Performance Enhancement and Utility Maximization for TCP in Wireless Mesh Networks

Hongquan Liu, Jiong Chen, Siming Song, and Yuantao Gu[*]

Received July 15, 2011; accepted Dec. 27, 2011.

This article appears in *China Communications*, 2012, 9(4):35-44.

Abstract

In Wireless Mesh Networks (WMNs), the performance of conventional TCP significantly deteriorates due to the unreliable wireless channel. To enhance TCP perform- ance in WMNs, TCP/LT is proposed in this paper. It adopts fountain codes into packet reorganization in the protocol stack of mesh gateways and mesh clients. Furthermore, it is compatible with conventional TCP. Regarded as a Performance Enhancement Proxies, a mesh gateway buffers TCP packets into several blocks. It simultaneously processes them using fountain encoders and then sends them to mesh clients. Besides improving the throughput of a unitary TCP flow, the entire network utility maximization can also be ensured by adjusting the scale of coding blocks for each TCP flow adaptively. Simulations show that TCP/LT presents high throughput gains over single TCP in lossy links of WMNs while preserving the fairness for multiple TCPs. As losses increase, the transmission delay of TCP/LT experiences a slow linear growth in contrast to the exponential growth of TCP.

Keywords: TCP; TCP/LT; fountain codes; wireless mesh networks; network utility maximization

1 Introduction

Wireless Mesh Networks (WMNs) are deployed as a high-performance and low-cost technique to provide users with community or city-wide Internet access [1]. In WMNs, local access points and wireless mesh routers make up a backbone network. They forward data packets between mobile clients and the Internet.

To provide a reliable end-to-end connected- oriented data delivery service, TCP is widely used as a transport layer protocol in wired and wireless networks. As the conventional TCP was designed specifically for wired networks, where the packet loss is mainly induced by congestion, when being applied in WMN, the performance of conventional TCP is significantly affected. One of the well-known reasons is that the conventional TCP fails to

---

[*] This work was partially supported by National Natural Science Foundation of China (NSFC 60872087 and NSFC U0835003). The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. The corresponding author of this paper is Yuantao Gu (e-mail: gyt@tsinghua.edu.cn).

differentiate congestion and non-congestion losses. As a result, TCP's throughput quickly drops when non-congestion losses occur. But the error-prone wireless medium of WMNs increases the occurrence of random losses. In multi-hop WMNs, packet collisions caused by hidden terminals and link failure due to mobility in mesh clients result in added non-congestion packet losses. In the recent studies of IEEE 802.11b based wireless mesh networks [2], it has been reported that packet loss rate is as high as 50%, which typically translates to low TCP throughput and high delay. As TCP is critically dependent on ACKnowledgement character (ACK), its performance can be severely impacted by both the loss of ACKs and network asymmetry, which are common cases in WMNs. To overcome the deterioration of TCP performance in WMNs, it is needed to modify the conventional TCP to cope with the increased packet loss.

Another important issues arising with the modification to conventional TCP is the compatibility. As for WMNs, Mesh clients in WMNs usually communicate with other network clients outside mesh networks (e.g., in the Internet), as illustrated in Fig. 1. As a result, there may exist both wireless links and wired ones from a sender to a receiver. The communication environment is different from other multi-hop networks, such as mobile ad hoc networks. To the best of our knowledge, only a few reliable transport protocols have been introduced specifically for WMNs. Therefore, it is a key issue that the enhanced TCP should be compatible with existing TCP in WMNs.

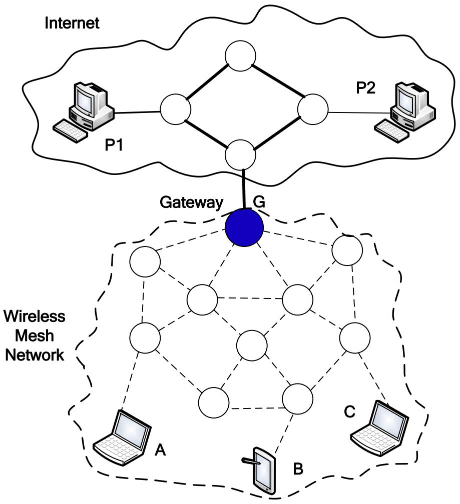

Fig.1 a demonstrate of WMNs

To enhance the performance of the conventional TCP in WMNs, an efficient transport protocol named TCP/LT is proposed in this paper using LT codes [3], a class of fountain codes. It can handle the high packet loss ratio and reduce the number of ACKs, which greatly enhances the performance of TCP in WMNs. Moreover, TCP/LT is grounded on a Network

Utility Maximization (NUM) frame. Different from the conventional TCP retransmission scheme, it introduces redundancy to guarantee reliability and adapti- vely adjusts the number of encoding blocks to achieve the NUM of WMNs. With the utility maximization approach, transmission rates can be allocated optimally and the achievable end to end throughputs can be maximized.

