Adaptive Batch Coding: A Balanced Congestion Control Strategy for Multi-Beam Antenna Networks

Qian Mao, Fei Hu, *Member, IEEE,* Ke Bao, Sunil Kumar

***Abstract*—Multi-Beam Smart Antennas (MBSAs) achieve concurrent communications in multiple beams, thereby providing higher throughput compared to regular directional antennas. Most of the transport control schemes used today are based on TCP, which are too conservative since they indiscriminately reduce the window size upon any packet loss. The other extremity is the de-congestion control strategy, which abandons both window size control and retransmission, aggressively saturates the network, and compensates packet loss via network coding. To adapt to the features of the MBSA-based networks, this work proposes a balanced transport control strategy between the above two categories, which is based on the idea of *Adaptive Batch Coding (ABC)*. The proposed ABC scheme resists random loss through redundant coding and copes with congestion via window shrink and retransmission. Using a cross-layer design (between transport and routing layer), both the coding scheme and the traffic allocation are adaptively adjusted according to network conditions. A customized simulation system has been developed to comprehensively evaluate the performances of the proposed ABC protocol. Experimental results show that the proposed scheme overcomes the drawbacks of those two extremities: it achieves both high throughput and high good throughput (under loss) in MBSA-based networks.**

***Index Terms*—Congestion Control, Multi-Beam Smart Antenna (MBSA), Network Coding, Adaptive Batching Coding (ABC), Cross-Layer Design.**

I. Introduction

The Multi-Beam Smart Antenna (MBSA) has multiple beams in different directions and can *simultaneously* send/receive packets through the beams by using simple beam control mechanism, therefore attracting many interests in both industry and academia [1, 2]. Meanwhile, due to its flexible deployment and improved throughput, wireless backhaul has drawn many attentions and has been standardized in 3GPP long term evolution (LTE) [3]. The in-band wireless backhaul provides many benefits in terms of hardware cost and frequency reuse, therefore becoming a promising technique for 5G network [4, 5]. The delay performances of various kinds of backhaul models have been studied, and the relationship among packet delay, packet size, number of hops, etc., has been deduced [6]. Using a delay-based access control policy, an improved performance was achieved in the heterogeneous backhaul network [7]. Considering the mesh structure of the wireless backhaul network, the MBSA is an ideal choice. In this work, we aim to design *a transport layer protocol that provides reliability and controls congestion with the consideration of MBSA features*, thereby improving the performances of wireless networks.

A popular option for transport layer protocol is TCP. However, when TCP is applied to wireless network, its conservative data dispatching strategy significantly constrains the throughput performance. Balakrishnan et al. [8] pointed out that wireless networks suffer from packet loss not only because of congestion but also due to link errors, node mobility, etc., and they drew the conclusion that cross-layer (transport/data link layer) design and selective acknowledgments could bring significant improvement to the throughput. After that, the research focusing on cross-layer design to distinguish the packet loss types has drawn some attentions [9, 10]. To increase network utilization and limit packet loss, Active Queue Management (AQM) was proposed and developed, which efficiently reduced congestion by intelligently dropping packets in a nearly full buffer [11]. However, due to its relatively conservative congestion control strategy, the average throughput of the TCP-based network is less than 50% of the maximum network capacity, and may be even worse in multi-hop, multi-path network architecture [12].

Another kind of transport layer design is called *de-congestion control*, which uses redundancy codes to resist packet loss. Network coding can be divided into two categories, i.e., fixed-rate codes and rateless codes. One of the most popular fixed-rate codes is the batch coding. To decrease the decoding delay, Chen et al. proposed a pipelined network coding scheme [13]. Another category of network coding is rateless codes, pioneered by LT codes and Raptor codes, and boomed by Fountain codes [14-16]. De-congestion control removes TCP-based congestion control and uses redundancy coding to retrieve the lost packets. It improves the throughput compared to TCP in some circumstances. However, applying de-congestion control directly to the MBSA networks has some drawbacks: 1)It does not consider the Concurrent Packet Transmission/Reception (CPT/CPR) constraints of the MBSA. If the coded symbols are simply dispatched to all beams, some beams may finish Tx earlier than others, thus wasting bandwidth. 2)It sacrifices good throughput. The multi-hop, multi-path wireless network increases the possibility of packet loss as well as the ACK feedback time. If the sender keeps sending symbols until a positive ACK is received, the ratio of the raw data could be very low. 3) It does not have efficient congestion control. When congestion occurs, merely sending more encoded symbols without decreasing the sending rate

Qian Mao is with the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL, 35401 USA; e-mail: qmao3@crimson.ua.edu

Fei Hu is with the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL, 35401 USA; e-mail: fei@eng.ua.edu

Ke Bao is with the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL, 35401 USA; e-mail: oucbaoke@hotmail.com

Sunil Kumar is with the Department of Electrical and Computer Engineering, San Diego State University, USA; e-mail: skumar@mail.sdsu.edu

Revised Manuscript submitted in August 2017.

may actually aggravate the congestion situation.

