of effective bytes received by the receiver. For the convenience of calculation, we use *TotalBytes* instead of throughput as the final result. From Fig.3, we can get the following formula:

$$TotalBytes = THR_{short} * (T_{total} - nT_0),$$

THR_{short} is the throughput of P_{short}, n is the number of T_0 occuring in P_{long}, T_{total} is the total time of connection. In fact, according to Ref.[12], THR_{short} can be calculated as follows:

$$\min\left(\frac{W_{\max}}{RTT}, \frac{1}{RTT}\sqrt{\frac{3}{2N_{cycle}}}\right) * PacketSize.$$

RTT is Round Trip Time. N_{cycle} is the number of P_{cycle} during the connection. *PacketSize* is the number of bytes per packet. Due to the limitation of space, we don't give the development process of this formula. Please refer to Ref.[12] for more details. Then *TotalBytes* can be calculated as follows:

$$\min\left(\frac{W_{\max}}{RTT},\frac{1}{RTT}\sqrt{\frac{3}{2N_{cycle}}}\right)*(T_{total}-nT_0)*PacketSize\ .$$

In next section, in each simulation, the average *RTT*, N_{cycle} and n will be measured, so one can calculate *TotalBytes* of each simulation. Comparison will then be made between this value and the result of *TotalBytes* obtained through simulation. We define:

$$Ratio=\frac{TotalBytes\text{ calculated from model}}{TotalBytes\text{ obtained from simulation}}\ .$$

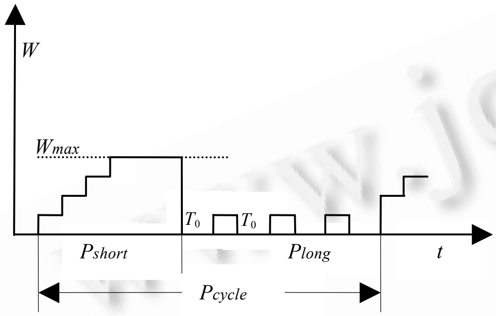

Fig.2 Evolution of window size when limited by $W_{\max}$

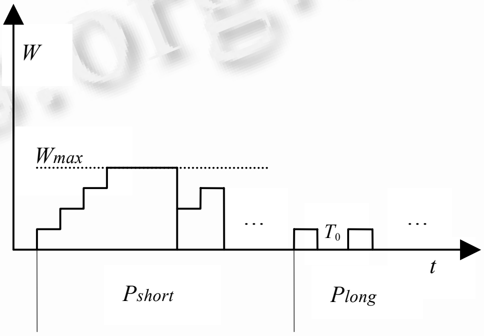

Fig.3 Evolution of window size when limited by $W_{\max}$ (modified version)

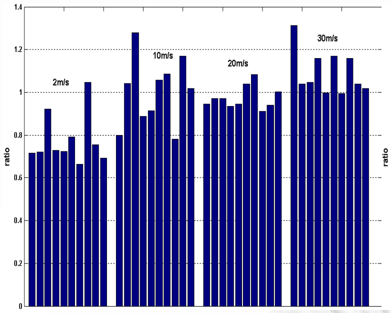

Fig.4 Ratio in mobility environment

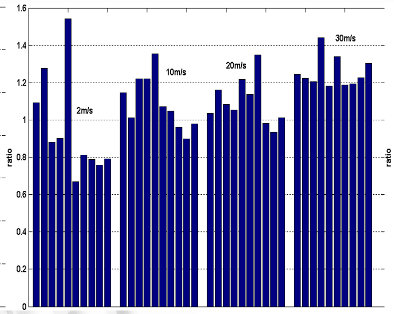

Fig.5 Ratio in congestion environment

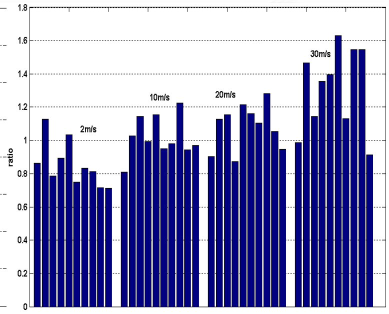

Fig.6 ratio in channel error environment

The proposed model is validated under three conditions, mobility environment, mobility environment with congestion and mobility environment with channel errors, as shown in Figs.4−6 for different mobile speeds, that is, 2m/s, 10m/s, 20m/s and 30m/s.

From Figs.4−6, it can be observed that the calculated value from our model is well close to the simulated one. It can also be noted that the ratio increases with the increase of the mobility speeds. The proposed model behaves very well in all conditions with mobile speeds 10m/s and 20m/s.

5 Simulation and Discussions

Our simulation tool used is NS2[13] (version ns2.1b9) with the following configuration, 30 mobile nodes move around randomly in a 400m x 800m rectangular area. Each mobile node has a transmission range of 250 meters, and 20 scenarios for each set of simulation are used, with each simulation lasting 300 seconds. The average value of the simulation results of 20 scenarios will be used for our purpose. IEEE 802.11 MAC layer protocol is used and DSR is used as network routing protocol. For comparison, TCP Reno[14], TCP ELFN and TCP Freeze-Probing are used as the transmission protocols, where TCP maximum window is set to 8 and packet size is 1460 bytes.