The rest of the paper is organized as follows. Section II presents related work. Section III first introduces the system model. Then it describes the NUM Formulations in section IV. The proposed approach and its implementation details are presented in section V. Performance results (ns-2 simulations) are given in section VI. Finally, section VII is the conclusion.

2 Related works

One fundamental mechanism to enhance TCP performance in wireless networks is to employ a TCP proxy called Performance Enhancing Proxy (PEP). PEP transparently splits a TCP session into multiple sub-sessions and forwards packets via them [3]. M.Ivanovich [4] proposed a novel TCP PEP called TRL-PEP which has a better performance than standard TCP and TCP Westwood [5]. However, its performance still deteriorates sharply when packet loss rate is greater than 5%. T.Shikama [6] investigates a TCP proxy that splits a TCP session into two parts consisting of a wireless link and a wired one, which is similar with WMNs environment. Whereas the simulation results show that the TCP throughput reduces to 20% while the packet loss rate is 10%, which is obviously unsatisfactory.

Fountain codes, which are a subclass of erasure codes, were introduced in recent years [7]–[11]. Unlike conventional codes, fountain codes are rateless codes. A sender transmits encoded packets with a specific rate according to the error ratio of each specific channel. A receiver can obtain the original information as soon as it has accepted enough packets. Fountain codes transmit packets with blocks. As a result, they only require few feedbacks to guarantee reliable transmission.

Fountain Codes have been suggested for One-To-Many TCP and steaming video applications[12][13]. Those proposed mechani- sms indeed enhanced the performance of their specific systems; despite they are not common solutions for end to end reliable transmission. A few Fountain Codes based Transport (FCT) protocols have been proposed [14], [15]. These protocols are alternative to TCP. They are neither transparent to the applications nor compatible with other protocols in the existing networks. Therefore they are unfit for the WMNs.

These approaches based on fountain codes above are proposed especially for in an 802.11 WLAN Cell and can’t work well in multi-hop WMNs. Sundararajan *et al.* proposed an approach (i.e., TCP/NC) to incorporate network coding which is also a rateless code as the same as fountain codes with TCP in multi-hop wireless networks [16][17]. In TCP/NC, redundant packets are injected into the network to mask losses from TCP. TCP/NC improves the throughput of TCP when packet losses occur. But the achieved throughput gain drops to 2.7% with an incorrect redundancy parameter. In addition, TCP/NC may also increase the end-to-end delay.

TCP/LT proposed in this paper is a TCP PEP architecture based on fountain codes which can greatly enhance TCP performance in multi-hop WMNs, and moreover, it is compatible with the current TCP variants.

3 System model

Consider the TCP traffic between a multi-hop WMN and the Internet.

1) *Networking*: As illustrated in Fig.1, mesh clients, mesh routers and mesh gateways form a single-radio WMN. More specifically, they are based on IEEE 802.11b technology. There are contentions and interferences among them.
2) *Wireless Transmission:* There are multiple wireless hops between a mesh gateway (e.g., G in Fig.1) and a mesh client (e.g., A in Fig.1). A node accesses channels following the IEEE 802.11 MAC protocol.
3) *TCP Flows*: Let S be the set of unicast TCP flows between senders and receivers in the network. A sender (a receiver) is usually located in the Internet, and a receiver (a sender) is located in WMNs. Each flow $s \in S$ has a rate x_s and a utility function $U_s(x_s)$. As well known, by the law of diminishing marginal utility, let $U_s(x_s)$ be $log x_s$, which is a concave function of x_s, and a proportionally fair resource allocation can be obtained through it.
4) *Achievable Path Capacity*: Let G be the set of gateways in WMNs. The routing path of a certain TCP flow between a gateway and a mesh client, denoted as r, has an end-to-end capacity C_r in the Transport Layer and a packet loss rate P_r. A gateway $g \in G$ has a capacity C_g.

Due to the contention and the interference among neighboring nodes in multi-hop WMNs, the achievable capacity in the Transport Layer of a certain path depends on network topology and specific traffic flows.

For IEEE 802.11-based wireless multi-hop networks, the achievable capacities of disjoint flow paths can be calculated based on previous study results [19] [20] [21] [22]. For an h hop chain-topology, the maximum number of simultaneous transmissions is upper bounded by $h/4$ [21]. As a result, in an h ($h \leq 3$) hop chain, the end-to-end achievable capacity in the Transport Layer of a path can be calculated as follows.

$$C_{Path} = \frac{L_{Data}}{h * T_{pkt}}, \tag{1}$$

where T_{pkt} is cycle time of transmitting a packet, and

$$T_{pkt} = (T_{Data} + T_{SIFS}) + T_{ACK} + (T_{RTS} + T_{SIFS}) + (T_{CTS} + T_{SIFS}) + \overline{CW} + T_{DIFS}. \quad (2)$$

The symbols presented above are fixed in IEEE 802.11-based wireless multi-hop networks. And table 1 lists the meanings as well as the typical values of these symbols.

