Migrating Unfairness Among Subflows in MPTCP With Network Coding for Wired–Wireless Networks

Kaiping Xue, *Senior Member, IEEE*, Jiangping Han, Hong Zhang, Ke Chen, and Peilin Hong

***Abstract*—Recently, two new technologies have been introduced into the transport layer. One is network coding, and the other is multipath transmission control protocol (MPTCP). Network coding is introduced into the transport layer to enhance the performance of transmission control protocol (TCP) in wireless networks. Benefiting from multi-interface devices, MPTCP is proposed to make full use of the network resource. Theoretically, combining these two technologies can utilize resources more adequately. However, network coding and multipath transportation cannot collaborate well with each other because network coding invalidates the load-balancing feature of MPTCP congestion control schemes. In this paper, we first discuss the unfair congestion control issue in MPTCP combined with network coding (MPTCP/NC). Then, a new end-to-end congestion control solution, named Couple+, is presented to deal with the unfairness among subflows. In Couple+, sender tries to slightly slow down sending rate if the reason of packet loss is not decided. After judging of packet loss reason based on the characteristics of packet loss events, the rate will be recovered soon if the loss is caused by wireless error (wireless noise or collision) or be further reduced if the loss is caused by congestion. By simulation, we compare the performances of Couple+ and the previous congestion control scheme of MPTCP. The performance analysis proves that unfairness among subflows indeed exists, and our scheme can balance congestion among coded and noncoded subflows and can stay friendly with TCP flow.**

***Index Terms*—Congestion control, fairness, load balance, multipath transmission control protocol (MPTCP), network coding.**

I. Introduction

TAKING advantage of multi-interface devices and various wireless communication technologies, the multipath transmission control protocol (MPTCP) splits data transmission into several paths concurrently [1]. The principles for MPTCP congestion control schemes are "Do No Harm" and "Balance Congestion." "Do No Harm" means MPTCP should be transmission control protocol (TCP) friendly. "Balance Congestion" means MPTCP should utilize the least congested path [1]–[3]. Several congestion control schemes have been proposed to achieve these two principles, such as equally weighted TCP (EWTCP), COUPLED, SEMI-COUPLED, and round-trip-time (RTT) Compensator [2]–[4] and dynamic window coupling [5]. All of these schemes are packet loss based, and they do not modify slow-start, fast-retransmission, and fast-recovery phases of the most commonly adopted congestion control schemes. The main enhancement is that the behaviors of the congestion window on different paths are not independent but coupled to satisfy the requirement of "Do No Harm" and "Balance Congestion." Delay-based congestion control schemes are also effective, such as weighted Vegas (wVegas) [6].

The characteristics of shared wireless media and the unpredictability of the wireless channel lead to link error-based packet loss (wireless loss) [7]–[9]. Network coding is introduced into transport protocol to prevent wireless loss from influencing the accuracy of congestion control decision [10]. Therefore, combining subflows of MPTCP with network coding (MPTCP/NC) can hide wireless loss [11] in wireless network. Carrying network coding in the connection level of MPTCP can simplify the packet scheduling scheme [12]–[15]. Online coding [10] is a simple implementation of TCP with network coding in the subflow level of MPTCP. In this mechanism, the receiver does not acknowledge the decoded segments but sends back acknowledgment based on the amount of independent information (i.e., the degree of freedom of the decoding matrix) in the form of "seen." Every time a receiver gets one unit of new information expected, it acknowledges the new information it "seen."

Apart from the advantages of network coding introduced into MPTCP, it makes the current congestion control scheme for MPTCP deviate from the "Do No Harm" and "Balance Congestion" principles. If the congestion control scheme is packet loss based, network coding hides part of the congestion loss (loss caused by congestion) and sender acts slower to congestion [16], [17]. If the scheme is delay based, the effectiveness is based on the parameters set for the scheme, and sometimes, the congestion control decision still needs to depend on packet loss [18]. The coded subflow becomes greedier when the coding factors, i.e., redundancy factor and coding window size, are higher than the required level. Therefore, the congestion degree in the coded path exposed by the sender is lower than the real degree. On one hand, the load cannot be migrated to a less congested path, which deviates from "Balance Congestion." On the other hand, the coded subflow may occupy the bandwidth released by the noncoded TCP flow competing for wired bottleneck, which deviates from "Do No Harm."

Manuscript received June 6, 2015; revised November 15, 2015 and January 30, 2016; accepted March 14, 2016. Date of publication March 18, 2016; date of current version January 13, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61379129 and Grant 61390513; by the National High-Tech Research and Development Plan of China (863 Program) under Grant 2014AA01A706; and by the Fundamental Research Funds for the Central Universities and the Chinese Academy of Sciences Youth Innovation Promotion Association. The review of this paper was coordinated by Dr. X. Huang.

The authors are with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China (e-mail: kpxue@ustc.edu.cn; jphang@mail.ustc.edu.cn; zhh006@mail.ustc.edu.cn; chenke13@mail.ustc.edu.cn; plhong@ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2016.2543842

Furthermore, an unreasonable congestion control scheme may lead to incorrect estimation of path state, which will further interfere with path selection scheme. This is because, when user moves away, the link state decreases and the throughput is low in the corresponding path. Path selection scheme always abandons the worst path [19], [20] to improve total throughput and save energy according to the performance of the flow, such as throughput.

To recover the character of "Do No Harm" and "Balance Congestion," several innovative works have been done in this paper. 1) To prove the necessity of Couple+, the unfairness among subflows of MPTCP/NC is highlighted and verified with dual decomposition theory [21] and simulations. 2) We propose a new congestion control scheme for MPTCP/NC named Couple+ to expose congestion and deal with congestion timely. In Couple+, the receiver is responsible for notifying packet loss events, and the sender is responsible for justifying the reason of packet loss. The key point in Couple+ is based on the coding parameters and the limitation of receiving buffer in the connection level. 3) To show the effectiveness of Couple+, we analyze the characteristics of wireless loss in a real network and the performance of Couple+ in the ns-3 simulator. The results demonstrate that Couple+ can deal with the congestion in coded path effectively and avoids damaging the performance of the congestion-sensitive flow in the bottleneck link.

The remainder of this paper is structured as follows. In Section II, we present a literature review of some congestion control schemes used in MPTCP and the basic knowledge of MPTCP/NC. Section III provides a proof about the invalidation of "Do No Harm" and "Balance Congestion" in previous schemes if being used for MPTCP/NC. The detail of the newly proposed Couple+ is given in Section IV. Simulation results and performance comparisons are given in Section V. Section VI concludes this paper.

II. Related Works

Here, we focus on some previous congestion control schemes for MPTCP and how MPTCP is combined with network coding. These works are tightly related to our work.

A. MPTCP Congestion Control Scheme

To achieve "resource pooling," MPTCP flow uses fragmented link resource and always allocates more traffic to the least congested path in the available paths. If all the available paths are not congested at all, MPTCP is allowed to increase throughput as high as possible in all the available paths. Loss-based congestion control scheme for MPTCP has been adopted by the Internet Engineering Task Force as RFC6356 [22]. Typical loss-based congestion control schemes for MPTCP include EWTCP, COUPLED, SEMI-COUPLED, and RTT Compensator [2]–[4], [22]. They are all modified from congestion control schemes for single-path TCP flow.

Assume that n is the number of subflows, w_t is the current congestion window size on path t, and w_{total} is the sum of congestion window size of all the paths in an MPTCP flow. The modification of the congestion window in different schemes is shown in Table I. The principle of EWTCP [23] is that MPTCP flow gets the same throughput as a regular TCP. Furthermore, considering the load diversity on different paths, COUPLED [24], [25] aims at shifting traffic onto the least congested path from more congested paths; hence, congestion loss rate across the whole network will tend to be balanced. COUPLED may abandon the most congested path in some circumstances. On the contrary, SEMI-COUPLED [4] always keeps a moderate amount of traffic on each path while still preferring to the less congested paths. These schemes all assumed that RTTs of all the subflows are the same. RTT Compensator [26] upgrades over SEMI-COUPLED, which takes RTTs of subflows into consideration. In RTT Compensator, the increment of congestion window is up limited by $1/w_t$, which ensures that the multipath flow can take no more capacity than a single-path TCP through the common bottleneck.

RTT Compensator is the most thoughtful congestion control scheme for MPTCP. There are two design principles of RTT Compensator.

