Computer Communications 000 (2016) 1–13

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

TCPJGNC: A transport control protocol based on network coding for multi-hop cognitive radio networks☆

Yang Qin[a,*], Xiaoxiong Zhong[a,b], Yuanyuan Yang[c], Li Li[a], Fangshan Wu[a]

[a] Key Laboratory of Network Oriented Intelligent Computation, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, PR China
[b] School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, PR China
[b] Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

A R T I C L E I N F O

Article history:
Received 12 November 2014
Revised 28 August 2015
Accepted 26 December 2015
Available online xxx

Keywords:
Cognitive radio networks
TCP
Network coding

A B S T R A C T

Cognitive radio (CR) has emerged as a promising solution to enhance spectrum utilization. In cognitive radio networks (CRNs), the secondary users (SUs) can opportunistically exploit frequency bands when the primary users (PUs) do not occupy the bands. However, TCP performance in CRNs may suffer from significant degradation due to this feature. In this paper, we investigate the limitations of TCP in multi-channel multi-radio multi-hop CRNs, and propose a novel transmission control protocol called TCPJGNC (TCP Joint Generation Network Coding, JGNC) based on network coding. In TCPJGNC, we dynamically adjust the number of packets involved in network coding according to the wireless communication environment to achieve better decoding probability. In the meantime, a coding scheme based on JGNC is provided which can reduce the number of retransmissions in TCPJGNC. In addition, we modify the TCP mechanism to fit into CRNs by considering the features of CRNs. An analysis of approximate expected throughput in TCPJGNC is provided and the simulation results indicate that TCPJGNC can significantly improve the network performance in terms of throughput, bandwidth efficiency and average end-to-end delay. To the best of our knowledge, TCPJGNC is the first transmission control protocol for multi-hop CRNs from a network coding perspective.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

With the demands of wireless technologies and applications, more and more spectrum resources are needed. Meanwhile, with the current spectrum allocation policy, all of the spectrum bands are exclusively allocated for licensed users (i.e., primary users PUs), and violation from unlicensed users (i.e., secondary users SUs) is not allowed. This is the main factor that leads to spectrum underutilization. The Federal Communications Commission (FCC) has indicated that temporal and geographical variations in the utilization of the assigned spectrum range from 15 to 85% [1]. Dynamic spectrum access [2] is proposed to solve the critical problem of spectrum scarcity. This new research area foresees the development of cognitive radio networks (CRNs).

The cognitive radio (CR) [3] principle has introduced the idea to exploit spectrum holes (i.e., bands) which result from the proven underutilization of the electromagnetic spectrum by modern wireless communication and broadcasting technologies. The exploitation of these holes can be accomplished by the notion of CRNs. CRNs have emerged as a prominent solution to improve the efficiency of spectrum usage and network capacity. In CRNs, the SUs can opportunistically exploit frequency bands when the PUs do not occupy the bands. Most of the research work that has been conducted in CRNs concentrates on the two lower layers, tackling PHYsical (PHY) layer and/or media access control (MAC) layer issues, including the definition of effective spectrum sensing, spectrum decision and spectrum sharing techniques [4,5]. Except for a few routing algorithms, the network layer protocols for CRNs are in the nascent stages of development [6–10]. Until very recently, the research community has started to realize the potentials of enhancing transmission control protocol (TCP) in multi-hop CRNs which can improve the performance of CRNs. TCP in CRNs exhibits similarities with the TCP in multi-channel multi-hop ad hoc networks/mesh networks, but with the additional challenge of having to deal with the dynamic behavior of the PUs, and their effects on changing spectrum opportunities (SOPs) of SUs. Some efforts have been taken to improve the TCP performance in CRNs through optimizing the lower-layer parameters [11–16] or modifying TCP itself [17–19]. Slingerland et. al. [11] evaluated the performance of TCP on CRNs. They considered different TCP variations including

☆ Part of this work was presented at IEEE Globecom 2011 [26].

* Corresponding author. Tel.: +8675526032794.
E-mail addresses: yqinsg@gmail.com (Y. Qin), xixzhong@gmail.com (X. Zhong), yuanyuan.yang@stonybrook.edu (Y. Yang), lili8503@aliyun.com (L. Li), 852001910@qq.com (F. Wu).

http://dx.doi.org/10.1016/j.comcom.2015.12.007
0140-3664/© 2015 Elsevier B.V. All rights reserved.

TCP NewReno [20] and TCP Vegas [21]. It was shown that the time taken by SUs in sensing mode affects the throughput of the TCP connections. Issariyakul et. al. [12] evaluated the performance of TCP NewReno on CRNs. They considered a new type of loss called service interruption loss, due to the existence of PUs. Their simulation results showed that there is an optimum number of channels for SUs to achieve maximum aggregate throughput. The optimum number of channels depends on the number of PUs and also the number of SUs on the network. In refs. [13,14], a scheme that optimizes TCP throughput without making changes to TCP was proposed. To achieve higher throughput in TCP Reno [22], their scheme optimized the low-layer parameters of the network, such as modulation and coding scheme in the physical layer, and frame size in the data-link layer. Similarly, Wang et. al. [15] investigated the TCP throughput performance enhancement for CRNs through lower-layer configurations. They studied the impacts of lower-layer parameters (e.g., packet error rate, queue length, spectrum sensing accuracy), PUs activities and channel conditions on the TCP throughput. However, they did not modify the congestion control mechanism in TCP Reno to respond to PU activities and spectrum sensing. Also they treat all types of losses as congestion losses in the scheme. Sarkar and Narayan [16] designed, implemented, and evaluated a transport protocol for CRNs. They augmented their protocol with TCP and TCP Westwood [23]. Their protocol handles temporary disconnections caused by spectrum sensing. Also, their protocol handles frequent bandwidth variation as the connection moves from one channel to another. However, they did not consider disconnection caused by PUs arrival.

On the other hand, from the perspective of modifying TCP itself, Felice et. al. [17] evaluated the performance of TCP over cognitive radio ad hoc networks (CRAHNs). They studied different TCP variations including TCP Reno, TCP NewReno and TCP Vegas by considering the impact of three factors on different TCP variants: (i) spectrum sensing cycle, (ii) interference from PUs and (iii) channel heterogeneity. Moreover, they analyzed the impact of CRAHNs characteristics over the route formation process, by considering different routing metrics and route discovery algorithms. Similarly, Chowdhury et al. [18,19] designed a transport protocol for CRAHNs. They modeled the transport protocol as a six-state system. Some of the events that cause the system to change state are route failure, congestion notification, node mobility, and spectrum change. However, it requires retransmissions from lower layers when data collision and loss of channel errors occur.

Network coding (NC) [24] has emerged as a promising technology to improve the performance of communications, especially in wireless networks. NC was originally proposed by Ahlswede et. al. in 2000, which allows intermediate nodes in the network to encode multiple packets together and forward more than one packet during a transmission. Using NC can enhance the performance of the wireless networks [25–27]. As we know, in traditional TCP, it uses feedback to acknowledge received packets in order. While incorporating NC into TCP, the feature is missing due to sending the linear combination of some original packets. TCP/NC [27] is one of the earliest TCP implementations which could incorporate NC with minor changes in the current protocol stack. However, in TCP/NC, source node transmits random linear combination of packets currently in the congestion window; it will be inefficient when the wireless environment changes, especially in multi-hop CRNs. In addition, in CRNs, due to the spectrum sensing and PUs activities, the available channels are instable. It will be more likely to drop packets than traditional wireless networks. Thus, the TCP performance of SUs will degrade when using the traditional TCP over CRNs. Therefore, designing a new TCP protocol for CRNs from a NC perspective is an urgent issue.

In this paper, we first study the characteristics of CRNs that lead to an obvious degradation in TCP performance, and then introduce the JGNC; finally, we evaluate the performance of incorporating JGNC in TCP over CRNs. We aim to improve the TCP performance from modifying TCP itself, rather than optimizing the lower-layer parameters as most of previous work. To the best of our knowledge, this is the first work studying NC incorporation with TCP in multi-hop multi-channel multi-radio CRNs.

The main contributions of this paper can be summarized as follows.

(1) We analyze decoding probability in wireless networks, which depends on the probability of linearly independent and packet loss rate. We also derive the decoding probability of JGNC in multi-channel multi-radio multi-hop CRNs, which considers PU activity.
(2) Considering the channel uncertainty in CRNs, we propose a novel TCP protocol TCPJGNC based on JGNC. In TCPJGNC, the number of the packets involved in network coding operation can be changed according to the wireless environment (packet loss rate) and redundancy factor. Also, we discuss the coding scheme of JGNC in TCP, which is set according to the changing ratio of ACK and the rank of the matrix consisting of the received packets' coding coefficient. In addition, we derive the expectation of the number of retransmissions in TCPJGNC, which is smaller than that of classical TCP and TCP/NC. TCPJGNC can significantly reduce the retransmissions and provide a higher decoding probability, and then enhance the TCP performance in multi-hop CRNs.
(3) We modify the TCP mechanism to fit into CRNs, by considering slow start, spectrum sensing state, spectrum changing state and presence of PUs, and give novel methods to calculate the effective window, retransmission time-out (RTO) and Round-Trip Time (RTT) for CRNs, respectively. In addition, an analysis of approximate expected throughput in TCPJGNC is provided.
(4) We present simulation results to evaluate the throughput, bandwidth efficiency and average end-to-end delay benefits of TCPJGNC under different network settings (e.g., PU activity, sensing time, and spectrum changing).

The rest of this paper is organized as follows. In Section 2, we first describe the system model considered in this study, and then analyze the challenges from four aspects: PU behavior, spectrum sensing, spectrum changing, and TCP itself in TCP performance over CRNs. We introduce the JGNC in Section 3. In Section 4, we describe our scheme TCPJGNC in detail, and give a throughput analysis model in Section 5. Simulation results are provided in Section 6. Section 7 concludes this paper.