In DSR, a node responds to the route request if it is the destination or if it has a cached route to the destination. According to Refs.[1,10], TCP performance is improved by disabling cached routes, as caching helps in propagation of the stale route information. So the responding to route requests is disabled based on cached routes.

A pair of nodes is chosen as the TCP sender and TCP receiver among 30 nodes. In order to assure the average hop number between sender and receiver to be of value 6, we fix the sender and receiver at the left-lower corner and right-upper corner in the rectangular area respectively (similar to the simulation topology in Ref.[3]). Here, the *goodput* is used as the main performance metric, and by adding congestion and channel error in simulation, the performance of TCP Freeze-Probing with TCP Reno and TCP ELFN is compared and analyzed.

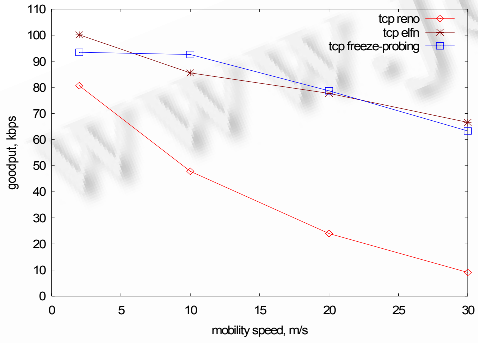

Fig.7 Performance comparison
in different mobility speeds

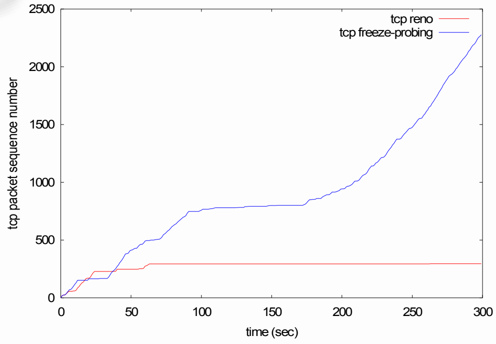

Fig.8 Performance comparison
of one specific scenario

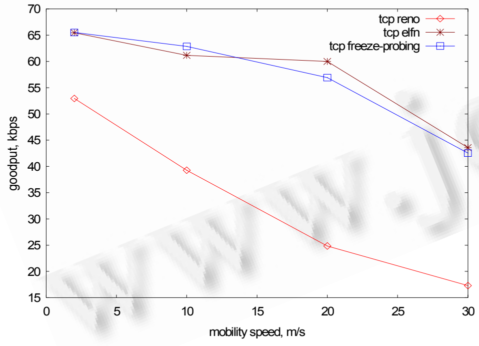

Fig.9 Performance comparison
in congestion environment

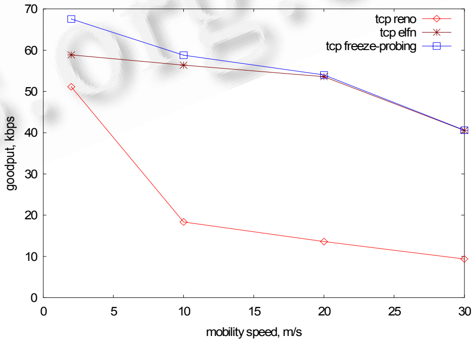

Fig.10 Performance comparison
in channel error environment

From Fig.7, it can be seen that TCP ELFN and TCP Freeze-Probing have better performance than TCP Reno in different mobility speeds. With the increase of mobility speed, the performance gap between TCP Reno and TCP Freeze-Probing increases from 16% to 600%. The reason is that in mobility environment packet losses are due to route change and link breakage but TCP Reno always considers the reason as congestion and incorrectly invokes the congestion control mechanisms. We can refer to Fig.8 to further understand the reason. It clearly shows that TCP Reno suffers in mobility scenario. The *goodput* of TCP Reno is about 13% of that achieved by TCP Freeze-Probing. When packet loss happens, TCP Reno doubles the retransmission timeout even the packet loss is due to short-term recoverable network anomalies. In this scenario, as TCP Reno experiences timeouts repeatedly, the time wasted by the retransmission timeout increases and TCP Reno could hardly recover from network anomalies. Consequently, TCP performance is surprisingly low. On the other side, TCP ELFN and TCP Freeze-Probing use the freezing mechanism to prevent the TCP sender from incorrectly enlarging the retransmission timer and ensure the efficient transmission. From the simulation results, it can also be observed that the TCP ELFN and TCP Freeze-Probing have nearly the same performance considering the simulation environment and simulation methods.