- An MPTCP flow should give a connection at least as much throughput as it would get with single-path TCP on the best of its paths.
- A multipath flow should take no more capacity on any path or collection of paths than if it was a single-path TCP flow using the best of those paths.

Let t be the symbol of the single path, T be the available path set, and U be the subset of T, which share a bottleneck. These principles can be expressed as follows:

$$\sum_{t\in T}\frac{\hat{w}_t}{\text{RTT}_t}\geq\max_{t\in T}\frac{\hat{w}_t^{\text{TCP}}}{\text{RTT}_t} \tag{1}$$

$$\sum_{t\in U}\frac{\hat{w}_t}{\text{RTT}_t}\leq\max_{t\in U}\frac{\hat{w}_t^{\text{TCP}}}{\text{RTT}_t}, U\subseteq T \tag{2}$$

where RTT_t is the round-trip time on path t, $\hat{w}_t$ is the equilibrium congestion window size on path t, $\hat{w}_t^{\text{TCP}}$ is the equilibrium window that would be obtained by a single-path TCP experiencing path t's loss rate. For any flow with additive increase–multiplicative decrease (AIMD), which is the default congestion window adjustment scheme for MPTCP, the control style of AIMD is that the increases and decreases of the congestion window must be balanced. Window w_t increases on the receiving of ACKs and decreases on the happening of packet loss. This concept can be expressed in (3), shown below, where p_t is the congestion loss rate on path t

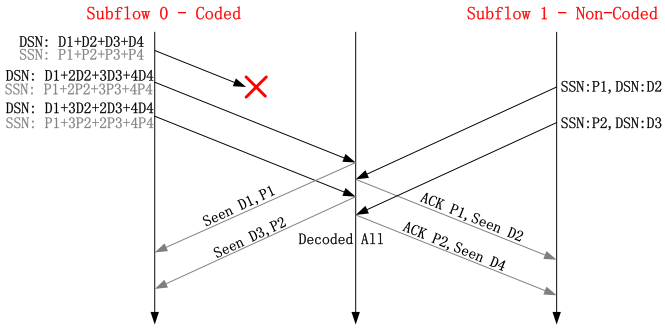
$$(1-p_t)\min\left(\frac{a}{\hat{w}_{\text{total}}},\frac{1}{\hat{w}_t}\right)=p_t\frac{\hat{w}_t}{2} \tag{3}$$

where parameter a controls the aggressiveness, and $\hat{w}_{\text{total}}$ is the sum of the equilibrium congestion window size of all the paths in an MPTCP flow. Making the approximation that p_t is small enough, thus $1-p_t\approx 1$, and (3) can be simplified as $\hat{w}_t^{\text{TCP}}=\sqrt{2/p_t}$. By combining with (1), (2), and $\hat{w}_t^{\text{TCP}}=\sqrt{2/p_t}$, finally, the increment pace can be computed, as shown in Table I.

The most important part of RTT Compensator is that the bandwidth occupied by subflows of the MPTCP flow in the specific bottleneck is limited by a single-path TCP flow in the same bottleneck link.

TABLE I
CONGESTION CONTROL SCHEME FOR MPTCP

Congestion control scheme	congestion control	cwnd increment	cwnd decrement
EWTCP [23]	AIMD	$\Delta w_t = a/w_t$, $a = 1/\sqrt{n}$	$w_t/2$
COUPLED [24], [25]	AIMD	$\Delta w_t = 1/w_{total}$	$w_{total}/2$
SEMI-COUPLED [4]	AIMD	$\Delta w_t = a/w_{total}$, a is a constant	$w_t/2$
RTT-Compensator [26]	AIMD	$\Delta w_t = \min(a/w_{total}, 1/w_t)$, $a = \hat{w}_{total}\frac{\max_t \hat{w}_t/RTT_t^2}{(\sum_t \hat{w}_t/RTT_t)^2}$	$w_t/2$



DSN:Data Sequence Number; SSN: Subflow Sequence Number

Fig. 1. How MPTCP/NC works.

B. MPTCP/NC

MPTCP/NC can be classified into three categories: 1) implementing network coding at the connection level; 2) implementing network coding at the subflow level; and 3) implementing network coding at both the connection and subflow levels of MPTCP. The first category can be also treated as a special issue of the second category by choosing all the same coding parameters for different subflows, respectively. Hence, we choose the second category as an example to explain how MPTCP/NC works.

The key idea of MPTCP/NC is the mixed use of regular subflows and network coded subflows, as shown in Fig. 1. In Fig. 1, there are wireless links on the path of Subflow 0; hence, Subflow 0 is a coded subflow. Subflow 1 is a noncoded subflow. MPTCP uses data sequence number (DSN) in the connection level and subflow sequence number (SSN) in the subflow level to ensure data order in both levels. In Fig. 1, Dx $(x = 1, 2, \ldots)$ refers to the packet in the connection level with DSN Dx, whereas Px $(x = 1, 2, \ldots)$ refers to the packet in the subflow level with SSN Px. Packets are coded into segments in the subflow level. We assume that three coded segments and two original segments are sent out through Subflow 0 and Subflow 1, respectively. Assume that one of the three packets transmitted through Subflow 0 is lost due to wireless transmission error. In traditional MPTCP, the packet loss event not only causes congestion misjudgment because of duplicated ACK but incurs head-of-line block [27] at the connection level because of the long delay of retransmission as well. However, if adopting network coding, both issues can be solved. As shown in Fig. 1, after the packets are lost, the receiver can still send back nonduplicated ACK because it can obtain information of new segments from received coded packets. Finally, segments D1–D4 will be received at the receiver side without triggering nonnecessary retransmission and head-of-line block. Therefore, based on the principles of network coding, the sender will not trigger an unnecessary congestion control procedure. The receiver also does not need to deal with blocking in the receiving buffer caused by long retransmission delay by utilizing redundant data to avoid explicit retransmission that needs a longer delay.

The two most important factors of TCP/MPTCP/NC are the coding window size and redundancy. The coding window size limits the number of original segments contained in a coded packet. The redundancy limits the number of redundant packets to be sent to cover potential wireless loss. Due to the heterogeneity of paths, coding parameters can be independently chosen for each path. Larger coding window and redundancy will lead to stronger resistance to packet loss. However, based on the influence of coding parameters, irrational congestion control schemes may cause imbalance of load among subflows and damage the performance of other congestion sensitive flows, which will be proved in the subsequent sections.

III. INVALIDATION OF "DO NO HARM" AND "BALANCE CONGESTION" AMONG SUBFLOWS

Here, the invalidation of "Do No Harm" and "Balance Congestion" when MPTCP combined with network coding under the congestion condition is discussed. Without loss of generality, the coding type here is online coding [10], in which the number of segments participating in coding is limited by the sliding coding window. We assume that online coding is implemented in the subflow level. We first prove the existence of packet loss detection delay, which makes the sender of coded flows less sensitive to congestion. Furthermore, the invalidation of "Do No Harm" and "Balance Congestion" is proved via dual decomposition theory [21] and the related simulation on the performance of the improper congestion control scheme is given in Section V.

A. Packet Loss Detection Delay

With network coding, although a packet loses due to congestion, receiver will send back nonduplicated acknowledgment if a new information unit is obtained in newly received coded packet. Hence, it hides part of congestion loss. It is reasonable if the loss is caused by wireless errors; however, if it is congestion based, it causes unfairness.