2. System model and TCP's challenges in CRNs

2.1. System model

We now describe the model used in this paper for TCP performance analysis in CRNs. In our previous work [7], we assume an interweave model [28], i.e., the SUs in the CRNs can only transmit data when the PUs are not active. In this paper, we consider a time slotted multi-hop cognitive radio network with num_s SUs and num_p PUs. Each node (including SU and PU) is equipped with the same number of radios. We assume half-duplex on each radio. Each SU is capable of sensing the locally available channels and has the capability of channel changing at packet level for data transmission. On each given channel, PU activity is modeled as a Poisson process in which the lengths of both periods are exponentially distributed with rate λ_{busy}(the channel is occupied by the PU) and λ_{idle}(the channel is available for the SU), respectively.

A time-slotted model for SU is assumed, with a fixed slot duration T. Each slot consists of a sensing period with duration

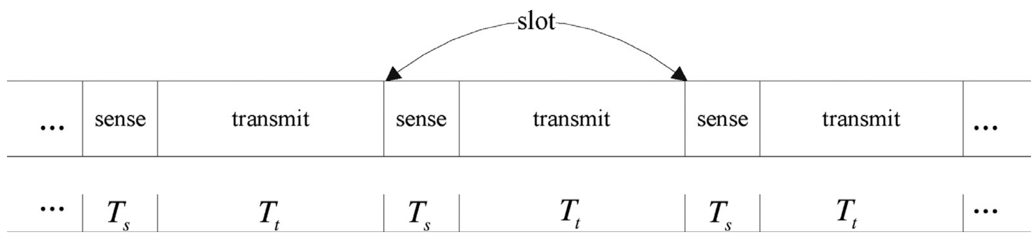

Fig. 1. The time-slotted model for CRNs.

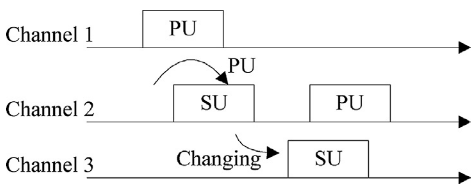

Fig. 2. Spectrum state in CRNs.

T_s and a data transmission period with duration T_t (T_t $=T$-T_s), as shown in Fig. 1.

2.2. TCP's challenges in CRNs

In CRNs, the SUs can opportunistically exploit frequency bands when the PUs currently do not occupy. When a PU appears on a channel, the SU should vacate it for the PU. The SU's communication mainly depends on the PU's activities, which is the biggest difference between CRNs and traditional wireless networks such as ad hoc networks and mesh networks. The process of the sharing spectrum between PUs and SUs is shown in Fig. 2. In this section, we discuss the impact on TCP performance in CRNs in terms of following key factors: (i) PU behavior, (ii) spectrum sensing, (iii) spectrum changing, and (iv) TCP itself.

2.2.1. PU behavior

In multi-hop CRNs, the PUs have the highest priority to use the channel, whether SUs are in the spectrum sensing or data transmitting phase. On detecting the presence of PUs, the SUs must immediately cease their operations on that channel and search for another vacant channel, which would cause a large amount of packet loss. If there are no vacant channels currently, SUs have to wait until the next sensing cycle. In this case, SUs have to disconnect the TCP connections. If an idle channel is available through sensing operations, the SUs can continue to transmit data packets.

If the duration between a PU arrival and the next sensing cycle is long, all the packets the source sent will be stored at an upstream node of the sensing node during this period. When the link connected with the sensing node and its upstream node is restored to normal state, the upstream node will send all cached data packets to the sensing node at maximum speed. It will cause a congestion and packet loss, leading to significant TCP throughput degradation.

2.2.2. Spectrum sensing

In CRNs, the spectrum sensing is a periodical process that monitors the current channel over a pre-defined sensing duration for the occurrence of a PU. In the time-slotted model, it includes two states: sensing state and transmitting state. In sensing state, the link is in a virtual connection state; and the nodes that perform spectrum sensing cannot transmit/receive data packets. In sensing state, the source node does not know the status of intermediate nodes. Thus, it still sends data packets to them, and a lot of data packets have to be stored at the node preceding the sensing node. If the sensing duration is long, the cache of the node will overflow, which will result in all data packets sent after sensing state be lost. On the other hand, in sensing state, the source node does not know that the destination node has sent back an ACK, thus, it still sends the packet that the destination node has received. It will be more duplicates at destination node, which will largely reduce the TCP throughput.

2.2.3. Spectrum changing

If a PU arrives, the affected SU immediately vacates the channel for its transmission. Thus, the SU ceases its transmission and searches for the idle channels. If it currently has an idle channel in its channel list, it changes from the former channel to the idle channel. In this case, there may have a large variation in bandwidth [11,16,18,19]. Also, in the spectrum changing state, it may induce interruption, which may increase the RTT or trigger a TCP's RTO.

2.2.4. TCP itself

In CRNs, the network topology frequently changes, due to PU activity. Thus, the CRNs are extremely unstable. There is a new type of packet loss in CRNs, called interruption loss, which is caused by PU's activity. This process frequently occurs due to PU arrivals on the current channel frequently. If we perform the same operations on TCP (TCP reduces the congestion window to 2 segments and resets to the slow-start state) as traditional packet loss occurs, the traditional TCP mainly keeps a small congestion window in CRNs, the TCP does not provide a better performance for CRNs. As a result, if we can distinguish the interruption loss causing by PU arrivals, the TCP throughput will be improved.

3. Joint Generation random linear network coding (JGNC)

In this section, we derive the probability of successfully decoding over three coding schemes: random linear network coding [29] (RLNC), Multi-Generation RLNC (MGNC), and JGNC.

3.1. Decoding probability

In RLNC scheme, the source generates a coding packet from the original packets by multiplying with a randomly generated code vector (namely coding coefficients), which consists of random elements of a selected finite field $GF(q)$. The destination will successfully decode the size of data B, only if it receives enough linearly independent packets. We know that to decode all original data packets, the probability of linearly independent and the packet loss rate between the source and the destination are crucial. The probability of linearly independent is related to the coding field size from finite field. When the $GF(q)$ is sufficiently large, e.g., $q = 2^8$, the probability of linearly independent p_{ind} is 0.996 [30]. Thus, we choose the field size $GF(2^8)$ in our following analysis.

Proposition 1. *In RLNC, given the size of data for network coding B and packet loss rate p on each channel. The probability that a node receives at least B coded packets out of n (n >B) packets transmitted by the node is*

$$p_r = 1 - \Phi\left(\frac{B-\mu}{\delta}\right) \quad (1)$$

where $\mu = n(1-p)$, $\delta^2 = np(1-p)$.

Proof. We assume the block size for network coding is B and packet loss rate is p for each channel. Assume the probability that a node receives at least B coded packets out of n packets transmitted by the sink node is p_r. Then we have

$$p_r = \sum_{i=B}^{n} \binom{n}{i} (1-p)^i p^{n-i} \tag{2}$$

According to the central limit theorem, when n tends to infinity, Bernoulli distribution is close to the normal distribution. In order to facilitate the calculation, we use the normal distribution to replace the Bernoulli distribution. So, we have

$$p_r = \int_B^n \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(x-\mu)^2}{2\delta^2}} dx \tag{3}$$

where $\mu = n(1-p)$, $\delta^2 = np(1-p)$.

To convert it into a standard normal distribution, we have

$$p_r = 1 - \Phi\left(\frac{B-\mu}{\delta}\right)$$

That completes the proof.

In practice, the p_r will be closed to the steady state when n is enough large, not tend to be infinity, which depends on B and p, e.g., when $n = 112$, the p_r will be 1, under this scenario, $B = 100, p = 0.1$.

3.1.1. Decoding probability of Multi-Generation RLNC (MGNC)

We assume the data size for network coding is B and packet loss rate is p for each channel. The entire data block is divided into m generations, so each generation contains $k = B/m$ packets.

Proposition 2. *In MGNC, the decoding probability for the entire information group p_b is*

$$p_b = 1 - \Phi\left(\frac{k-\mu_j}{\delta_j}\right) = 1 - \Phi\left(\frac{1}{\sqrt{m}} \times \frac{B-\mu}{\delta}\right) \tag{4}$$

where $\mu = n(1-p)$, $\delta^2 = np(1-p)$.

Proof. Assume the probability that a node receives at least k coded packets of the same group transmitted by the source node is p_j. We have

$$p_j = \sum_{i=k}^{n/m} \binom{n/m}{i} (1-p)^i p^{n/m-i} \tag{5}$$

As $m << n$, when n tends to infinity, n/m will also tend to infinity. According to the central limit theorem, Bernoulli distribution is close to normal distribution.

$$p_j = \int_k^{n/m} \frac{1}{\sqrt{2\pi}\delta_j} e^{-\frac{(x-\mu_j)^2}{2{\delta_j}^2}} dx \tag{6}$$

where $\mu_j = (n/m)(1-p)$, ${\delta_j}^2 = (n/m)p(1-p)$.

Let p_b denote the probability of decoding for the entire group, we have

$$\begin{aligned} p_b &= \sum_{j=1}^{m} (p_j/m) \\ &= \sum_{j}^{m} \left(\frac{1}{m} * \left(\int_k^{n/m} \frac{1}{\sqrt{2\pi}\delta_j} e^{-\frac{(x-\mu_j)^2}{2{\delta_j}^2}} dx\right)\right) \\ &= \int_k^{n/m} \frac{1}{\sqrt{2\pi}\delta_j} e^{-\frac{(x-\mu_j)^2}{2{\delta_j}^2}} dx \end{aligned} \tag{7}$$

To convert it into a standard normal distribution, we have

$$p_b = 1 - \Phi\left(\frac{k-\mu_j}{\delta_j}\right) = 1 - \Phi\left(\frac{1}{\sqrt{m}} \times \frac{B-\mu}{\delta}\right)$$

That completes the proof.

In practice, the p_b will be closed to the steady state when n and m are enough large, not tend to be infinity, which depends on B and p, e.g., when $n = 120$ and $m = 35$, the p_b will be 0.6596, under this scenario, $B = 100, p = 0.1$.

In fact, RLNC is a special form of MGNC. The decoding probability is determined by the formula (1) and formula (4). When B and n are fixed, we have the following cases.