To make a congestion environment in the network, two UDP flows between intermediate nodes in each scenario are added. These UDP flows run during the periods of [50,250] and [100,200] respectively. The sending rate of each UDP flow is 200kbps. From Fig.9, we observe that TCP ELFN and TCP Freeze-Probing have better performance than TCP Reno in different mobility speeds. In congestion condition, the performance gap between TCP Freeze-Probing and TCP Reno increases from 32% to 160%, which is less than that of mobility environment without congestion. The reason is that TCP Reno's congestion control mechanism works in this condition. From the simulation results, we also observe that TCP ELFN and TCP Freeze-Probing have nearly the same performance considering the simulation environment and simulation methods.

In fact, IEEE802.11 protocol in MAC layer can effectively handle the random channel error. Therefore, severe channel condition is added in simulation to illustrate the channel error effect to transmission protocol, where the channel error rate is set to be 10% in our simulation scenarios. Simulation results in Fig.10 show that TCP ELFN and TCP Freeze-Probing have better performance than TCP Reno. The performance gap between TCP Freeze-Probing and TCP Reno increases from 20% to 300%. The reason is as described earlier: TCP Reno cannot distinguish packet losses due to congestion from packet losses due to channel errors. Simulation results also show that TCP Freeze-Probing has better performance than TCP ELFN. In severe channel error environment, ELFN message can also be lost due to channel errors, which makes TCP ELFN's feedback mechanism less efficient than the freeze-probing mechanism of TCP Freeze-Probing.

6 Conclusions

In this paper, a TCP enhancement in MANET named TCP Freeze-Probing is presented, and it is shown that the TCP performance can be improved significantly in MANET. A simple throughput model for TCP Freeze-Probing is also proposed and validated through simulation. By dividing the network anomalies into two classes, short-term recoverable network anomalies and long-term network anomalies, and by a simple yet efficient algorithm, better performance compared with TCP Reno is achieved. TCP Freeze-Probing is even better than TCP ELFN in noisy channel environment. In addition, since TCP Freeze-Probing belongs to end node detection approaches, there is no need for the cooperation of intermediate nodes in networks.

References:

[1] Holland G, Vaidya NH. Analysis of TCP performance over mobile ad hoc networks. Wireless Networks, 2002,8(2-3):275−288.

[2] Oliveira R, Braun T. TCP in wireless ad hoc networks. Technical Report, IAM-02-003, 2002.

[3] Fu ZH, Meng XQ, Lu SW. How bad TCP can perform in wireless ad hoc network. In: Proc. of the 7th Int'l Symp.on Computers and Communications (ISCC2002). 2002. 298−303.

[4] Fu ZH, Greenstein B, Meng XQ, Lu SW. Design and implementation of a TCP-friendly transport protocol for ad hoc networks. In: Proc. of the 10th IEEE Int’l Conf. on Network Protocols (ICNP’02). 2002. 216−225.

[5] Chandran K, Raghunathan S, Venkatesan S, Prakash R. A feeback based scheme for improving TCP performance in ad-hoc wireless networks. In: Proc. of the Int’l Conf. on Distributed Computing Systems ICDCS’98. 1997. 472−479.

[6] Liu J, Singh S. ATCP: TCP for mobile ad hoc networks. IEEE Journal on Selected Areas in Communications, 2001,19(7): 1300−1315.

[7] Wang F, Zhang Y. Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response. In: Proc. of ACM Mobihoc’02. 2002. 217−255.

[8] Dyer TD, Boppana RV. A comparison of TCP performance over three routing protocols for mobile ad hoc networks. In: Proc. of ACM MobiHoc2001. 2001. 56−66.

[9] Johnson DB, Maltz DA, Broch J. DSR: The dynamic source routing protocol for multiple wireless ad hoc networks. In: Perkins C, ed, Ad Hoc Networking. Addison-Wesley, 2001. 139−172.

[10] Monks J, Sinha P, Bharghavan V. Limitation of TCP-ELFN for ad hoc networks. In: Workshop on Mobile and Multimedia Communication. 2000.

[11] Tsaoussidis V, Badr H. TCP-Probing: Towards an error control schema with energy and throughput performance gains. In: Proc. of the 8th IEEE Conf. on Network Protocols (ICNP2000). 2000. 12−21.

[12] Padhye J, Firoiu V, Towsley D, Kurose J. Modeling TCP reno performance: A simple model and its empirical validation. IEEE/ACM Trans. Networking, 2000,8(2):133−145.

[13] The NS Manual. http://www.isi.edu/nsnam/ns/ns-documentation.html

[14] Stevens WR. TCP slow start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. IETF, RFC2001, 1997.

[15] Sundaresan K, Anatharaman V, Hsieh H, Sivahkumr R. ATP: A reliable transport protocol for ad-hoc networks. In: Proc. of the ACM MobiHoc2003. 2003. 64−75.

[16] Yi Y, Shakkottai S. A hop-by-hop congestion control over a wireless multi-hop network. In: Proc. of the IEEE INFOCOM2004. 2004.