Fig. 2 shows the congestion window behavior of the TCP flow and TCP/NC flow. Assume that the path quality (i.e., congestion loss rate and RTT) is identical for the two flows. The relationship between the loss detection delay and the loss rate is denoted by (4), show below, where p_c is the average congestion loss rate in TCP flows and $p_{\rm nc}$ is the average congestion loss rate in network coding flows. Referring to [17], without regarding to the integer constraint, the delay is

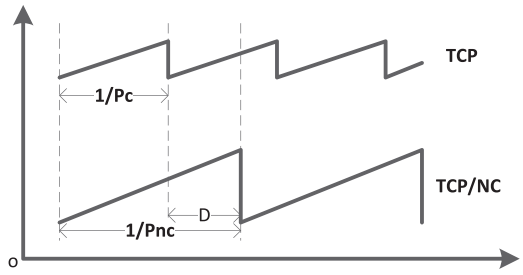


Fig. 2. Delay in detection of packet loss.

represented approximately in (5), shown below, where W is the coding window size, R is the redundancy parameter, and $P_{\max}$ is the maximum value of congestion loss rate in the congestion period. Obviously, the sender of network coding flows senses less loss rate as in (6). Network coding is mainly used to cover potential wireless loss, but it may also cover part of the congestion loss that should have to be completely exposed. Hence, the congestion loss rate sensed by the sender is less than the real one. The sender in MPTCP always migrates load to less congested paths, and the path quality is tightly related to the average congestion loss rate and RTT [3], [4], [22]. Misconception of the congestion level leads to misjudgment of path quality, which will eventually lead to the imbalance of load among subflows of MPTCP

$$\frac{1}{p_{\rm nc}} = \frac{1}{p_c} + D \tag{4}$$

$$D = \frac{WR}{1 - R(1 - p_{\max})} \tag{5}$$

$$\begin{aligned} p_{\rm nc} &= \frac{(1 - R(1 - p_{\max}))\, p_c}{(1 - R(1 - p_{\max})) + WRp_c} \\ &< \frac{(1 - R(1 - p_{\max}))\, p_c}{1 - R(1 - p_{\max})} = p_c. \end{aligned} \tag{6}$$

B. Analysis With Dual Decomposition Theory

The congestion control scheme must ensure that the total utility of flows in the network is maximized [6], [21]. The related parameters are given in Table II.

The parameters in Table II are used for the following analysis. There are three main concepts in Table II, namely, link, path, and flow. Link refers to the point-to-point physical link. Path refers to the end-to-end routing path, which is made up of one or more links in series. Flow is the path set of TCP or MPTCP. TCP flow has only one path, whereas MPTCP flow with multiple subflows has more than one path.

As shown below in (7), to maximize $\sum_{s\in S} U_s(y_s)$, the most important constraint is $\sum_{t\in T_s} a_{\rm lt} \cdot \sum_{s\in S} x_{\rm st} r_{\rm st} w_{\rm st} \le c_l$, which means that the total rate is constrained by the link capacity, and the effects of coding window size and redundancy parameter need to be considered. For instance, the coding window is 4 and the redundancy is 1.25; thus, every packet is transmitted at most $4 * 1.25$ times, such as original segments $(p1, p2, p3, p4, \ldots)$

TABLE II
DEFINITION OF VARIABLES IN ANALYSIS

variables	meaning
L	link set
S	flow set including multipath flows and single path flows in the network
C	finite capacities of link set
T	path set, where T_s is the Path set of flow s
A	the relation between the path set T and the link set L, where $a_{lt} = 1$ if link l is on path t
X	the flow rate matrix, where x_{st} is the rate of flow s path on t
R	redundancy matrix, where r_{st} is the redundancy parameter of flow s on path t
W	the coding window size, where w_{st} is the coding window size of flow s on path t
P	the link loss rate, where p_{st} is the link loss rate of flow s on path t and r_{st} must be no less than $\frac{1}{1-p_{st}}$ to provide enough redundancy
Y	the rate of flow, $y_s \in Y$ and $y_s = \sum_{r\in R_s} x_{st} r_{st}$
$U_s(y_s)$	the utility of flow s, $U_s(0) = -\infty$, where $U_s(\cdot)$ is increasing, twice-differentiable and strictly concave function

may be transmitted in forms of $p1, p1 + p2, p1 + p2 + p3$, $p1 + p2 + p3 + p4, p1 + p2 + p3 + 2p4, \ldots$. Each segment is conveyed in more than one coded packet

$$\begin{aligned} \max_X \quad & \sum_{s\in S} U_s(y_s) \\ \text{s.t.} \quad & x_{\rm st} \ge 0, \quad s \in S,\ t \in T_s \\ & \sum_{t\in T_s} a_{\rm lt} \cdot \sum_{s\in S} x_{\rm st} r_{\rm st} w_{\rm st} \le c_l. \end{aligned} \tag{7}$$

The corresponding Lagrangian function is given in

$$\begin{aligned} L(X, \lambda) &= \sum_{s\in S} U_s(y_s) - \sum_{l\in L} \lambda_l \left(\sum_{s\in S, t\in T_s} a_{\rm lt} x_{\rm st} r_{\rm st} w_{\rm st} - c_l \right) \\ &= \sum_{s\in S} U_s(y_s) - \sum_{s\in S, t\in T_s} \left(\sum_{l\in L_s} \lambda_l a_{\rm lt} \right) x_{\rm st} r_{\rm st} w_{\rm st} \\ &\quad - \sum_{l\in L} \lambda_l c_l. \end{aligned} \tag{8}$$

Assume that $q_{\rm st} = \sum_{l\in L_s} \lambda_l a_{\rm lt}$ and $E = \sum_{l\in L} \lambda_l c_l$, where λ_l is the Lagrange multiplier, e.g., link price, associated with the linear flow constraint on link l, and $q_{\rm st}$ can be seen as the aggregate path congestion price of those links used by flow s on path t. Equation (8) can be further simplified as follows:

$$L(X, \lambda) = \sum_{s\in S} \left(U_s(y_s) - \sum_{t\in T_s} (q_{\rm st} x_{\rm st} r_{\rm st} w_{\rm st}) \right) + E. \tag{9}$$

The goal in this dual problem is to get the optimal X and λ. According to dual decomposition theory [21], the original

optimization can be decomposed into a master problem in (10), shown below, and several subproblems, as shown below in (11). In the subproblem, for any flow, the rate is adjusted according to local congestion signal

$$D(X^*, \lambda^*) = \min_{\lambda \geq 0} \left(\sum_{s \in S} L_s(\lambda) + E \right) \tag{10}$$

$$L_s(\lambda) = \max_{X \geq 0} \left(U_s(y_s) - \sum_{t \in T_s} q_{\mathrm{st}} x_{\mathrm{st}} r_{\mathrm{st}} w_{\mathrm{st}} \right). \tag{11}$$

When will the congestion control procedure end? Referring to the "Congestion Equality Principle" mentioned in [6], supposing flow s has n $(n > 0)$ paths, given $\lambda_l \geq 0$, the corresponding congestion prices k_{si} $(i = 1, 2, \ldots, n)$ are sorted in ascending order: $k_{s1} = \cdots = k_{\mathrm{sm}} < k_{s(m+1)} \leq \cdots \leq k_{\mathrm{sn}}$. Then the optimal solution $x^*_{\mathrm{st}}(\lambda)$ for the sub problem satisfies (12). $U_s{}'(\cdot)$ is the derivative of $U_s(\cdot)$

$$\begin{aligned} & U_s'(y_s) - k_{s1} = 0 \\ & x^*_{\mathrm{st}} \neq 0, t = 1, 2, \ldots, m \\ & x^*_{\mathrm{st}} = 0, t = m+1, m+2, \ldots, n. \end{aligned} \tag{12}$$

The principle means that, to achieve "Balance congestion," each subflow adjusts its congestion window according to the congestion price sensed by the sender. In general, load balancing is achieved if and only if congestion price on each chosen path is equal, i.e., $\forall\, i, j \in T_s,\ i \neq j,\ k_{\mathrm{si}} = k_{\mathrm{sj}} = U'$. Since all the subflows are coupled, the load can be migrated to a less-congested path. All subflows adjust their rate until they feel that the qualities of all paths are roughly equal. Previous congestion control schemes are not fit for the scenario of MPTCP/NC because they do not take coding factors into consideration. In previous congestion control schemes such as RTT Compensator, they only take q_{st} as the congestion price of path t sensed by the sender, such as packet loss or longer RTT, while the true congestion price is $q_{\mathrm{st}} r_{\mathrm{st}} w_{\mathrm{st}}$ according to the model. Sender needs to consider the coding factors because network coding may insert a few more data into the network and make the sender sense a lower congestion degree on the path.

The congestion price sensed by the sender must be based on both coding parameter and real congestion price if network coding exists. In previous schemes, due to the ignorance of coding factors, the sensed congestion level will not reflect the real congestion level correctly. The larger the coding parameter is, the less sensitive the sender reacts to congestion. Without loss of generality, for two paths i and j, if $r_{\mathrm{si}} w_{\mathrm{si}} > r_{\mathrm{sj}} w_{\mathrm{sj}}$, when the congestion control is completed, i.e., $q_{\mathrm{si}} = q_{\mathrm{sj}}$, the real congestion price results in $k_{\mathrm{si}} > k_{\mathrm{sj}}$, which deviates from the guidance given in "Congestion Equality Principle." Subflows with larger coding parameters are less sensitive to congestion. Therefore, although network coding flow and noncoding flow sense the same congestion-based packet loss rate, network coding flow will be more congested because part of the congestion loss is hidden by network coding.