If the source node sends n ($n \geq B$) coded packets and the destination node receives more than B packets, we have $(B-\mu)/\delta < 0$. It means that the packet loss rate is considered to be low. When the number of packets sent by the source node is fixed, the smaller generation is divided, the lower decoding probability it has. As shown in Fig. 3, when $(B-\mu)/\delta = -0.5$, the decoding probability of RLNC is the shaded area on the right of the straight line $x = -0.5$; while using generation technology and dividing the generation into 4 small generations on average, the decoding probability is the shaded area on the right of the straight line $x = -0.25$. This indicates that a large size of NC generation could be used for the wireless channels with low loss.

If the source node sends n ($n \geq B$) coded packets and the destination node receives less than B packets, we have $(B-\mu)/\delta > 0$. It means that the packet loss rate is considered to be high. When the number of packets sent by the source node is fixed, the smaller generation is divided, the higher decoding probability it has. As shown in Fig. 4, when $(B-\mu)/\delta = 0.5$, the decoding probability of RLNC is the shaded area on the right of the straight line $x = 0.5$; while using generation technology and dividing the generation into 4 small generations on average, the decoding probability is the shaded area on the right of the straight line $x = 0.25$. Clearly, in this case, it can improve the decoding probability using generation technology. It means that the size of generation should be smaller in the wireless networks with high loss.

From the above analysis, we can observe that the generation size of network coding should be selected according to the wireless environments. Thus, in our scheme, the generation size k^* is set according to the packet loss rate and redundancy, therefore, it should be

$$\hat{k} = \max\{k | (1+R) \times (1-p)^k \geq 1, k \in \mathbb{Z}^+\} \tag{8}$$

$$k^* = \min\{\hat{k}, cwnd\} \tag{9}$$

where R is redundancy factor, p is the packet loss rate, k is the generation size, and the *cwnd* is the size of the TCP congestion window.

The probability that the destination node receives a total of k packets is $(1-p)^k$, thus, only if $(1+R) \times (1-p)^k \geq 1$, the destination node can decode the original k packets. Note that, we choose the maximal value of k for $(1+R) \times (1-p)^k \geq 1$. In addition, if we incorporate NC into TCP, the generation size k^* should be smaller than or equal to the size of the congestion window. Also, if $k^* = 1$(e.g., $R = 0.2$, $p = 0.1$), we set k^* to be 2 due to performing network coding operations.

3.1.2. Joint Generation RLNC

In JGNC, suppose the data is divided into m generations, and each generation has k packets and be independently encoded as illustrated in Fig. 5. Clearly, the entire data block has $n = m \times k$ packets. The destination only needs to receive k independent packets for decoding the original packets over each generation. When

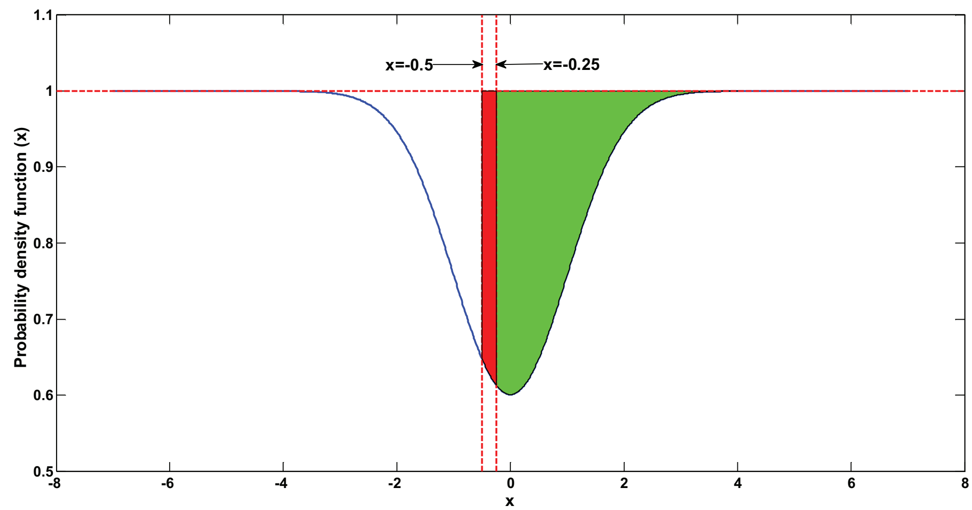

Fig. 3. Decoding probability in a network with low packet loss rate.

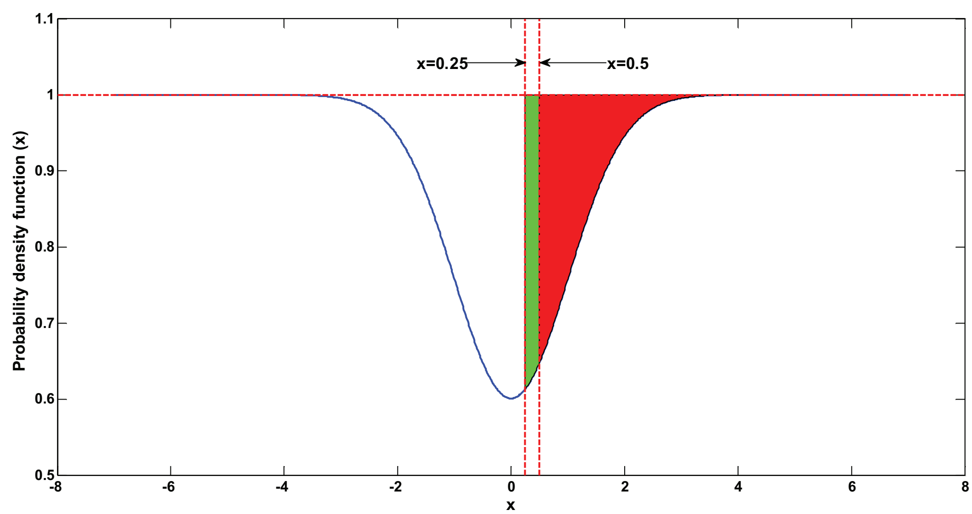

Fig. 4. Decoding probability in a network with high packet loss rate.

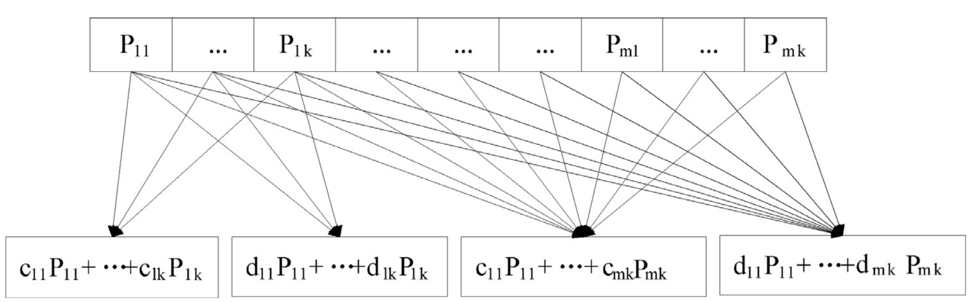

Fig. 5. Joint generation RLNC.

the destination receives less than k independent packets belonging to the first generation, the source will not transmit first generation again; instead, it combines this generation with the second generation to be encoded, then transmit. It is still possible for that generation can to be decoded with the help of other subsequent generations. When the destination receives $2k$ independent packets, it can decode these two generation packets. If not, all the packets of these two generations are encoded with the packets of the third generation; when the destination receives $3k$ independent packets, it can decode the packets. And so on, until k generation packets are encoded, as shown in Fig. 5. There are n packets in the buffer, P_{11},...P_{1k}, ..., P_{m1},...,P_{mk}. c_{11},...,c_{1k}, d_{11},...,d_{1k}, c_{11},...,c_{mk}, and d_{11},...d_{mk} are coding coefficient. Of course, if the destination receives enough packets to decode a generation, then it is not necessary to joint this generation with the next generation to be encoded. If we joint all the generations, it still does not decode, then we should retransmit some encoding packets to the destination node for obtaining the original packets.

We assume the data size for network coding is B and packet loss rate is p for each channel. The data are divided into m

Please cite this article as: Y. Qin et al., TCPJGNC: A transport control protocol based on network coding for multi-hop cognitive radio networks, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2015.12.007

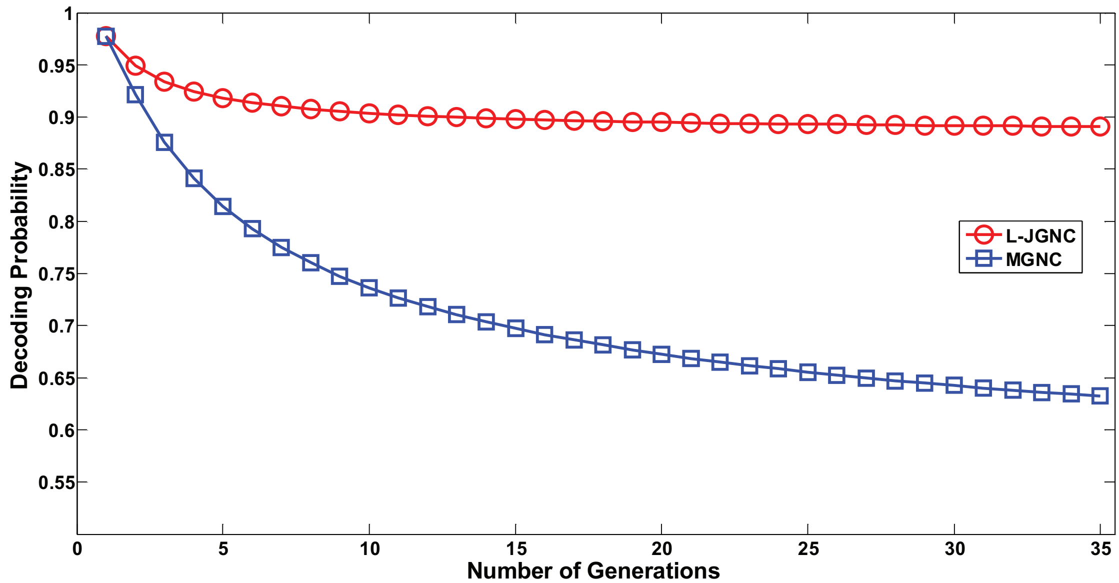

Fig. 6. Decoding probability of the L-JGNC and MGNC when $(B-\mu)/\delta=-2$.