In a 3-hop chain, for an example, the end-to-end capacity is

$$C_{Path} = \frac{L_{Data}}{3 * T_{pkt}} = 273.056 Kbps.$$

Table I. Symbols used in an 802.11 network

Symbol	Meaning & Value
L_{Data}	Length of data packet (1000B)
T_{Data}	Time required to transmit a packet (T_P+T_{PHY}+ (L_{H_MAC}+L_{Data})/ data_rate)
T_{SIFS}	SIFS time (10μs)
T_{ACK}	Time required to transmit an ACK (T_P+T_{PHY}+ L_{ACK}/ *data_rate*)
T_{RTS}	Time required to transmit RTS (L_{RTS} / *basic_rate*)
T_{CTS}	Time required to transmit CTS (L_{CTS} / *basic_rate*)
$\overline{CW}$	Time required for back off *(CWmin* Tslot /2)*
T_{DIFS}	DIFS time (50μs)
T_P	Time required to transmit preamble (144μs)
T_{PHY}	Time required to transmit physical layer header (48μs)
L_{H_MAC}	Length of MAC header (28B)
L_{ACK}	Length of ACK (14B)
L_{RTS}	Length of RTS (44B)
L_{CTS}	Length of CTS (38B)
data_rate	Speed of transmitting MAC packets (1Mbps)
basic_rate	Speed of transmitting control frames (1Mbps)
CW_{min}	Size of the initial contention window (31)
T_{slot}	Slot Time (20μs)

4 Network utility maximization formulations

In WMNs, each TCP flow passing through a mesh gateway (e.g., G in Fig.1) has a rate x_s, TCP/LT proposed in this paper aims to solve the following NUM problem,

$$\text{Max} \sum_{s \in S} U_s(x_s),$$

$$\text{s.t.} \sum_{s \in S} x_s \le C_g, \tag{3}$$

$$x_s \ge 0, \forall s \in S. \tag{4}$$

Since $U(x) = \log x$, and $\log x \to -\infty$ as $x \to 0$, so the constraint (4) is not active.

In infrastructure/backbone WMNs [1], the mesh gateways and mesh routers have minimal mobility, hence the achievable capacity of each hop of them can be obtained when a WMN is set up. As a result, a gateway can calculate the achievable capacity of a certain path between it and a mesh client. If two flow paths have a joint, the two paths have a joint capacity C_r which can be calculate by their gateway. For single-radio WMN, the paths through a gateway have a joint capacity C_g all together. C_g usually depends to the longest path, and it can be calculated by the gateway, as mentioned before (in section 2).

Let λ_g be the *Lagrange multiplier* corresponding to the capacity constraints on a gateway g, then, the *Lagrangian* is given by

$$L(x, \lambda) = \sum_{s \in S} U_s(x_s) - \sum_{g \in G} \lambda_g \left(\sum_{s \in S_g} x_s - C_g\right), \tag{5}$$

where x is the vector of rates of all flows, λ is the vector of *Lagrange multipliers*, S is the set of unicast TCP flows, G is the set of gateways in WMNs, x_s is the rate of flow s belonging S, S_g is the set of TCP flows passing through a gateway g. TCP flows passing through different gateways are mutually independent, then Eq. (5) can be expressed as

$$L(x, \lambda) = \sum_{g \in G} \left(\sum_{s \in S_g} U_s(x_s) - \lambda_g \left(\sum_{s \in S_g} x_s - C_g\right)\right). \tag{6}$$

If all the flows have a same utility function $\log x_s$, then with the Karush-Kuhn-Tucker (KKT) condition, the solutions (x_s) to the NUM formulation above can be derived in polynomial time, and x_s is

$$x_s = \frac{C_g}{|S_g|}, \tag{7}$$

where $|S_g|$ is the number of TCP flows passing through a gateway g. Eq. (7) presents that for a specified gateway, the joint capacity C_g is equally allocated for each TCP flows. However the joint capacity may be different for varied traffic patterns. Moreover, the utility function for each TCP may also vary according to its importance. Therefore in order to achieve the network utility maximization, each rate x_s should be adjusted.