In the aforementioned verification, the problem is assumed to be concave. In the following, concavity is proved. The rate of the subflow can be represented by $x_{\mathrm{st}} = \mathrm{cwnd}_{\mathrm{st}}/\mathrm{RTT}_{\mathrm{st}}$, and the congestion window size is related to the path quality: $\mathrm{cwnd}_{\mathrm{st}} = \sqrt{2/(p_c)_{\mathrm{st}}}$, where $(p_c)_{\mathrm{st}}$ is the congestion loss rate of flow s on path t. The rate of the MPTCP flow can be represented as

$$\begin{aligned} y_s &= \sum_{t \in T_s} x_{\mathrm{st}} = \sum_{t \in T_s} \frac{\mathrm{cwnd}_{\mathrm{st}}}{\mathrm{RTT}_{\mathrm{st}}} \\ &= \sum_{t \in T_s} \sqrt{\frac{2}{(p_c)_{\mathrm{st}}}} \cdot \frac{1}{\mathrm{RTT}_{\mathrm{st}}}. \end{aligned} \tag{13}$$

Let $k_{\mathrm{st}} = (\sqrt{(p_c)_{\mathrm{st}}/2} \cdot \mathrm{RTT}_{\mathrm{st}})$ be the congestion price sensed by the sender. Let the relation between congestion price and total rate be $y_s = \sum_{t \in T_s} (1/k_{\mathrm{st}}) = n_s/k_{s1}$. Given $U'(y_s) = k_{s1}$, the utility function is $U(y_s) = n_s \log(y_s)$, which is an increasing, twice-differentiable, and strictly convex function. Considering that the constraint set is also concave, this is a concave optimization problem.

Therefore, network coding will lead to unfair load allocation among subflows if they are configured with different coding parameters. Network coded subflow may also damage the performance of the congestion-sensitive flows. It is obvious that previous congestion control schemes cannot keep both "Balance Load" and "Do No Harm." The aggressiveness of the coded subflow is under the influence of the competition between the overhead and network coding gain.

IV. Couple+

To overcome the failure of "Do No Harm" and "Balance Congestion," we design a new scheme fit for MPTCP/NC. We simply call it Couple+, which is modified from RTT Compensator [4]. Without loss of generality, online coding is also taken as the implementation of network coding in the subflow level. Meanwhile, acknowledgment mechanism is carried in the subflow level to locate the congested subflows. The first and most important thing here is how to expose congestion. The startpoints of congestion exposure schemes are the influence of network coding factors and the limitation of receiving buffer at the connection level. Using Couple+, it can expose congestion and retransmit lost packet caused by congestion timely to avoid long decoding delay and large occupation of receiving buffer in the connection level of MPTCP. Couple+ contains two main steps, which are both implemented in the subflow level. It first records packet loss information and then turns into "Transient state," during which the detailed reason is analyzed. Three new concepts are raised in Couple+.

- Transient state: In the transient state, the actual reason for packet loss is not clear, and further analysis about the loss is carried on.
- Transient window (twnd): This is used to limit the flow rate in the transient state.
- Redundant cycle: This starts from the first nonredundant packet and ends at another redundant packet. Its duration is related to the value of redundancy.

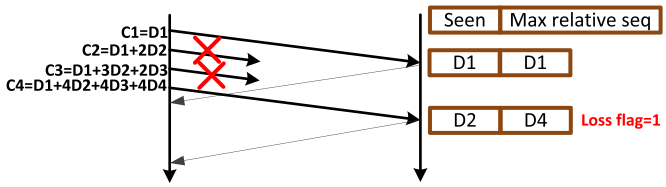


Fig. 3. ACKs notify the number of lost packets.

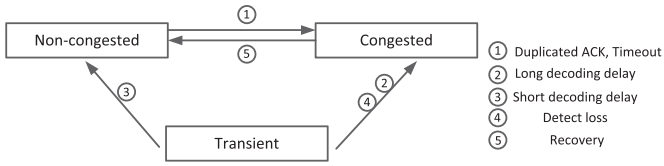


Fig. 4. State transition diagram.

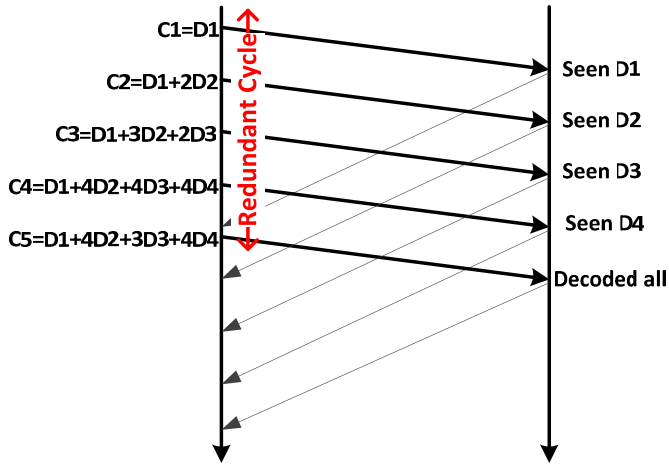


Fig. 5. Redundant cycle.

A. Packet Loss Detection

The field "loss flag" (one bit) is set in the network coding header. Receiver records the amount of data it received and the maximum relative sequence number[1] of data in a coded packet. When no loss happens, the amount of received data is equal to the maximum relative sequence number in coded packet (because of online coding). If the amount of received data turns less than the maximum relative sequence number, then the receiver will set the "loss flag" to notify the packet loss in the subflow. The amount of received data and the maximum relative sequence number is conveyed in ACKs. From this information, the amount of lost packets can be obtained by the sender. As shown in Fig. 3, the second ACK notifies that there are two packets lost between former ACK and current ACK.

B. Congestion Window Behavior in Transient State

If the sender of the subflow detects that the "loss flag" is set to "1," the transient window size is set to the value a little smaller than that of current congestion window size and enters "Transient state." In the transient state, the actual reason of packet loss is not clear, and further analysis about the packet loss reason is carried on. The increment style of congestion window (cwnd) and transient window in this state is the same with the RTT Compensator algorithm. The sending rate is limited by both twnd and cwnd. As shown in Fig. 4, when leaving the transient state, it will enter either the congestion state or the noncongestion state. If entering the noncongestion state, "cwnd" and "twnd" take the maximum value of both. If entering the congestion state, all the windows are halved and the new congestion window is set to the minimum value of the two windows.

C. Congestion Exposure

Duplicate ACK and timeout are two obvious notifications of congestion. If the number of lost packets is not large enough to trigger these notifications, then the packet loss reason is analyzed based on the behavior of packet loss event. Given this, wireless loss is scarce, whereas congestion loss is bursty. If the loss can be recovered by redundant packet soon and incur a short delay, it means the loss is caused by wireless link error and that the state turns to noncongested. On the contrary, if packets are lost nearly continuously and incur long decoding delay (i.e., the number of packets transmitted after the first undecodable packet), they can be inferred as congestion-based loss. Therefore, we propose that the duration of the "Transient state" must be upper limited and some extra procedures must be done in the "Transient state" to expose congestion. The upper limit of the duration of the "Transient state" is "Transient state threshold" or ΔT. If the decoding delay caused by packet loss is within the range of ΔT, the loss can be inferred as wireless loss or it can be inferred as congestion based. In the "Transient state," the number of transmitted packets (successfully or failed) after the first undecodable packet is recorded by a transient counter.