generations, so each generation contains $k=B/m$ packets. Then we can obtain the decoding probability of JGNC p_{joint}

$$\begin{aligned} p_{joint} > \; & \frac{1}{m}\sum_{i=mk}^{n}\binom{n}{i}(1-p)^i p^{n-i} \\ & + \frac{1}{m}\sum_{i=(m-1)k}^{\frac{(m-1)n}{m}}\binom{\frac{(m-1)n}{m}}{i}(1-p)^i p^{\frac{(m-1)n}{m}-i} \\ & + \cdots + \frac{1}{m}\sum_{i=k}^{\frac{n}{m}}\binom{\frac{n}{m}}{i}(1-p)^i p^{\frac{n}{m}-i} \\ = \; & \frac{1}{m}\left(1-\Phi\left(\frac{B-\mu}{\delta}\right)\right)+\frac{1}{m}\left(1-\Phi\left(\frac{\sqrt{m-1}}{\sqrt{m}}\times\frac{B-\mu}{\delta}\right)\right) \\ & + \cdots + \frac{1}{m}\left(1-\Phi\left(\frac{1}{\sqrt{m}}\times\frac{B-\mu}{\delta}\right)\right) = p_{L-JGNC} \end{aligned} \quad (10)$$

where $\mu=n(1-p)$, $\delta^2=np(1-p)$.

In the following, we compare the decoding probability of JGNC and MGNC. We can see that when $(B-\mu)/\delta<0$, e.g., $(B-\mu)/\delta=-2$, $(n=117,\ B=100,\ p=0.1)$ (it means that the destination receives more than B packets), the decoding probability of JGNC (the lower bound of JGNC, $p_{L\text{-}JGNC}$) is larger than that of MGNC, as shown in Fig. 6. It means that the size of generation should be larger in this scenario. We should choose the JGNC coding scheme for providing higher decoding probability if we adopt generation technology in the wireless environments with low loss.

On the other hand, according to our PU activity model, we can obtain the probability that at least one of the num_p PUs affecting a link will be active over time period t, which can be written

$$p_{busy}=1-e^{-t\sum_{i=1}^{num_p}\lambda_{busy}} \quad (11)$$

From the above analysis, we can see that there are three factors affecting the decoding probability of JGNC in multi-hop CRNs. The probability of successfully recovering all original packets is:

$$p_d=p_{joint}\times p_{ind}\times(1-p_{busy}) \quad (12)$$

where p_{joint} is the decoding probability of JGNC and p_{ind} is the probability of linearly independent.

Next, we give some numerical results for decoding probability in CRNs, as shown in Fig. 7. We set the number of PUs is 4 or 6 or 8, and the p_{joint} is 0.90 (joint 10 generations in JGNC). The p_{ind} is 0.996 (the field size is $GF(2^8)$) and t is 2 min. From Fig. 7, we can see that as the PU arrival rate increases, the decoding probability will decrease. Also, as the number of PUs increases, the decoding probability will decrease. It means that the features of CRNs can affect the decoding probability.

4. TCPJGNC: A transport protocol based on JGNC for multi-hop CRNs

In this section, we describe a novel transmission control protocol, TCPJGNC for multi-channel multi-radio multi-hop CRNs in detail in which we modify some mechanisms of the traditional TCP according to the features of CRNs.

Compared to the traditional wireless networks, the channel availability is uncertainty in CRNs, thus, the communications of SUs are intermittent. We use an adaptive coding scheme in TCPJGNC according to wireless environment, which can provide a better performance for multi-hop CRNs. Thus, we make some following improvements:

(1) Different from TCP/NC, (uses all the packets in the congestion window for network coding), TCPJGNC can dynamically choose the number of packets involved in network coding operation. And we set it according to the formulas (8) and (9), which consider the redundancy factor, packet loss rate and the TCP congestion window. The probability that the destination node receives a total of k (generation size) packets is $(1-p)^k$, thus, only if $(1+R)\times(1-p)^k\geq 1$, the destination node can receive k useful packets to be able to decode the k original packets successfully. Also, the generation size should be smaller than or equal to the size of the TCP congestion window.

(2) We give the condition of coding scheme for the TCPJGNC, which is set according to the changing ratio of ACK and the rank of a matrix consisting of the received packets' coding coefficient. When the rank is smaller than generation size k and the changing ratio of ACK is larger than zero, we deploy JGNC scheme for a high decoding probability. Also, the condition of ceasing the JGNC is provided, which depends on

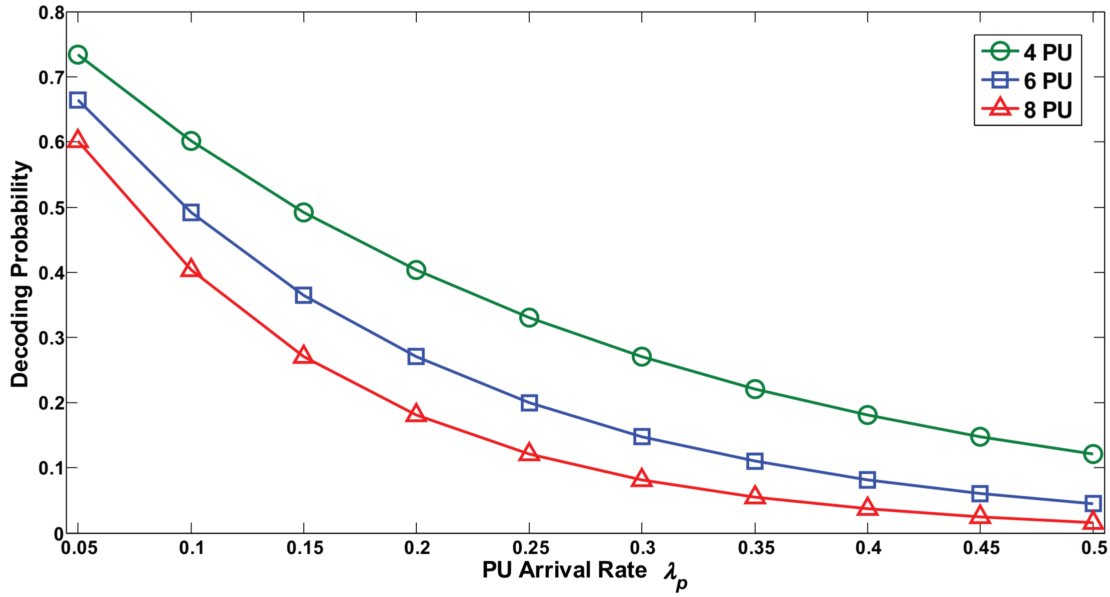

Fig. 7. The decoding probability vs. PU arrival rate.

decoding probability with a joint generation scheme. When the gain from joint generation network coding increases slowly, we will cease JGNC scheme. If the changing ratio of ACK is smaller than zero, we retransmit some coded packets, which are the random linear combination of the original packets from the current generation, considering the difference in the number of packets that the receiver has seen and the generation size, and the redundancy.

(3) We modify the TCP mechanism to be suitable for CRNs from the following four aspects: (i) PU behavior, (ii) spectrum sensing, (iii) spectrum changing, and (iv) TCP itself.

4.1. TCPJGNC design

In TCPJGNC, we add a coding window in the TCP layer. Thus, the TCP layer maintains two windows: TCP window and coding window.

The sender module receives original packets from the TCP source and delivers them to the coding window. After receiving a certain number of original packets, the coding window generates a linear combination of the packets in the coding window and delivers them to the IP layer.

The receiver module also maintains a buffer of linear combinations of packets that have not been decoded yet. Upon receiving a coding packet, the receiver module first retrieves the coding coefficients from the packet's header and appends it to the basis matrix which stores each received packet's coding coefficient. Then calculates the rank of the matrix as follows: if the rank increase by 1, then the receiver module constructs an ACK packet and sends it to the source, otherwise, the receiver will discard this packet and return to wait for another incoming packet. Once the value of rank is equal to *End-Start* (generation/block size), the receiver module decodes the packets by performing Gaussian elimination, then delivers original data to the TCP sink and clears the data in the coding window; otherwise, it will continue to buffer the encoded packet.

If the value of rank received by the sender is smaller than generation size, it means that some coded packets are lost or the corresponding ACKs are lost. In the case of corresponding ACKs are lost, the retransmission is unnecessary. To prevent this, we can adopt JGNC scheme. However, since only when the joint generation size of network coding is in a small range, the coding gain of

Algorithm 1 Encoding algorithm.

k: generation size, the number of packets in each generation. The sequence is from *Start* to *End*.
diff_recvACK_new: the difference in the number of the received ACK for this period of time.
diff_recvACK_old: the difference in the number of the received ACK in the previous period.
R: redundancy factor.
rank: the number of packets the receiver has seen in current generation.

$Start \leftarrow 0, End \leftarrow k, NUM \leftarrow 0, NUM \leftarrow NUM + R + 1$
while $\lfloor NUM \rfloor > 0$ **do**
  Generate an encoded packet, which is a random linear combination of the original data packets from the current generation.
  $NUM \leftarrow \lfloor NUM \rfloor - 1$
end while
if $rank < End - Star$ **then**
  if $diff\_recvACK\_new - diff\_recvACK\_old \geq 0$ **then**
    Generate an encode packet, which is a random linear combination of the original data packets from the current generation to the next generation.
  end if
  if $diff\_recvACK\_new - diff\_recvACK\_old < 0$ **then**
    Retransmit $((End - Start) - rank) \times (1 + R)$ coded packets which are random linear combination of the original data packets from the current generation.
  end if
end if
if $rank = End - Star$ **then**
  $Start \leftarrow End, End \leftarrow End + k$
end if

JGNC can significantly increase, we only allow limit generations to be involved according to (10). The encoding algorithm is shown in **Algorithm 1**.