5 The Proposed Approach

5.1 The Proposed Approach

The lossy links and mobility of mesh clients in multi-hop WMNs lead to either high loss rate or out-of-order of ACKs. As a result, the throughput of conventional TCP is sharply degraded. TCP/LT aims to enhance TCP performance in WMNs with the NUM formulation as mentioned above. And mean- while it should be compatible with the conventional TCP. As illustrated in Fig.2, TCP/LT introduces a fountain codes layer between TCP and Internet Layer in the protocol stack of a mesh client. Meanwhile it also adds one between Internet Layer and Link Layer in the protocol stack of a mesh gateway. It has minimal changes to the protocol stack. The exact operations of these modules are described in the following text.

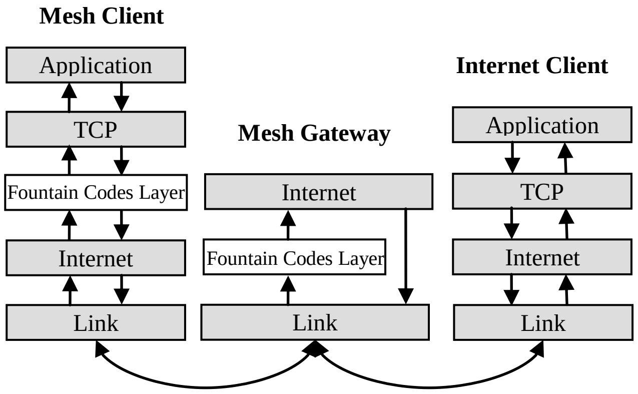

Fig.2 New fountain codes layer in the protocol

A mesh client module and a mesh gateway module act differently associated with whether the mesh client is a sender or a receiver, as illustrated below.

1) Scenario I: a mesh client is a receiver, while an internet client is a sender.

This scenario denotes a mesh client downloads data from an internet client. The added modules work as follow.

Mesh gateway side

As seen in Fig.2, the fountain codes layer module of a mesh gateway accepts upstream packets via its Link Layer and checks whether they are from a TCP application (expressed as TCP flow s) of an Internet client. If the answer is positive, the packets are buffered into an encoding cache until they are acknowledged by a mesh client receiver; otherwise they are directly forwarded to the Internet Layer. Then the module divides the buffered packets into several blocks and independently encodes each block into $R_s *L$ LT packets, where L is the block size, R_s is the redundancy parameter related to the packet loss rate of this TCP flow's path. There is a trade-off between transmission delay and code efficiency while setting the block size. The transmission delay is shorter with a smaller block size. On the contrary, the percentage of overhead is higher with that [18]. Let N_s be the number of blocks, then the total number of $L*N_s$ is associated with the NUM formulation described before. To be more specific, the gateway solves the NUM formulation with the method in section 3, and it gives a x_s to this TCP flow *s,* which is the upper bound of flow s' transmission rate. Let P_s be the path's packet loss rate, thus R can be obtained by equation $R_s = 1/(1-P_s)$. Assuming that the average transmission time of each block is T_s, then, flow s' rate:

$$x_s = \frac{8L_{Data}RN_sL}{T_s} = \frac{8L_{Data}N_sL}{(1-P_s)T_s} \le x_s \ , \tag{8}$$

where P_s and T_s can be estimated by the way that Round-Trip Time (RTT) in TCP does. If P_{s_this} and T_{s_this} are new observation values of P_s and T_s, then P_{s_new} and T_{s_new} can be estimated by

$$P_{s\_new} = \alpha\, P_{s\_old} + (1-\alpha)\, P_{s\_this} \ , \tag{9}$$

$$T_{s\_new} = \alpha\, T_{s\_old} + (1-\alpha)\, T_{s\_this} \ , \tag{10}$$

where, α is a smoothing factor, for experiences and simplicity, α can be taken as 7/8.

According to inequation (8),

$$N_s \le N_{s-MAX} = \left\lfloor \frac{x_s(1-P_s)T_s}{8L_{Data}L} \right\rfloor . \tag{11}$$

Thus, for each TCP flow *s*, there is a maximum of N_s, and then the fountain codes layer module can generate N_{s-MAX} blocks of LT packets simultaneously to achieve the NUM.

The added module then formats the LT packets into TCP packets and forwards them up to the Internet Layer. At the last stage, the Internet Layer sends them to the mesh client destination. If the encoded LT packets are successfully decoded at the receiver, the gateway module will receive an LT-ACK, and then the buffered TCP packets are released and an ACK is sent to the TCP sender in the Internet.

Mesh client side

The mesh client side module accepts upstr- eam LT packets, and buffers them into a decoding cache. For every received LT packet, the module tries to decode it together with the received LT packets of this block before. If a block of encoded LT packets are successfully decoded, the module forwards them up to TCP and generates a LT-ACK to the gateway module with the received packets number of this encoded block.

The received packets number of a certain block at a LT decoder is delivered to the gateway via a LT-ACK, and with this informa- tion as well as the sent packets number of that certain block, the gateway module can estimate the path packet loss rate P_s.