Before choosing an appropriate value for ΔT, we first introduce new concepts, namely, "Redundant Cycle" and "Redundant Cycle Size." According to the design principle of TCP with network coding, the redundant packets are sent periodically to mask potential packet loss. Periodical redundant packet divides transmission duration into "Redundant Cycles." It starts from the first nonredundant packet and ends at another redundant packet, as shown in Fig. 5. The number of packets transmitted in the Redundant Cycle, named "Redundant Cycle Size," is related to the redundancy parameter r. Therefore, we set it to $r/(r-1)$. For example, if the redundant parameter is 1.25, then ΔT is equal to 5. Assuming that, if no packet has been lost in previous cycles, there are two situations. If only one packet loses in the current cycle, it can be recovered by the redundant packet, and all the packets can be decoded. Under this circumstance, the minimum decoding delay is 0 and the maximum decoding delay is equal to "Redundant Cycle Size." If two or more packets are lost in a Redundant Cycle, it will cost redundant packets in later redundant cycles to recover loss, which incurs much longer decoding delay. According to the principle of TCP with network coding, the redundancy

[1]Relative sequence number is the value that the current sequence number subtracts the first sequence number.

parameter is usually a little higher than the required value to make the "Redundant Cycle" small enough to avoid the second situation mentioned previously.

According to the design rules of TCP with network coding, if the loss is wireless error based, there may be only one packet lost in a redundant cycle and all the coded packets can be decoded in a redundant cycle. Therefore, we set the value of ΔT to the Redundant Cycle size. In Couple+, when sender gets an ACK with "loss flag" set to one, it enters "Transient state" and the transient counter starts. The counter increases as the sender gets a nonduplicated ACK. If the newly received ACK conveys new packet loss information, the increase pace of the transient counter is equal to the number of lost coded packets plus one; otherwise, the increase pace is just one. The value of transient counter reaching ΔT means that the number of packets got by the receiver is ΔT. If all the packets are decoded before the counter reaches the upper limit, then it is noncongestion. A rare event is that several wireless link error-based packets are losing, which happens continuously. To avoid misjudging these packets as congestion, delay is the complementary criterion to judge congestion. Hence, if all the packets cannot be decoded when the counter reaches the upper limit and the RTT is higher than the minimum RTT in the noncongested state (at least twice higher), then the path is congested. The granularity is coarse. We do not need to analyze delay every time a loss happens.

Another factor to infer congestion is the limitation of the receiving buffer in the connection level. According to the design rules of MPTCP, packets will be stored in the receiving buffer if they cannot be organized in order. Given that, if the paths of MPTCP are almost symmetric or properly scheduled [27], [28], the occupation of the receiving buffer will be low. When congestion happens, congestion loss leads to large amount of packet loss and long decoding delay. Large amount of lost packets leads to insufficient ACKs to notify congestion. Meanwhile, congested path may also block transmission of the ACKs, which causes delay in feedback. Before the sender in the congested subflow detects a timeout or gets useful ACKs, the receiving buffer in the connection level may be largely occupied by packets from other subflows and further limiting the sending rate of all the subflows. Therefore, when the sending rate in the connection level is limited by the receiving buffer, it means that the receiving buffer is occupied by disorder data heavily and congestion may happen.

Based on this analysis, the main point to differentiate the reasons of packet loss is based on the decoding delay. The master judgment is that, if the receiver cannot decode all the received coded packets in a redundant cycle, congestion happens. The supplementary judgment is that, if the receiving buffer is nearly full in "Transient state," it must be congested.

V. Performance Analysis

Here, the efficiency of Couple+ is analyzed from two aspects. First, the behavior of packet loss is traced in a real network to verify the rationality of the newly proposed scheme. Second, the performance of Couple+ is analyzed by using the ns-3 network simulator [29].

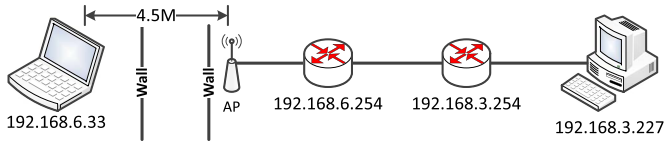


Fig. 6. Configuration of the test network.

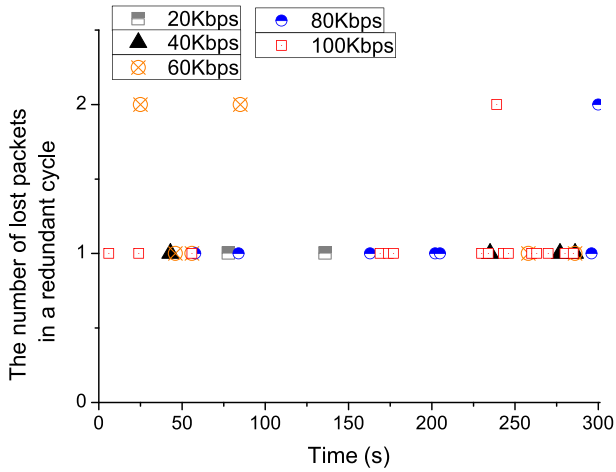


Fig. 7. Number of lost packets in a redundant cycle for wireless transmission.

A. Behavior of Packet Loss Caused by Wireless Errors

The main difference between congestion loss and random loss is the correlation between lost packets. Congestion loss is caused by queue overflow; hence, there seems to be nearly continuous packet loss in a congestion loss incident, particularly when the queue type is drop tail [30]. Bursty loss model is widely used to characterize loss caused by congestion [31], [32]. On the contrary, wireless transmission error is under the description of the Gilbert–Elliott (GE) mathematical model [33], [34]. The packet loss gap seems large with high probability if the packet loss is caused by wireless transmission error.

To prove the different characteristics of packet loss behavior caused by wireless interference more forcefully, we conduct experiments in a real network. The experiment network is shown in Fig. 6. Host 192.168.3.227 is the sender, and host 192.168.6.33 is the receiver. Packets are generated by Iperf [35], which is a test tool of network performance, including bandwidth, delay, jitter, and loss rate. The experiment includes two parts: 1) The behavior of lost packets caused by wireless error is recorded if host 192.168.6.33 accesses network via a wireless link (WiFi, IEEE 802.11 b/g/n); and 2) the packet loss rate in the noncongested state is recorded if host 192.168.6.33 accesses network via a wired link.

All the experiments are carried out in noncongested state. First, if host 192.168.6.33 accesses network via a wireless link, the available bandwidth measured by Iperf is fluctuated between 400 Kb/s and 3 Mb/s. To avoid congestion, the transmission rate is limited to no more than 100 Kb/s. Data is conveyed by user datagram protocol. In Fig. 7, it shows the number of lost packets in a redundant cycle. Assume that, if TCP with network coding is adopted here, the proper value of redundancy may be 1.05; hence, the redundant cycle size is 21,

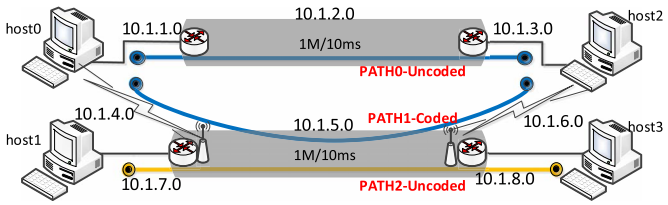


Fig. 8. Simulation topology.

which means that the number of transferred packets in a redundant cycle is 21. Because the maximum transmission rate is 100 Kb/s, all the lost packets can be inferred as wireless error based. The measured loss rate caused by link error is zero if host 192.168.6.33 accesses network via a wired link.

From the experiments, we can obtain the following observations: 1) The link error-based packet loss indeed exists in a wireless network, whereas this kind of packet loss does not occur in a wired network; and 2) the number of lost packets in a redundant cycle is one with extremely high probability, which proves that it is reasonable to set the upper limit of duration of transient state to redundant cycle size.

B. Performance of Couple+

The unfairness problem is due to network coding. In simulation experiments, ignoring the detailed differences in previous congestion control schemes for MPTCP, RTT Compensator, detailed in Section II-A, is chosen as a comparison scheme. We implement "Couple+" in ns-3.

1) Data Allocation Among Subflows of an MPTCP Socket and Performance of TCP in Common Bottleneck: To verify that network coding can lead to unfair load allocation among subflows of MPTCP and Couple+ can recover the fairness between subflows or between subflows and TCP flow, simulations are conducted in ns-3. Fig. 8 shows a wired–wireless network with two bottlenecks, one of which is used by subflow 1 (sub1) of MPTCP and a noncoded TCP flow. All the paths have equal basic RTT of 60 ms (each link has equal delay of 10 ms). The bandwidth of each link is 1 Mb/s. MPTCP flow and TCP flow compete for the bandwidth in subnet of 10.1.5.0. The total amount of data for each socket is 5 MB. During simulation, both TCP flow and MPTCP flow are activated. Only the statistics in competition period are recorded and averaged. The link between 10.1.4.0 and 10.1.6.0 is wireless based, and subflow 1 can be coded to mask wireless loss from interfering congestion control. The TCP flow and subflow 0 keep noncoded. We assume that all the terminals are static here.