In **Algorithm 1**, we use the changing ratio of ACK for the condition of coding scheme in TCPJGNC. Hence, we should calculate the number of the received ACK over a certain time T_{cal}. The "old" means the calculation of the number of the received ACK in the previous period, and the "new" means the calculation of the number of the received ACK in this period of time. The *diff_recvACK_old* is the difference in the number of the received ACK in the previous period. The *diff_recvACK_new* is the difference in the number of the received ACK for this period of time. Actually, these two variables are used to calculate the number of the received ACK over a certain time. Also, the duration of the two periods of time is equal.

Note that, in our scheme, the number of coding generations is limited. It can be calculated by the formula (10). If the coding gain - the decoding probability increases slowly, the JGNC will cease. Also, the period of time T_{cal} for calculating the difference in the number of the received ACK, plays a key role in TCPJGNC. It should be more than one RTO. In our simulation, we set the maximal number of coding generations to be 15 and T_{cal} is 5RTO.

Theorem 1. *The scheme of incorporating JGNC into TCP has fewer retransmissions than the classical TCP and TCP/NC.*

Proof. We assume the data are divided into m generations, and each generation has k packets. Clearly, the entire data block has $B = m \times k$ packets. Let p denote the packet loss rate for each channel. We sent n ($n > B$) packets in the entire data transmission. Also, we assume that each lost packet can be correctly received after one retransmission.

No-NC: If the ith packet is lost, which means the former (i-1) packets are correctly received, then the system has to retransmit (n-i+1) packets. Thus, we can obtain the expectation of retransmission without NC:

$$\begin{aligned} E[N_1] &= \sum_{i=1}^{n} (1-p)^{i-1} \times p \times (n-i+1) \\ &= (n+1)\sum_{i=1}^{n} (1-p)^{i-1} \times p - \sum_{i=1}^{n} (1-p)^{i-1} \times p \times i \\ &= (n+1)[1-(1-p)^n] + n \times (1-p)^n - \frac{1-(1-p)^n}{p} \\ &\approx n - \frac{1-(1-p)^n}{p} (\text{when } n \text{ is very large}) \end{aligned} \tag{13}$$

NC: in TCP/NC, if a coded packet is lost, the system simply retransmits a new coded packet. Thus, the expectation of retransmission is

$$E[N_2] = \sum_{i=0}^{n} C_n^i \times (1-p)^{n-i} \times p^i \times i = n \times p \tag{14}$$

JGNC: in TCPJGNC, the number of transmission of the coded packets of a generation is at most K times, $K \in (k, 2k)$. Let $M = \min(K-k, k)$, the expectation of retransmission of the first generation is

$$\sum_{i=0}^{M} C_k^i \times (1-p)^{k-i} \times p^i \times i \leq \sum_{i=0}^{k} C_k^i \times (1-p)^{k-i} \times p^i \times i = k \times p \tag{15}$$

When the first and second generations are jointly coded, the number of transmission of the coded packets of these two generations is at most 2 K times, including K times for the first generation. The expectation of retransmission of the first two generations is

$$\begin{aligned} &\sum_{i=0}^{M} C_k^i \times (1-p)^{k-i} \times p^i \times i + \sum_{i=k+M}^{k} C_k^i \times (1-p)^{k-i} \times p^i \times M \\ &+ \sum_{i=0}^{M} C_k^i \times (1-p)^{k-i} \times p^i \times i \\ &\leq 2 \times \sum_{i=0}^{k} C_k^i \times (1-p)^{k-i} \times p^i \times i = 2k \times p \end{aligned} \tag{16}$$

Similarly, after transmitting m generations, the number of retransmissions should be no greater than $m \times k \times p = n \times p$, denoted as:

$$E[N_3] \leq n \times p \tag{17}$$

According to (13), (14), (17), we can obtain

$$E[N_3] \leq E[N_2] < E[N_1] \tag{18}$$

That completes the proof.

4.2. Modifications in TCP

In this section, we modify the TCP from the following four aspects: (i) PU behavior, (ii) spectrum sensing, (iii) spectrum changing, and (iv) TCP itself.

4.2.1. PU behavior

In CRNs, PUs have the highest priority to use channels. SUs will give up the channels for PUs when PUs arrive. There is a new loss-service interruption for SUs due to the arrivals of PUs. The more activities PUs have, the less time the SUs use to transmit data. As a result, the number of service interruptions and loss probability become larger; TCP throughput of SUs is also lower. In our scheme, we exploit an Explicit Congestion Notification (ECN) to notify the source node that a service interruption occurs, which is generated by the affected node. It distinguishes the interruption loss from traditional loss caused by congestion. To improve TCP throughput, the TCP window will not decrease over a service interruption. For data collision and channel loss, we adopt network coding for data transmission, so that most of the lost data can be recovered at the receiver, reducing retransmissions and the probability of collision between the SU and the PU.

4.2.2. Spectrum sensing

In sensing state, the sensing node does not know the status of successor/predecessor nodes. The existing wireless TCP protocol does not know the lower layer's behavior, thus, it still sends data packets. As a result, a lot of data packets have to be stored at the node preceding the sensing node. Moreover, there are no ACKs sent from the successor of the sensing node. If the sensing duration is long, the cache of the node will overflow. Thus, we modify the effective window, *ewnd* at the sender to

$$ewnd = \min\left\{cwnd, rwnd, B_{residual\_buffer\_\min}\right\} \tag{19}$$

where *cwnd* is the current congestion window, *rwnd* is the receive window and $B_{residual\_buffer\_\min}$ is the minimum residual buffer of the node i which the TCP sessions pass through. We have

$$B_{residual\_buffer\_\min} = \min\left\{B_{residual\_buffer}(i)\right\} \tag{20}$$

where $B_{residual\_buffer}(i)$ is the residual buffer of the node i.

On the other hand, in sensing state, it will take some time to sense the channel over a sensing cycle, to avoid a timeout occurs earlier, we should modify the calculating method of RTO to

$$RTO = RTO + T_s + T_c \tag{21}$$

where

$$T_s = \sum_i t_s \tag{22}$$

$$T_c = \sum_i t_c \tag{23}$$

In Eqs. (22) and (23), t_s is sensing time of a node, t_c is spectrum changing time of a node, and i is the number of sensing /changing nodes passing through the TCP session.

4.2.3. Spectrum changing

RTO mechanism is a key factor in achieving reliable TCP protocols, which is set according to RTT. In CRNs, the RTT is closely related to the channel used by the user. When the channel bandwidth used by a SU is larger, it has a faster transmission rate.

Please cite this article as: Y. Qin et al., TCPJGNC: A transport control protocol based on network coding for multi-hop cognitive radio networks, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2015.12.007

The RTT value is relatively small, the probability that a SU conflicts with a PU in a SU data transmission is small, and vice versa. When channel bandwidth used by a SU changes significantly, the RTO will become too large or too small. To precisely calculate the value of RTO, we update RTT according to the formula (24) [18] and the new method of calculating RTT in coded TCP [27]. $L^T_{i,i-1}/L'^T_{i,i-1}$ is the bidirectional link latency between node *i* and node *i-1* after/before spectrum changing, the RTT' represents the *RTT* before spectrum changing. Since the bandwidth will change after the spectrum changing, in order to improve the utilization of the channel, we set the congestion window according to Eq. (25), where B_{new}/B_{old} represents the bandwidth after/before spectrum changing, and $cwnd_{new}/cwnd_{old}$ is the size of the congestion window after/before spectrum changing.

$$RTT = RTT' + L^T_{i,i-1} - L'^T_{i,i-1} \tag{24}$$

$$cwnd_{new} = cwnd_{old} \times \frac{B_{new}}{B_{old}} + 1 \tag{25}$$

4.2.4. TCP itself

When a connection is established, in order to prevent a large number of packets from being immediately injected into the network, causing network congestion, TCP initially sets $cwnd = 2$, if this segment is ACKed before the timeout, the *cwnd* increases by one until it reaches the maximum segment size. However, in CRNs, the *cwnd* may not increase to half of the maximum value due to channel uncertainty. If we use traditional method that reduces *cwnd* by a half when a congestion event occurs, it greatly reduces the efficiency of spectrum resources. Thus, after service incorruptions, if the timeout does not occur, we let *cwnd* linearly grow until the maximum size is reached or a timeout or congestion due to other loss occurs. If the congestion occurs due to other types of loss, we update *cwnd* by its half value, and then *cwnd* grows linearly until the maximum size is reached or a timeout occurs or congestion occurs.

5. TCPJGNC throughput analysis

In this section, we derive an analytical expression for TCPJGNC throughput in multi-hop CRNs. In CRNs, we analyze the throughput performance from three states: normal state, spectrum sensing state and spectrum changing state, which is similar to ref. [19]. However, in our scheme, we modify the calculating method of throughput in normal state, which is exploiting network coding in TCP protocol to mask the random losses within the network. In addition, our scheme is based on TCP Vegas, whose control mechanism is smoother using RTT, compared to the more abrupt loss-based variations of TCP Reno.

In ref. [19], the authors derived the TCP CRAHN throughput (B_t) from these three aspects: normal state, spectrum sensing state and spectrum changing state.

$$B_t = \frac{B_n \times P_n \times T_p + B_s \times P_s \times T_s + B_c \times P_c \times T_c}{P_s \times T_s + P_c \times T_c + P_n \times T_p} \tag{26}$$

where P_n /P_s /P_c is the probability that the network is in the normal state/spectrum sensing state/spectrum changing state, B_n /B_s /B_c is the TCP throughput in the normal state/spectrum sensing state/spectrum changing state, T_p /T_s /T_c is the time used for data packets transmission/spectrum sensing/spectrum changing in the network. According to Chowdhury et al. [19], we can calculate the parameters B_s, B_c, P_n, P_s, P_c, also, the T_s /T_c can be obtained from Eqs. (22) and (23). However, in our scheme, TCPJGNC, B_n is different from that in [19], which provides a new method to incorporate NC into TCP protocol.