2) Scenario II: a mesh client is a sender, while an internet client is a receiver.

This scenario denotes a mesh client uploads data to an Internet client. The added modules work as follow.

Mesh client side

The mesh client module works almost the same as the gateway module in Scenario I except that the mesh client module receives packets from TCP and sends encoded LT packets to Internet Layer. And also x_s is assigned by the gateway module.

Mesh gateway side

Again the gateway module operates almost the same as the mesh client module in Scenario I. Only the difference is that it receives encoded packets from Link Layer, afterwards it sends decoded TCP packets to the Internet client via Internet Layer.

Above two scenarios coexist with each other in real-world WMNs.

5.2 Implementation

A practical implementation of TCP/LT is proposed following the NUM formulation structure. Mesh clients and mesh gateways operate differently, and their algorithms are separately described using pseudo code. This specification assumes a one-way TCP flow. Table 1 lists the meanings of symbols used in Algorithm 1&2.

Table II. Symbols used in Algorithm 1&2

Symbol	Meaning
Src	Source Address of Packet *Ps*
Dst	Destination Address of Packet *Ps*
BRTTs	Block Round Trip Time of Flow *s*
Seq	*Ps*’ Sequence Number
Bs	Encoding Buffer of Flow *s*
EBs	Num. of *s*’ Encoded Packets
Ns	Num. of *s*’ Encoded Blocks
Ns-sent(i)	Num. of Block *i*’s Sent Packets
Ns-recv(i)	Num.of Block *i*’s received Packets
Ds	Decoding Buffer of Flow *s*

1) Operation of mesh gateways

Algorithm 1

```
A gateway module processes a packet $P_{s.}$
1) If Src is in the Internet & Ps' Type =TCP
    a) If Seq = 0, calculate and save a new $x_s$ , forward Ps, return.
    b) Insert Ps to Bs,
        i) If $|Bs|-EBs \geq L$ & $Ns+1 \leq N_{s\text{-}MAX}$
        create $\lceil L*R \rceil$ LT packets and forward them
```

```
2) If Src is in the Internet &Ps’ Type = ACK
  a) If Seq = 0, add $x_s$ to Ps and forward Ps
3) If Src is in the WMN
  a) If Ps’ Type = TCP & Seq = 0, Calculate and save a new $x_s$ , forward Ps
  b) If Ps’ Type = ACK & Seq = 0, forward Ps
  c) If Ps’ Type = LT,
    i) Insert Ps to Ds, decode it.
    ii) If successfully decode this block, generate a LT-ACK Pa, cancel Receiving Timer(i), return.
    iii) Start Receiving Timer(i)
  d) If Ps’Type = LT-ACK
    i) If successfully decoded, update BRTTs, recalculate $N_{s\text{-}MAX}$ by equation (11), return.
    ii) Generate $\lceil L*R - Ns\text{-}recv(i) \rceil$ LT packets and forward them.
Receiving Timer (i) expires: Generate an LT-ACK, add Ns-recv(i) to it, then forward it, return.
```

2) Operation of mesh clients

```
Algorithm 2
A mesh client module processes a packet $P_s$.
1) If Src is himself & Ps’ Type = TCP
  a) If Seq = 0, forward Ps, return.
  b) Insert Ps to Bs
    i) if $|Bs|-EBs \geq L$ && $Ns+1 \leq N_{s\text{-}MAX}$, create $\lceil L*R \rceil$ LT packets and forward them.
2) If Src is himself & Ps’ Type = ACK
  a) If Seq = 0, forward Ps, return.
3) If Src is in the Internet
  a) If Ps’ Type = TCP && Seq = 0, forward Ps up to the TCP session, return.
  b) If Ps’ Type = LT
    i) Insert Ps to Ds, decode it.
    ii) If successfully decode this block, generate a LT-ACK Pa, forward it, cancel Receiving Timer(i), return
    iii) Start Receiving Timer(i)
```

```
    c) If Ps' Type = ACK && Seq = 0, derive $x_s$ from Ps, calculate the $N_{s\text{-}MAX}$ by equation (11), forward Ps up to the TCP session, return
    d) If Ps'Type = LT-ACK
        i) If successfully decoded, update BRTTs, recalculate $N_{s\text{-}MAX}$ by equation (11), return.
        ii) Generate $\lceil L*R - Ns\text{-}recv(i) \rceil$ LT packets and forward them.
Receiving Timer (i) expires: Generate an LT-ACK, add Ns-recv(i) to it, then forward it, return.
```

6 Performance evaluations

The simulation results of TCP/LT are presented in this section. Ns-2.33 [23] is used for these simulations. The simulations are based on TCP-NewReno, which is a widespread and practical variant of TCP.