Considering the influence of redundant factor, the simulations are conducted under different values of redundancy parameter. When using Couple+, subflow 1 is coded. Setting the wireless loss rate to 0.01, 0.05, and 0.1, and coding window size to 6, the performance of Couple+ with different redundancy parameters is shown in Fig. 9. Fig. 9 gives the throughput comparison of two MPTCP subflows in competition period if RTT Compensator and Couple+ are adopted, respectively. Accordingly, Fig. 10 gives the throughput comparison of TCP flows sharing bottleneck with subflow 1 if RTT Compensator and Couple+ are adopted, respectively.

Figs. 9 and 10 should be compared to observe. The comparison of Figs. 9 and 10 shows that, if RTT Compensator is adopted as the congestion control scheme in MPTCP/NC, the throughput of subflow 1 is many times higher than the throughput of TCP flow in the same bottleneck. For example, in Fig. 9(a), when the wireless loss rate is 0.01, the redundancy is 1.05 and the throughput of subflow 1 is around 700 Kb/s, whereas in Fig. 10(a), the corresponding TCP flow gets throughput less than 100 Kb/s. If using Couple+, as shown in Fig. 10, the throughput of TCP flow is significantly increased, which is nearly equal to that of subflow 1. This obeys the principle "Do No Harm." Couple+ weakens the aggressiveness of coded subflows. The maximum throughput of coded subflow is nearly equal to or less than that of TCP flow in the congested bottleneck. Although the throughput of coded subflow 1 is less than TCP flow, the total throughput of MPTCP/NC is higher than that of TCP flow, which obeys the principle that MPTCP can improve the transmission rate of flow by taking advantage of fragmented resources on paths. The difference of throughput between subflow 0 and subflow 1 is much higher adopting Couple+ than that adopting RTT Compensator. This obeys the principle "Balance Congestion." The phenomena are similar in Fig. 9(b) and (c), respectively, as compared with Fig. 10(b) and (c).

It is obvious that the throughput of subflow 1 is higher if the congestion control scheme is RTT Compensator. It verifies that network coded subflow can bear more load and further improves the whole throughput of MPTCP if adopting RTT Compensator. However, it sacrifices the performance of other flows, particularly those congestion-sensitive flows. In Fig. 10, the data show the throughput of TCP flow in the competition phase. It is obvious that coded subflow conveys more load at the cost of damaging the performance of congestion sensitive flows. On the contrary, Couple+ is friendly to congestion-sensitive flows. Adopting Couple+, the performance of TCP is much better.

2) Performance in a Mobile Scenario: Here, we compare the performance of Couple+ and RTT Compensator in a mobile scenario. In Fig. 11, the mobile terminal accesses network via 4G and WiFi concurrently; thus, both subflows are coded. The wireless loss rate of 4G access network is 0.001; hence, the redundancy of sub0 is 1.05 and its coding window size is 6. The wireless loss rate of WiFi is 0.05; therefore, the redundancy of sub1 is 1.15. The user starts moving away from the coverage of WiFi at 20 s and moves back to the initial location at 32 s, which means that the link state of WiFi becomes worse at 20 s and recovers to the original state at 32 s. In the worst link state, the user cannot access the network via WiFi, the duration of which lasts for 2 s.

Figs. 12 and 13 show the instantaneous throughput of flows if the used congestion control scheme for coded subflows is Couple+ or RTT Compensator, respectively. Despite the fairness existing between subflows and between coded subflow and TCP flow if using our proposed scheme, here, what to be noticed is the instantaneous throughput of subflow 1, which is conveyed by a WiFi network. In Figs. 12(b) and 13(b), throughput starts decreasing after 20 s when the link state becomes worse. Adopting RTT Compensator, the lowest throughput of sub1 is higher than that if adopting Couple+. Adopting RTT

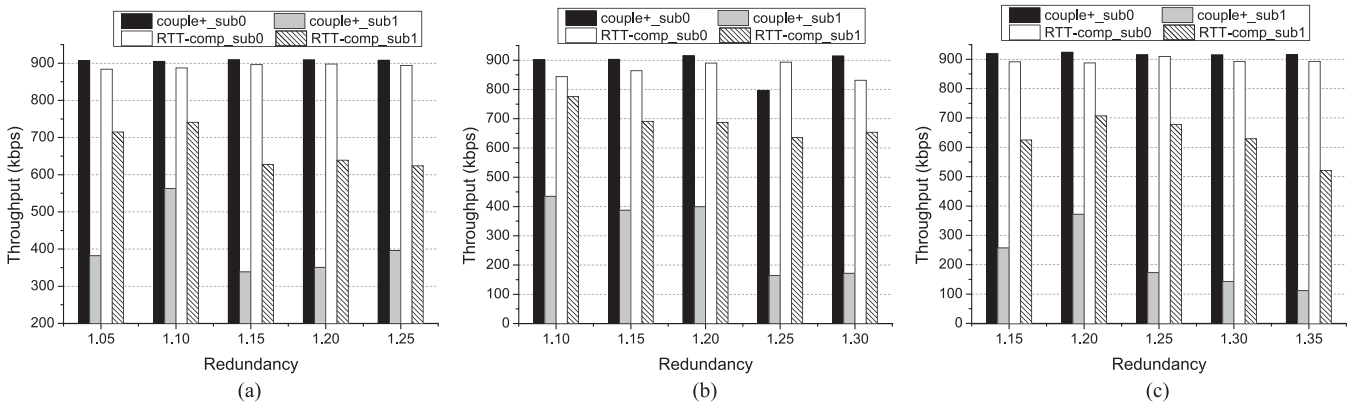


Fig. 9. Throughput of subflows in competition period when the wireless loss rate is 0.01, 0.05, and 0.10, respectively; all the coding window size is 6. (a) Random loss rate is 0.01. (b) Random loss rate is 0.05. (c) Random loss rate is 0.10.

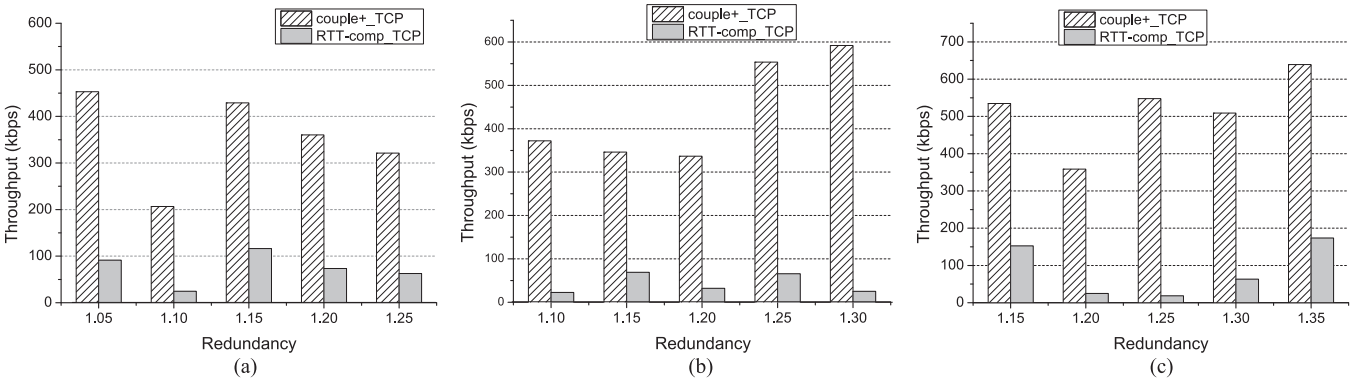


Fig. 10. Throughput of TCP flow in competition period when the wireless loss rate is 0.01, 0.05, and 0.10, respectively; all the coding window size is 6. (a) Random loss rate is 0.01. (b) Random loss rate is 0.05. (c) Random loss rate is 0.10.