In the following, we focus on how to calculate the parameter B_n and give an approximate model of throughput in the normal state. In TCP Vegas Time-Outs (TOs) model [31], loss episodes are identified by duplicate ACKs or by TOs. A TO occurs if after a loss episode, not enough duplicate ACKs return to the sender to trigger lost packet retransmissions. Similarly, we assume that all TO series consist of a single TO. The probability of the packet error assuming a single bit error is irrecoverable is $p_e = 1 - (1 - BER)^{DATA}$ where *DATA* is the size of the TCP segment, and *BER* is the bit error rate. At any given round *i*, TCPJGNC sender transmits $R \times E[W_i]$ coded packets, and $p \times R \times E[W_i]$ will be lost, where $p = (1 - p_{ind}) + (1 - (1 - {p_e}^L)^k)$, *R* is the redundancy factor in TCPJGNC, W_i is the window size in round *i*, and *L* is the maximum number of retries at MAC layer.

Based on [19,31], we can obtain an approximate model of throughput over *r* rounds in the normal state for TCPJGNC.

$$B_n = \frac{1}{r} \times \sum_{i=1}^{r} \frac{(N_E + 1) \times (\frac{1-p}{p} + W_i)}{N_{SS2TO} \times RTT_i + T_o} \tag{27}$$

where

$$N_{SS2TO} = 2\log W_i + (N_E + 1)\frac{1-p}{pW_i} + \left(1 + \frac{N_E}{4}\right)\frac{W_i}{8} + \frac{9N_E}{8} - \frac{11}{4} + \frac{4 - 2^{\log W_i}}{W_i},$$

(*SS2TO*: Slow-Start-to- Timeout), as defined in [31],
$W_i = \min(E[W_1] + i\min\{1, R(1-p)\}, W_{\max})$,
$E[W_1]$ is the initial window size, and set $E[W_1] = 1$,
α and β are the TCP Vegas throughput thresholds, let $\alpha = 1$ and $\beta = 3$ in our simulation,
RTT_i is the *RTT* in round *i*,
N_E is expected number of consecutive LFPs (Loss Free Period) during an SS2SS (Slow-Start-to- Slow-Start) period,
$W_{\max}$ is the maximum window size advertised by the receiver,
T_o is the average duration of the first TO in a TO series.

Note that in the classical TCP throughput model [32], it does not have the additional spectrum sensing and spectrum changing states. Thus, in CRNs, similar to [19], we should modify it, where the method of calculating RTT is different. In addition, as mentioned earlier in **Theorem 1**, incorporating JGNC into TCP has fewer retransmissions, which will eventually provide smaller RTT and TO time. Hence, TCPJGNC can provide significant throughput gains in CRNs.

6. Performance evaluation

In this section, we present simulation results to evaluate the throughput, average end-to-end delay and bandwidth efficiency benefits of our new protocol. The simulations are based on TCP-Vegas (denoted TCP), TCP/NC (denoted TCPNC) and TCPJGNC using NS2 [33] and CRCN model [34]. The topology for the simulations consists of two PUs and five SUs, as shown in Fig. 8. Each node is equipped with 2 radios. The total number of the available channels is 3. We assume that the probability that all nodes along the TCP connection perform spectrum sensing simultaneously is small. This is because in our simulation, we set the sensing time t_s of a node to be $t_s + random(0, 0.01)$ ms. The transmission ranges of the PU and the SU are 550 and 250 m. One FTP application wants to communicate from the source node to the destination node. All these links have a bandwidth of 2 Mbps and a propagation delay of 50 ms. The packet size is 1000 bytes. The *R* is set to 0.2 in the schemes based on network coding. The buffer size is set to 200. The TCP receive window size is set at 100 packets. The sensing time t_s is 5 ms, and the sensing cycle is 400 ms. The spectrum changing time t_c is 80 μs. The PU on time, $(1/\lambda_{busy})$, is 1 s. The simulations are averaged over 50 times and each lasts 100 s.

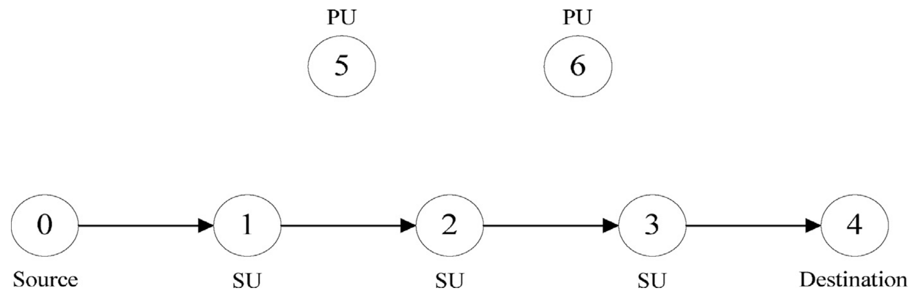

Fig. 8. A multi-hop cognitive radio network.

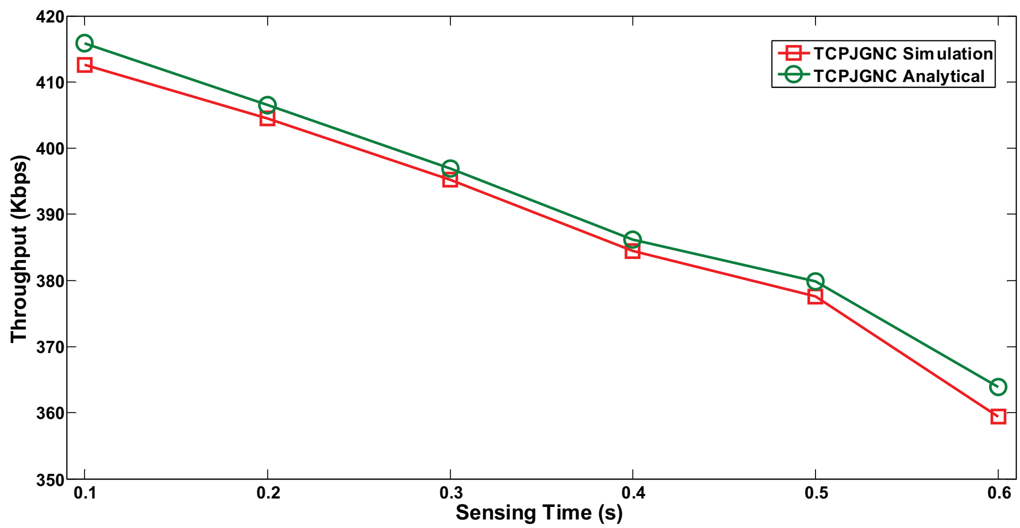

Fig. 9. The analytical throughput and simulation throughput of TCPJGNC vs. sensing time.

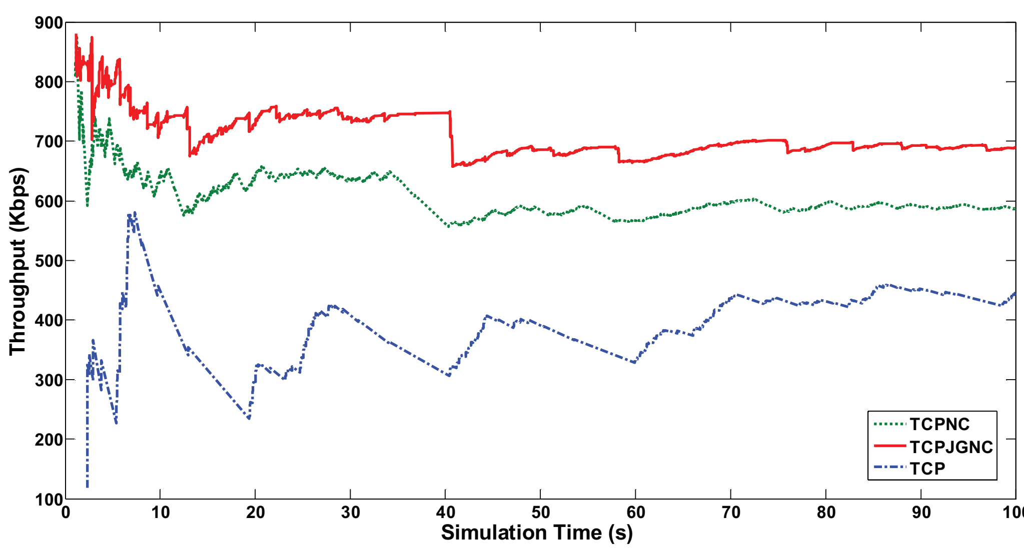

Fig. 10. The throughput of TCP, TCPNC and TCPJGNC vs. simulation time.

First of all, we give the analytical and simulation results for throughput comparison of TCPJGNC under different sensing time. As the sensing time increases, the throughput falls gracefully and the simulation result is close to the analytical, as shown in Fig. 9.

From Fig. 10, we can see that our proposed scheme can significantly improve TCP throughput compared to the other two schemes: TCP and TCPNC. The system needs take some time to reach steady state. The TCPNC and TCPJGNC can fast converge into the steady state, which incorporate network coding into TCP. In the steady state, The TCPJGNC can obtain the highest TCP throughput which can dynamically adjust the number of packets to code according to the network condition. In addition, in TCPJGNC, it modifies the TCP mechanism in spectrum sensing, spectrum changing and PU's activity states.

In the following, we study the behavior of TCPJGNC under the scenarios of (1) spectrum sensing, (2) PU activities, and (3) spectrum changing, which are the key features of CRNs.