As illustrated in Fig.3, the topology for simulations is a wired and wireless hybrid network. The bandwidth of wired link between node 1 and node 2 is 2 Mbps, and transmission delay is 50 ms. Node 2 to 5 make up a WMN with a 3-hop chain-topology using IEEE 802.11 technologies. The distance of each wireless link among them is 200 m, and the data rate and the basic rate of wireless links are both 1 Mbps. In the simulator, the effective transmission range is 250 m, and the interfering range is about 550 m. There is no MAC layer retransmission in these simulations. The loss rate of wired link 1-2 is kept at zero; the loss rate of each wireless path is *P* which is varied from 0% to 20%, thus the end-to-end loss rate is $1-(1-P)^3$. The TCP packet size is set to 1000 bytes in the simulator, and the simulation time is 5000s

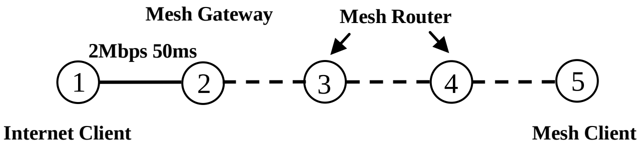

Fig. 3 Simulation topology

6.1 Effects of Block Size

Block size is a key characteristic of TCP/LT. The effects of block size to throughput and transmission delay are discussed in this section. LT codes with small block size require a higher percentage overhead [18], as expected, but they offer the advantage of lower overall delay. To test this effect to TCP/LT, the throughput and the transmission delay are measured for block size *L*=8, 16, and 32. The simulation is set up with a TCP flow from 1 to 5. As seen in Fig. 4, a smaller block size corresponds to a shorter transmission delay. However, due to the added overhead, the throughput of TCP/LT drops sharply as L decreases, as seen in Fig. 5. There is a trade-off between transmission delay and efficiency while choosing the block size. For experiences, L is set to 16.

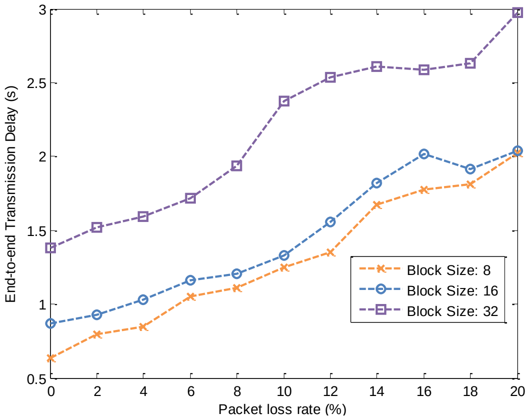

Fig. 4 the effect of block size to delay

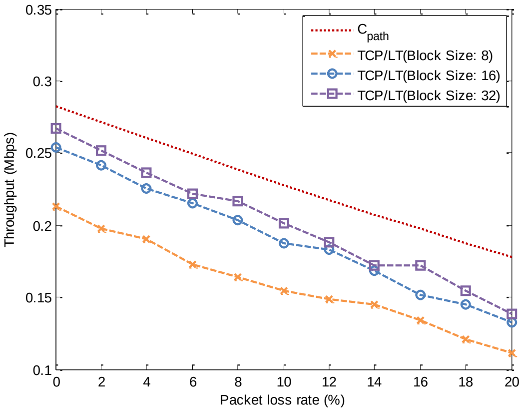

Fig. 5 the effect of block size to throughput

6.2 Throughput of TCP/LT

The variation of throughput versus loss rate for TCP, TCP/NC and TCP/LT is studied. In these simulations, there is a TCP connection from node 1 to node 5.

Fig.6 shows that, TCP’s throughput falls rapidly as losses increase. TCP/NC is very robust to losses and reaches a reasonable throughput with linear reduction.However, TCP/LT outperforms TCP/NC and obtains a higher throughput which is close to the end-to-end capacity of that path. The throughputs of TCP and TCP/NC are far lower than maximum throughput, even with no packet loss, as seen in Fig. 6. And this is caused by the transmission of ACKs. For a 3-hop chain-topology, when there is a link lossrate P, C_{path} can be calculated as follow:

$$C_{path} = \frac{L_{Data}(1-p)^3}{T_{pkt}(1+(1-p)+(1-p)^2)} \tag{12}$$

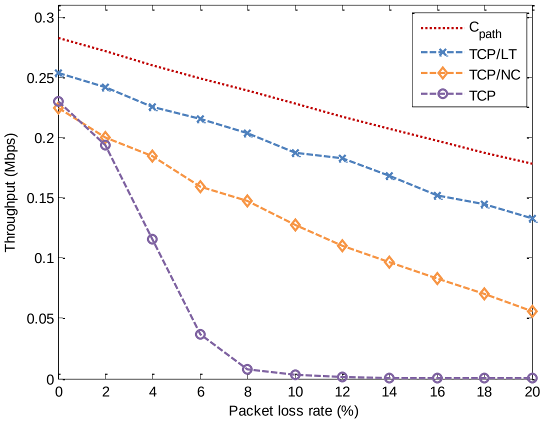

Fig. 6 Throughput versus loss rate for TCP and TCP/LT

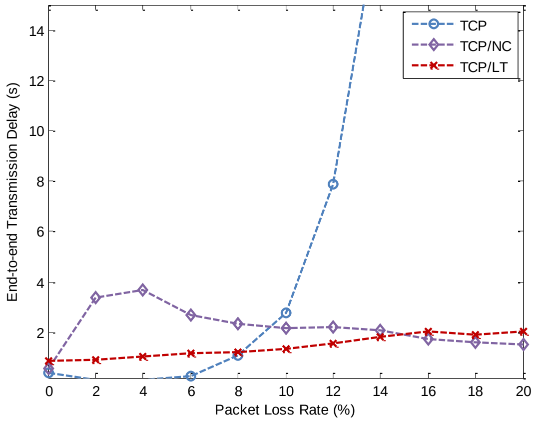

Fig. 7 Transmission delay versus loss rate for TCP and TCP/LT

TCP/LT transmits packets and acknowle- dges them with block. As a result, its average transmission delay is longer than conventional TCP when loss rate is low, as seen in Fig. 7. However as losses increase, the transmission delay of TCP/LT experiences a slow linear growth in contrast to the exponential growth of TCP. In most instances, the delay of TCP/NC is larger than TCP/LT and this is caused by incorporating network coding with TCP in a sliding way, as stated before. The losses of TCP packets and ACKs make conventional TCP retransmit those packets frequently so that TCP has a longer delay than TCP/LT when packet loss rate is higher than 8%. Fig. 7 also shows that the delay of TCP is firstly increased and then decreased as losses increase, and this is due to that TCP has a big congestion window when packet loss rate is low. And as a result, the length of packet queue is long which leads to long delay.

6.3 Fairness of TCP/LT

The capacity of fairness with loss rate for multiple TCP/LT flows is also studied. The simulation scenario has four TCP flows: two TCP flows from 1 to 5; two TCP flows from 5 to

1. Fig. 8 shows that TCP/LT can guarantee a good fairness among both downstream flows and upstream flows based on the NUM formulation.

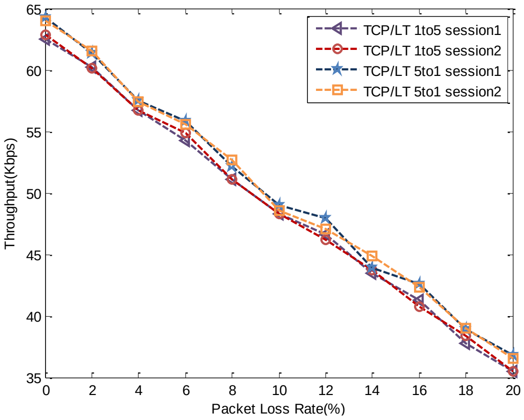

Fig. 8 Throughput versus loss rate with multiple TCP/LT

7 Conclusions

In this study, a new approach TCP/LT was proposed to enhance TCP performance in WMNs by harnessing LT codes, a class of fountain codes. TCP/LT is grounded on a NUM frame with minor changes to the protocol stack. Moreover it is transparent to TCP sessions at Transport Layer. Therefore it can work well with existing TCP variants without specific modification, which facilitates its deployment in an existing WMN. The block size is a key characteristic of TCP/LT, and it can be set to 16 as an empirical value. The simulations show that the proposed changes lead to high throughput gains over TCP in loss links of WMNs. Contrary to the conventional TCP, the transmission delay of TCP/LT experiences a slow linear growth when losses increase, which verify that the proposed approach can be utilized to overcome the deficiency in the WMNs. Another metric of the proposed approach is that the maximum of network utility can be achieved by ensuring the fairness of each TCP flow. This work can be extended by combing other fountain codes to reorganize the TCP packets. The automatic method to tuning the TCP/LT parameters is also in the scope of our interest.

References

[1] AKYILDIZ I F, WANG X, WANG W. Wireless mesh networks: A survey[J]. Computer Networks, 2005, 47(4): 445-487.

[2] AGUAYO D, BICKET J, BISWAS S, et al. Link-level measurements from an 802.11b mesh network[C]// Proc. of ACM SIGCOMM: Aug 30 - Sep 3, 2004, Portland, USA. New York, NY, USA: ACM, 2004: 121-132.

[3] BORDER J, KOJO M, GRINER J, et al. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations[S], IETF, RFC3135, 2001.