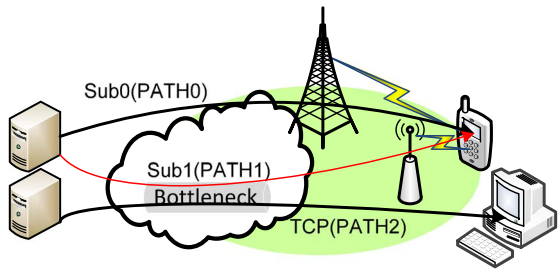


Fig. 11. Simulation topology in a mobile scenario.

Compensator, sender cannot detect the abnormal packet loss timely, which is due to congestion or link failure. Hence, sender will keep a large congestion window and make the throughput higher than the normal level. Under the circumstances, sender cannot make the right decision to close the bad path timely, which will further incur heavy disorder in the connection level and waste energy of the terminal.

3) Performance of Couple+ in Noncongested Scenarios: Any congestion control scheme must promise that it not only can release resources effectively and keep fair with each other but also can ensure that the flow can achieve throughput as high as possible. Hence, we also conduct simulations to evaluate the performance of Couple+ and standard MPTCP (both subflows without network coding) in a noncongested network and give the result in Fig. 14. The wireless loss rate is set to 0.01, 0.05, and 0.10, respectively. In these scenarios, only MPTCP flow is activated; thus, there is no congestion in subnet 10.1.5.0.

First, we conduct the simulation with Couple+, where Subflow 1 is coded and Subflow 0 is uncoded. The network topological structure has been shown in Fig. 8. With different redundancy parameters, the volume of data transferred by subflows is respectively shown as "Couple+sub0" and "Couple+sub1" in Fig. 14. Then, we conduct the simulation with standard MPTCP (adopting RTT Compensator), where both subflows are uncoded. The volume of data transferred is shown in lines as "sub0 without coding" and "sub1 without coding" in Fig. 14. The volume of data sent by sub1 is low when it is not coded because of the often reduced congestion window, which is caused by wireless random loss. Although the volume of data sent by sub1 (with coding) in Couple+ scenario is only a little less than that transferred by sub1 (without coding) in standard MPTCP scenario, the throughput in the former is much higher than that in the latter. In other words, network coded flow

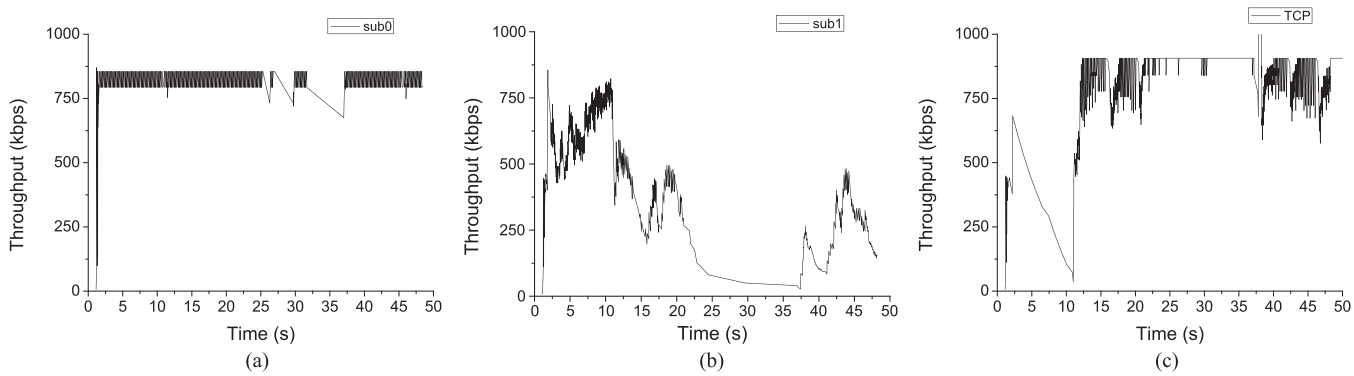


Fig. 12. Instantaneous throughput of flows if the congestion control scheme for coded subflows is Couple+. (a) Instantaneous throughput of sub0. (b) Instantaneous throughput of sub1. (c) Instantaneous throughput of TCP.

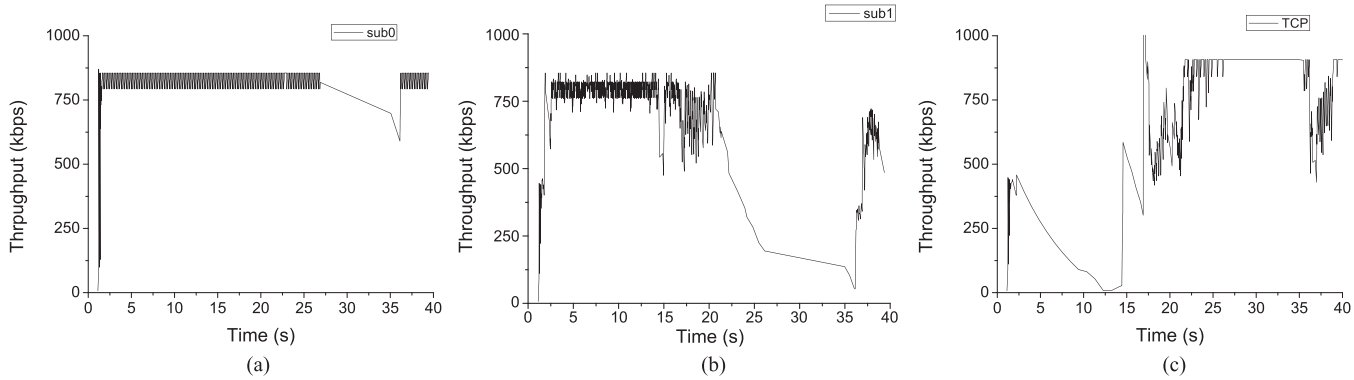


Fig. 13. Instantaneous throughput of flows if the congestion control scheme for coded subflows is RTT Compensator. (a) Instantaneous throughput of sub0. (b) Instantaneous throughput of sub1. (c) Instantaneous throughput of TCP.

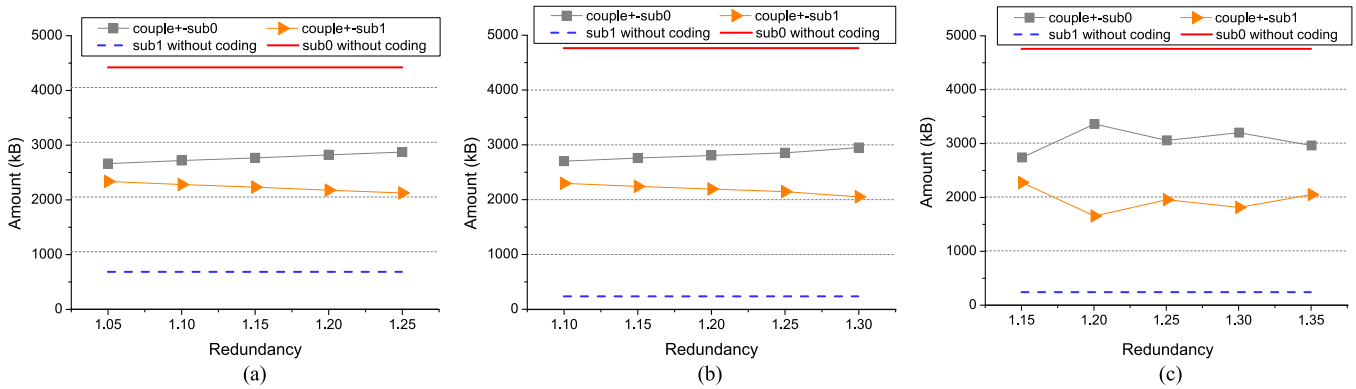


Fig. 14. Performance comparison of RTT Compensator and Couple+ in noncongested state when the wireless loss rate is 0.01, 0.05, and 0.10, respectively. (a) Random loss rate is 0.01, and coding window size is 6. (b) Random loss rate is 0.05, and coding window size is 6. (c) Random loss rate is 0.10, and coding window size is 6.

with Couple+ can keep the advantage to that network coding brings to TCP flow in wireless links.

C. Summary

From the simulations, we can summarize some key points of our work.