Please cite this article as: Y. Qin et al., TCPJGNC: A transport control protocol based on network coding for multi-hop cognitive radio networks, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2015.12.007

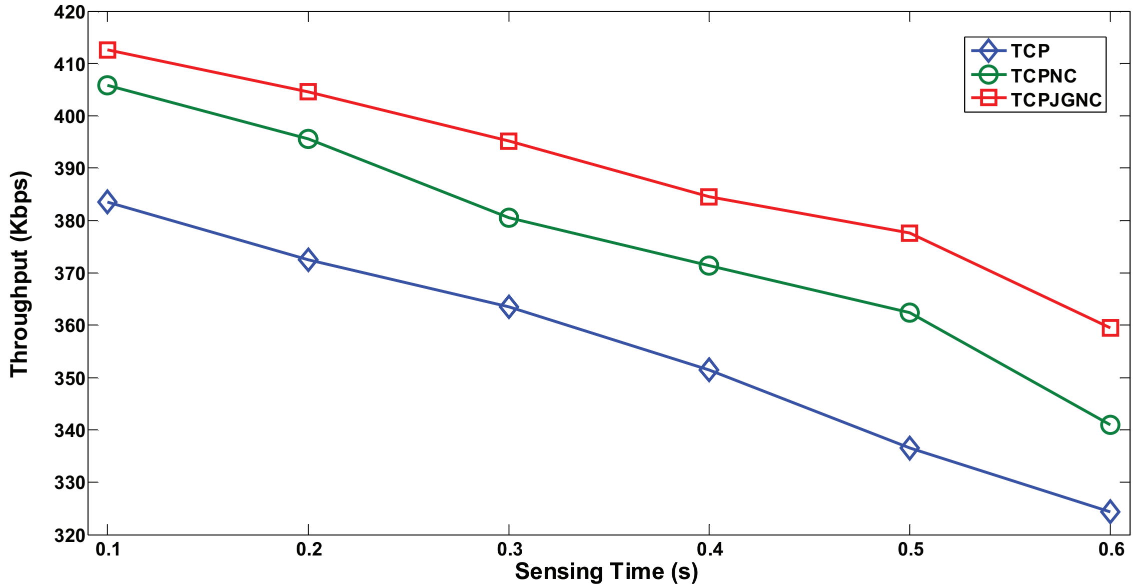

Fig. 11. The throughput of TCP, TCPNC and TCPJGNC vs. sensing time.

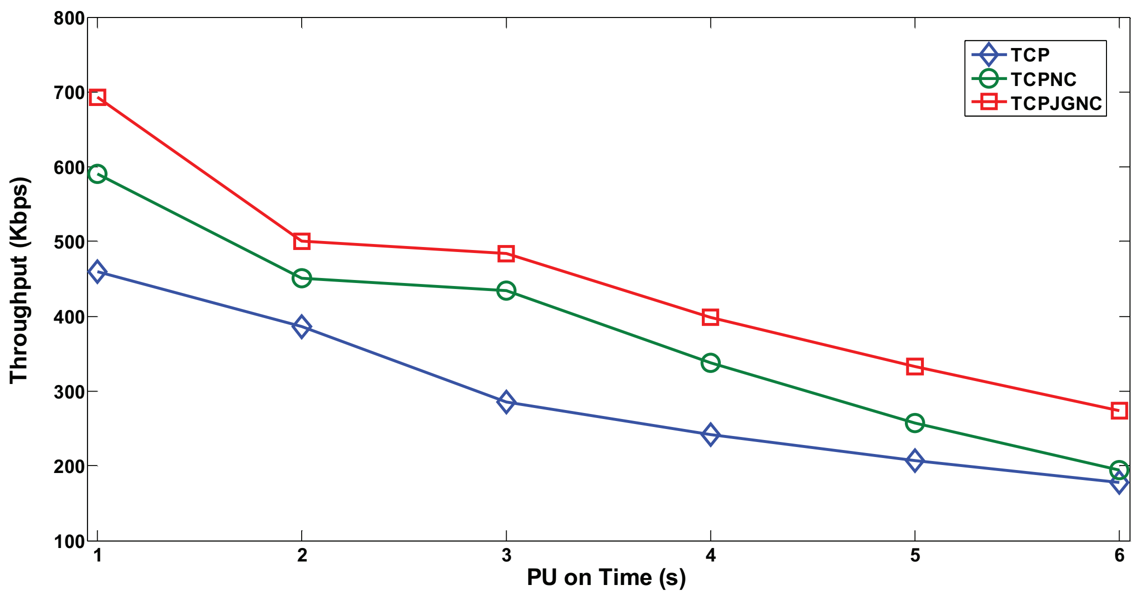

Fig. 12. The throughput of TCP, TCPNC and TCPJGNC vs. PU on time.

6.1. Effects of sensing time on TCP throughput

In Fig. 11, with increasing sensing time, the time for data transmission is reduced, thus, the effective throughput correspondingly decreases. The TCPNC and TCPJGNC can obtain a higher throughput than TCP. This is because that NC could compensate for the lost packets to improve the throughput. We adopt network coding for data transmission, so that most of the lost data can be recovered at the receiver, reducing retransmissions and the probability of collision between the SU and the PU. The TCPJGNC throughput is higher than that of TCPNC, whose source node transmits random linear combination of packets currently in the congestion window. While the source node transmits random linear combination of part of packets currently in the congestion window according the network condition in TCPJGNC scheme. Also, we modify the TCP mechanism in the calculating methods of RTO and effective window which are more efficient in utilizing spectrum during spectrum sensing phase.

6.2. Effects of PU activities on TCP throughput

In Fig. 12, we can see that the throughput among TCP, TCPNC, and TCPJGNC will decrease as the PU on time increases. From Section 2, we recall that when PUs are active, the affected SUs immediately interrupt transmission and then find an available channel to continue to transmit data packets if any, or wait for an opportunity that the PUs currently do not occupy the channels. When the duration of PU occupation is long, the time for SUs to exploit available channels to transmit data packets is small. Thus, the effective throughput will decrease in all three schemes.

In addition, we can see that when the PU on time is large, the performance of TCPNC is better than that of TCP. This is because the source node transmits random linear combination of packets currently in the congestion window in TCPNC. In TCPJGNC, it can dynamically adjust the number of packets which are participating in network coding operation; also, we modify the TCP mechanism

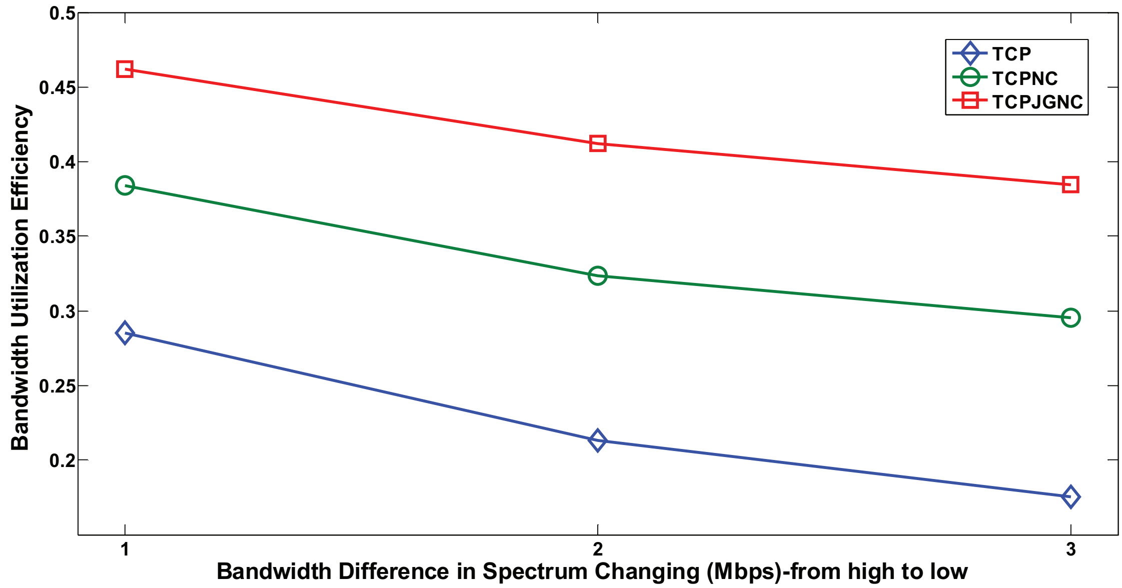

Fig. 13. Bandwidth utilization efficiency vs. difference in spectrum changing (from high to low).

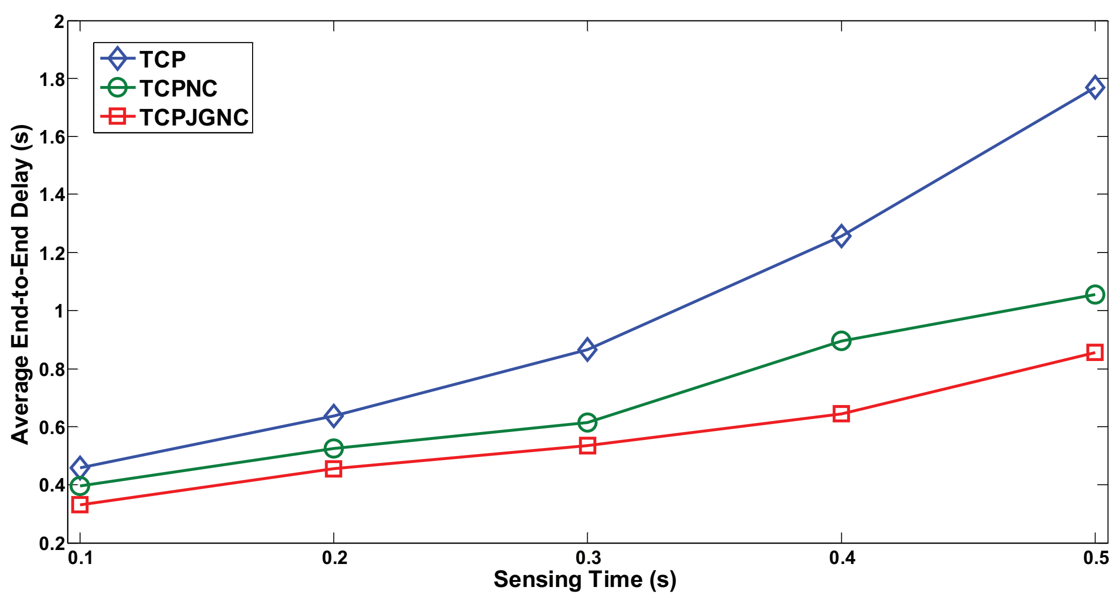

Fig. 14. Average end-to-end delay vs. sensing time.

during the PU's activities. Thus, it still has a higher throughput than TCPNC and TCP when increasing PU on time.