[4] IVANOVICH M, BICKERDIKE P, LI J. On TCP Performance Enhancing Proxies in a Wireless Environment[J]. IEEE Communications Magazine, 2008, 46: 76–83.

[5] GERLA M, SANADIDI M Y, REN W, et al. TCP Westwood: Congestion Window Control Using Bandwidth Estimation[C]// Proc. IEEE GLOBECOM ’01: Nov 25-29, 2001, San Antonio, USA. Piscataway, NJ, USA: IEEE, 2001: 1698-1702.

[6] SHIKAMA T. Mitigation of bursty packets by a TCP proxy improving TCP performance in a wired and wireless network[C]// IEEE GLOBECOM Workshops: Dec 6-10, 2010, Miami, USA. Piscataway, NJ, USA: IEEE, 2010: 425-429.

[7] LUBY M. LT codes[C]// Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science: Nov 16-19, 2002, Vancouver, Canada. Piscataway, NJ, USA: IEEE, 2002: 271–280.

[8] BYERS J W, LUBY M, MITZENMACHER W. A digital fountain approach to asynchronous reliable multicast[J]. IEEE J. Select. Areas Commun., 2002, 20:528–1540.

[9] SHOKROLLAHI A. Raptor codes[J]. IEEE Transactions on Information Theory, 2006, 52(6):2551-2567.

[10] MITZENMACHER M. Digital fountains: A survey and look forward[C]// Proc. IEEE Inform. Theory Workshop: Oct 24-29, 2004, San Antonio, USA. Piscataway, NJ, USA: IEEE, 2004: 271–276.

[11] MACKAY D J C. Fountain codes[J]. IEE Proc.-Commun., 2005, 152(6): 1062–1068.

[12] SEJDINOVIC D, PIECHOCKI R, DOUFEXI A, *et al*. Fountain Code Design for Data Multicast with Side Information[J]. Wireless Communications, IEEE Transactions on, 2009, 8(10): 155 – 5165.

[13] HYUNCHUL J, DONGJU L, SANGCHUN H, *et al*. Fountain Code-based Live Video Streaming System over Heterogeneous Wireless Networks[C]// Proc. of Information and Communication Technology Convergence: Nov 17-19, 2010, Jeju Island, Korea. Piscataway, NJ, USA: IEEE, 2010: 1-5.

[14] LOPEZ L, FERNANDEZ A, CHOLVI, V. A Game Theoretic Analysis of Protocols Based on Fountain Codes[C]// Proc. of 10th IEEE Sympo- sium on Computers and Comm.: June 27-30, 2005, Murcia, Spain. Piscataway, NJ, USA: IEEE, 2005: 625-630.

[15] KUMAR D, CHAHED T, ALTMAN E. Analysis of a Fountain Codes Based Transport in an 802.11 WLAN Cell[C]// Proc. of 21st International Teletraffic Congress: Sep 15-17, 2009, Paris, France. Piscataway, NJ, USA: IEEE, 2009: 1-8.

[16] SUNDARARAJAN J, SHAH D, MEDARD M, et al. Network Coding Meets TCP[C]// Proc. of INFOCOM: April 20-25, 2009, Rio de Janeiro, Brazil. Piscataway, NJ, USA: IEEE, 2009: 280-288.

[17] SUNDARARAJAN J, SHAH D, MEDARD M Network Coding Meets TCP: Theory and Implementation[J]. Proceedings of the IEEE, 2011, 1-23.

[18] BODINE E. Characterization of Luby Transform codes with small message size for low-latency decoding[C]// Proc. of ICC: May 19-23, 2008, Beijing, China. Piscataway, NJ, USA: IEEE, 2008: 1195-1199.

[19] LEU F, HUANG Y. Maximum Capacity in chain-topology wireless mesh networks[C]// Wireless Telecommunications Symposium: Apr 24-26, 2008, Pomona, USA. Piscataway, NJ, USA: IEEE, 2008: 50-259.

[20] YOO J. Maximum end-to-end throughput of chain-topology wireless multi-hop networks[C]// Proc. of Wireless Communications and Networking: Mar 11-15, 2007, Hong Kong, China. Piscataway, NJ, USA: IEEE, 2007: 4279-4283.

[21] FU Z, LUO H, ZERFOS P, et al. The impact of multihop wireless channel on TCP performance[J]. Transactions on Mobile, 2005, 4(2): 209-221.

[22] WANG K, YANG F, ZHANG Q, et al. Modeling path capacity in multi-hop IEEE 802.11 networks for QoS services[J]. IEEE Transactions on Wireless Communications, 2007, 6(2): 738-749.

[23] Network simulator (ns-2). [Online]. Available: http://www.isi.edu/nsnam/ns/