1) The previous congestion control scheme is not fit for MPTCP/NC because network coding makes the subflow less sensitive to congestion. Previous schemes may cause several problems, for example, invalidation of "Balance Load" among subflows and unfriendliness to other congestion-sensitive flows, i.e., "Do No Harm."
2) Couple+ can overcome the aforementioned problems. It can recover the allocation of load among subflows and release bandwidth resources to other congestion-sensitive flows.
3) Couple+ can make sender detect abnormal packet loss timely, which may contribute to efficient path selection scheme.
4) Couple+ can still keep the advantage that network coding brings to TCP flow in wireless links.

Therefore, Couple+ can be treated as an effective congestion control scheme, particularly for MPTCP with network coded subflows.

VI. Conclusion

In this paper, focusing on MPTCP/NC, the invalidation of "Do No Harm" and "Balance Congestion" in the congestion control scheme for MPTCP/NC in a wired–wireless network has been highlighted. Introducing network coding into TCP makes the sender less sensitive to congestion because part of the congestion loss is hidden by network coding. To deal with this problem, we first prove this issue with optimization theory and then propose a new scheme to overcome the unexpected side effect of network coding in MPTCP. The results show that network coding indeed interferes with load balancing among subflows. By choosing proper coding parameters, our scheme can recover the basic congestion control principle of "Do No Harm" and "Balance Congestion" for MPTCP/NC flow.

Acknowledgment

The authors would like to thank the anonymous referees for their invaluable suggestions that have led to the present improved version of the original manuscript.

References

[1] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath TCP development," IETF, Fremont, CA, USA, Inf. RFC 6182, 2011.
[2] A. Singh, M. Xiang, A. Konsgen, C. Goerg, and Y. Zaki, "Enhancing fairness and congestion control in multipath TCP," in *Proc. 6th Joint IFIP Conf. WMNC*, 2013, pp. 1–8.
[3] C. Raiciu, D. Wischik, and M. Handley, "Practical congestion control for multipath transport protocols," Univ. College London, London, U.K., Tech. Rep., 2009.
[4] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," in *Proc. 8th USENIX Conf. NSDI*, 2011, vol. 11, pp. 8–8.
[5] S. Hassayoun, J. Iyengar, and D. Ros, "Dynamic window coupling for multipath congestion control," in *Proc. 19th IEEE ICNP*, 2011, pp. 341–352.
[6] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath TCP," in *Proc. 20th IEEE ICNP*, 2012.
[7] K. Tan, F. Jiang, Q. Zhang, and X. Shen, "Congestion control in multihop wireless networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 863–873, Mar. 2007.
[8] M. Bottigliengo, C. Casetti, C. Chiasserini, and M. Meo, "Enhancing fairness for short-lived TCP flows in 802.11 b WLANs," *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 206–217, Jan. 2007.
[9] C. Chiasserini and M. Meo, "A reconfigurable protocol setting to improve TCP over wireless," *IEEE Trans. Veh. Technol.*, vol. 51, no. 6, pp. 1608–1620, Nov. 2002.
[10] J. K. Sundararajan *et al.*, "Network coding meets TCP: Theory and implementation," *Proc. IEEE*, vol. 99, no. 3, pp. 490–512, Mar. 2011.
[11] J. Cloud *et al.*, "Multi-path TCP with network coding for mobile devices in heterogeneous networks," in *Proc. 78th IEEE VTC Fall*, 2013, pp. 1–5.
[12] Y. Cui, X. Wang, H. Wang, G. Pan, and Y. Wang, "FMTCP: A fountain code-based multipath transmission control protocol," in *Proc. 32nd IEEE ICDCS*, 2012, pp. 366–375.
[13] G. Sarwar, P.-U. Tournoux, R. Boreli, and E. Lochin, "eCMT-SCTP: Improving performance of multipath SCTP with erasure coding over lossy links," in *Proc. 38th IEEE Conf. LCN*, 2013, pp. 476–483.
[14] M. Li, A. Lukyanenko, and Y. Cui, "Network coding based multipath TCP," in *Proc. IEEE INFOCOM WKSHPS*, 2012, pp. 25–30.
[15] M. Li, A. Lukyanenko, S. Tarkoma, Y. Cui, and A. Ylä-Jääski, "Tolerating path heterogeneity in multipath TCP with bounded receive buffers," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, 2013, pp. 375–376.
[16] P.-L. Ageneau, N. Boukhatem, and M. Gerla, "Fairness evaluation of pipeline coded and non coded TCP flows," in *Proc. IEEE ICC*, 2014, pp. 2754–2760.
[17] H. Zhang, K. Xue, P. Hong, and S. Shen, "Congestion exposure enabled TCP with network coding for hybrid wired–wireless network," in *Proc. 23rd ICCCN*, 2014, pp. 1–8.
[18] T. Bonald, "Comparison of TCP Reno and TCP Vegas via fluid approximation," INRIA, Rocquencourt, France, Tech. Rep., 1998.
[19] Y. Chen, X. Wu, and X. Yang, "MAPS: Adaptive path selection for multipath transport protocols in the Internet," Duke Univ., Durham, NC, USA, TR-2011-09, 2011. [Online]. Available: http://catbert.cs.duke.edu/~yuchen/papers/maps.pdf
[20] S. Chen, Z. Yuan, and G.-M. Muntean, "An energy-aware multipath-TCP-based content delivery scheme in heterogeneous wireless networks," in *Proc. IEEE WCNC*, 2013, pp. 1291–1296.
[21] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.
[22] C. Raiciu, M. Handley, and D. Wischik, "Coupled congestion control for multipath transport protocols," IETF, Fremont, CA, USA, Inf. RFC 6356, 2011.
[23] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda, "An energy-aware multipath-TCP-based content delivery scheme in heterogeneous wireless networks," in *Proc. 7th Int. Workshop PFLDNeT Workshop*, Tokyo, Japan, May 2009, pp. 19–24.
[24] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley, "Overlay TCP for multi-path routing and congestion control," in *Proc. IMA Workshop Meas. Model. Internet*, 2004, pp. 1–24.
[25] F. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 2, pp. 5–12, Apr. 2005.
[26] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "MPTCP is not Pareto-optimal: Performance issues and a possible solution," in *Proc. 8th Int. CoNEXT*, 2012, pp. 1–12.
[27] F. H. Mirani, N. Boukhatem, and M. A. Tran, "A data-scheduling mechanism for multi-homed mobile terminals with disparate link latencies," in *Proc. IEEE 72nd VTC Fall*, 2010, pp. 1–5.
[28] D. Ni, K. Xue, P. Hong, and S. Shen, "Fine-grained forward prediction based dynamic packet scheduling mechanism for multipath TCP in lossy networks," in *Proc. 23rd ICCCN*, 2014, pp. 1–7.
[29] ns-3. [Online]. Available: https://www.nsnam.org/
[30] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82, Jul. 1997.
[31] K. Zhou, K. L. Yeung, and V. O. Li, "On bursty packet loss model for TCP performance analysis," in *Proc. Workshop HPSR*, 2005, pp. 292–296.
[32] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 303–314, Oct. 1998.
[33] E. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, Sep. 1963.
[34] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, Sep. 1960.
[35] Iperf. [Online]. Available: https://iperf.fr/

Kaiping Xue (M'09–SM'15) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2007, respectively.

Currently, he is an Associate Professor with the Department of Information Security and the Department of Electronic Engineering and Information Science, USTC. His research interests include next-generation Internet, distributed networks, and network security.

Jiangping Han will receive the B.S. degree in July 2016 from the University of Science and Technology of China (USTC), Hefei, China, where she will be working toward the M.S. degree in communication and information systems with the Department of Electronic Engineering and Information Science.

Her research interests include next-generation Internet performance optimization.

Hong Zhang received the Master's degree from the University of Science and Technology of China (USTC), Hefei, China, in 2015.

Her research interests include multipath transmission control protocol performance optimization.

Ke Chen received the B.S. degree from Sichuan University, Chengdu, China, in 2013. She is currently working toward the M.S. degree in communication and information systems with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China.

Her research interests include mobility management and multipath transmission control protocol performance optimization.

Peilin Hong received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 1983 and 1986, respectively.

Currently, she is a Professor and Advisor of Ph.D. candidates with the Department of Electronic Engineering and Information Science, USTC. She has published two books and more than 150 academic papers in several journals and conference proceedings. Her research interests include next-generation Internet, policy control, internet protocol quality of service, and information security.