6.3. Effects of spectrum changing on bandwidth utilization efficiency

On PUs arriving, the affected SUs should switch channel to continuously transmit data packets. In this process, the bandwidth may change, which mainly affects the RTT. Thus, in our scheme, we modify the calculating method of RTT. We observe that TCPJGNC has a higher bandwidth efficiency than TCPNC and TCP when three channels, Ch1, Ch2, and Ch3, have varying raw channel bandwidth: 1 Mbps, 2 Mbps and 4 Mbps, respectively. The switch sequence is 2Mbps→1 Mbps, 4Mbps→2 Mbps, 4Mbps→1 Mbps. The differences are 1 Mbps, 2 Mbps, and 3 Mbps. As seen in Fig. 13, it shows that as the bandwidth difference grows, the bandwidth utilization efficiency reduces. However, TCPJGNC performance improves significantly over TCP, implying that our scheme is effective in fully utilizing the spectrum resource.

6.4. Effects of spectrum sensing on end-to-end delay

We compare the average end-to-end delay over TCP, TCPNC and TCPJGNC schemes. In Fig. 14, we can see that the network coding based TCP schemes (TCPNC and TCPJGNC) have a smaller end-to-end delay than that of TCP. This is because network coding can mask loss from the congestion control over lossy networks, and then they have fewer retransmissions than the traditional TCP. However, the TCPJGNC becomes considerably lower. The reason is that it encodes packets according to performing JGNC condition, which also considers ACKs loss in CRNs. Moreover, in TCPJGNC design, we consider the features of CRNs and modify the TCP. Hence, the TCPJGNC has fewer retransmissions than the classical TCP and TCPNC. Also, we can see that as sensing time increases, the end-to-end delay increases as well. This is because with increasing sensing time, the time for data transmission is reduced, thus, it takes more time for data transmission over multi-hop CRNs.

Please cite this article as: Y. Qin et al., TCPJGNC: A transport control protocol based on network coding for multi-hop cognitive radio networks, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2015.12.007

7. Conclusion

In this paper, we propose a novel transport protocol TCPJGNC for multi-hop CRNs from a network coding perspective. We jointly consider network coding and modifications in TCP mechanism for CRNs to improve the TCP performance. We first analyze the decoding probability of JGNC over multi-hop CRNs, which considers packet loss rate, PU activity, and the probability of linearly independent. Then, we incorporate JGNC into TCP and modify the TCP from the following four aspects: PU behavior, spectrum sensing, spectrum changing, and TCP itself, for multi-hop CRNs. Simulation results were presented to illustrate the effectiveness of TCPJGNC. It is shown that TCP throughput, bandwidth utilization efficiency and average end-to end delay can be improved significantly by modifying traditional TCP mechanism and using network coding technology. Network layer and MAC layer will be considered for TCP performance optimization of CRNs in our future work.

Acknowledgments

This work was supported by the Science and Technology Fundament Research Fund of Shenzhen under grant JCYJ20140417172417131 and the Natural Science Foundation of Guangdong Province under grant 2014A030313698. We would like to acknowledge the editor and the reviewers whose comments and suggestions significantly improved this paper.

References

[1] FCC, Et docket no 03-222 notice of proposed rulemaking and order, Federal Communications Commission, Tech. Rep., December 2003.
[2] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, S. Mohanty, Next generation dynamic spectrum access cognitive radio wireless networks: a survey, Elsevier Comput. Netw. 50 (13) (2006) 2127–2159.
[3] J. Mitola III, Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio, Ph.D Thesis, KTH Royal Institute of Technology, 2000.
[4] C. Cormio, K.R. Chowdhury, A survey on MAC protocols for cognitive radio networks, Elsevier Ad Hoc Netw. 7 (7) (2009) 1315–1329.
[5] J. Wu, Y. Dai, Y. Zhao, Effective channel assignments in cognitive radio networks, Elsevier Comput. Commun. 36 (4) (2013) 411–420.
[6] G. Cheng, W. Liu, Y. Li, W. Cheng, Joint on-demand routing and spectrum assignment in cognitive radio networks, in: Proceedings of IEEE ICC, 2007, pp. 6499–6503.
[7] K.C. How, M. Ma, Y. Qin, Routing and QoS provisioning in cognitive radio networks, Elsevier Comput. Netw. 55 (1) (2011) 330–342.
[8] M. Caleffi, I.F. Akyildiz, L. Paura, OPERA: optimal routing metric for cognitive radio ad hoc networks, IEEE Trans. Wirel. Commun. 11 (8) (2012) 2884–2894.
[9] X. Zhong, Y. Qin, Y. Yang, L. Li, CROR: Coding-aware opportunistic routing in multi-channel cognitive radio networks, in: Proceedings of IEEE GLOBECOM, 2014, pp. 100–105.
[10] Y. Qin, X. Zhong, Y. Yang, Y. Li, L. Li, Joint channel assignment and opportunistic routing for maximizing throughput in cognitive radio networks, in: Proceedings of IEEE GLOBECOM, 2014, pp. 4592–4597.
[11] A.M. Slingerland, P. Pawelczak, R. Prasad, A. Lo, R. Hekmat, Performance of transport control protocol over dynamic spectrum access links, in: Proceedings of IEEE DySPAN, 2007, pp. 486–495.
[12] T. Issariyakul, L.S. Pillutla, V. Krishnamurthy, Tuning radio resource in an overlay cognitive radio network for TCP: greed isn't good, IEEE Commun. Mag. 47 (7) (2009) 57–63.
[13] C. Luo, F.R. Yu, H. Ji, V.C.M. Leung, Cross-layer design for TCP performance improvement in cognitive radio networks, IEEE Trans. Veh. Technol. 59 (5) (2010) 2485–2495.
[14] C. Luo, F. Yu, H. Ji, V. Leung, Optimal channel access for TCP performance improvement in cognitive radio networks, Springer Wirel. Netw. 17 (2) (2011) 479–492.
[15] J. Wang, A. Huang, W. Wang, TCP throughput enhancement for cognitive radio networks through lower-layer configurations, in: Proceedings of IEEE PIMRC, 2012, pp. 1424–1429.
[16] D. Sarkar, H. Narayan, Transport layer protocols for cognitive networks, in: Proceedings of IEEE INFOCOM Workshops, 2010, pp. 1–6.
[17] M.D. Felice, K.R. Chowdhury, L. Bononi, Modeling and performance evaluation of transmission control protocol over cognitive radio ad hoc networks, in: Proceedings of ACM MSWiM, 2009, pp. 4–12.
[18] K.R. Chowdhury, M.D. Felice, I.F. Akyildiz, TP-CRAHN: a transport protocol for cognitive radio ad-hoc networks, in: Proceedings of IEEE INFOCOM, 2009, pp. 2482–2490.
[19] K.R. Chowdhury, M.D. Felice, I.F. Akyildiz, TCP CRAHN: a transport control protocol for cognitive radio ad hoc networks, IEEE Trans. Mobile Comput. 12 (4) (2013) 790–803.
[20] S. Floyd, B. Henderson, RFC 2582: the NewReno modifications to TCP's fast recovery algorithm, Internet Eng. Task Force (IETF) (April, 1999).
[21] L.S. Brakmo, L.L. Paterson, TCP Vegas: end to end congestion avoidance on a global Internet, IEEE J. Select. Areas Commun. 13 (8) (1995) 1465–1480.
[22] V. Jacobson, Congestion avoidance and control, Comput. Commun. Rev. 18 (4) (1988) 314–329.
[23] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, R. Wang, TCP westwood: end to end congestion control for wired/wireless networks, Springer Wirel. Netw. 8 (5) (2002) 467–479.
[24] R. Ahlswede, N. Cai, S.-Y.R. Li, R.W. Yeung, Network information flow, IEEE Trans. Inf. Theory 46 (4) (2000) 1204–1216.
[25] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, J. Crowcroft, XORs in the air: practical wireless network coding, in: Proceedings of ACM SIGCOMM, 2006, pp. 243–254.
[26] Y. Qin, X.T. Xu, Y.Y. Yang, J.L. Zhou, H.P. Wang, Joint generation network coding in unreliable wireless networks, in: Proceedings of IEEE GLOBECOM, 2011, pp. 1–5.
[27] J.K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, J. Barros, Network coding meets TCP: theory and implementation, in: Proceedings of the IEEE, 99, 2011, pp. 490–512.
[28] A. Goldsmith, S.A. Jafar, I. Maric, S. Srinivasa, Breaking spectrum gridlock with cognitive radios: an information theoretic perspective, in: Proceedings of the IEEE, 97, 2009, pp. 894–914.
[29] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, IEEE Trans. Inf. Theory 52 (10) (2006) 4413–4430.
[30] D. Wang, Q. Zhang, J.C. Liu, Partial network coding: theory and application for continuous sensor data collection, in: Proceedings of IEEE IWQoS, 2006, pp. 93–101.
[31] C. Samios, M.K. Vernon, Modeling the throughput of TCP Vegas, in: Proceedings of ACM SIGMETRICS, 2003, pp. 71–81.
[32] N. Parvez, A. Mahanti, C. Williamson, An analytic throughput model for TCP NewReno, IEEE/ACM Trans. Netw. 18 (2) (2010) 448–461.
[33] Network Simulator (ns2). [Online]. Available at: http://www.isi.edu/nsnam/ns/, 2012.
[34] Michigan Tech. Univ., Cognitive radio cognitive network simulator, available at http://stuweb.ee.mtu.edu/ljialian/index.htm, 2013.

Please cite this article as: Y. Qin et al., TCPJGNC: A transport control protocol based on network coding for multi-hop cognitive radio networks, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2015.12.007