We would like to point out that the CPT/CPR feature of MBSAs poses certain challenges for transport layer design. For example, in a multi-path, coupled transmission (Fig.1 (c)) which can better utilize MBSA's parallel data delivery in all beams, all the nodes in the same hop (i.e., with the same hop distance to the source) must be in coarse synchronization status, i.e. either All Tx (transmission) or All Rx (reception) status. Thus the congestion control scheme must consider such a coarse schedule synchronization between different hops. For example, if one hop is in Tx mode, the next hop must be in Rx mode if the nodes of next hop want to receive packets from last hop. Such a special MBSA data forwarding pattern requires that the MBSA transport control must be integrated with MBSA-oriented routing architecture. (in this work, we will integrate our transport scheme with a special grid routing topology that is suitable to MBSA CPT/CPR patterns). Moreover, the queues belonging to the same node (but in different beams) must all be cleared out within approximately the same time. Otherwise, one beam may finish transmission earlier than other beams and sit there in 'idle' status. This requires that some sort of fair traffic allocation strategy be used among different beams to guarantee approximately the same queue clearance time among all the queues. In this work, we will propose a beam-adaptive traffic allocation strategy.

Since TCP is overly conservative in congestion control and the de-congestion control is too aggressive in saturating network capacity, some protocols combine those two categories to achieve a balanced performance. A congestion control method using network coding was proposed by Chen et al., which focused on the flow control for multicast flows [17]. Kiss et al. proposed a network coding based congestion control scheme performed in the network layer of wired networks [18]. A TCP adaptation scheme based on rateless codes and opportunistic data forwarding was proposed in [19]. Hou et al. studied how to 'smartly' drop packets at the source to balance the caching size and the packet loss ratio during network coding [20]. The impact on the stability of TCP-Reno scheme when a network coding layer is inserted in the TCP/IP stack was comprehensively analyzed in [21]. However, to the best of our knowledge, mo research has been directed toward the efficient congestion control for MBSA-equipped networks. Considering the specific features of the MBSAs, much improvement can be achieved if the network coding is incorporated with the special routing features of the MBSA network. Therefore, in this work, a novel Adaptive Batch Coding (ABC) based transport layer protocol is proposed, which makes the following contributions:

- 1) *Using a novel network coding scheme to balance the congestion control strategies.* The proposed ABC scheme combines the redundancy coding and window size control to provide reliability and congestion control, thereby increasing the good throughput.
- 2) *Cross-layer design.* Link quality parameters are collected from the routing layer and are used to adjust the ABC coding scheme. Furthermore, each node dynamically allocates traffic among multiple beams based on the cross-layer information.
- 3) *Congestion control based on loss differentiation.* The ABC-based protocol distinguishes the types of packet loss by using redundancy coding: the random loss is retrieved by redundancy, and the retransmission and window size shrinking are launched upon congestion.
- 4) *Perfectly fitting the features of MBSAs.* A pipe-like data forwarding architecture based on the ABC scheme is proposed, which explores the MBSAs concurrent communication capability and increases good throughput.

The proposed congestion control scheme is designed for wireless mesh networks, and it can be easily transplanted to 5G wireless backhaul networks. The rest of the paper is organized as follows: Section II briefly introduces the related works. The detailed ABC scheme is described in Section III. In section IV, a congestion control strategy based on the ABC scheme and cross-layer design is presented. The simulation results and analyses are presented in Section V, followed by the conclusions in Section VI.

II. Related Works

To provide transmission reliability and congestion control to mesh network, multi-path-oriented TCP schemes have been proposed. A typical one, called Multipath TCP (MPTCP), has been standardized by Internet Engineering Task Force (IETF) in 2013 [22]. It enables the hosts to build additional paths between the sender and the receiver, and to use multiple paths with different IP addresses to transmit data simultaneously, as shown in Fig. 1 (a).

The MPTCP only exploits multiple output/input links for the transmitter and receiver, and the intermediate nodes still work at a single-input-single-output pattern. To exploit the multi-directional communication capability of the MBSAs, our previous work [23] proposed a Ripple-Diamond-Chain (RDC) routing topology (Fig.1 (b)). In this scheme, the entire route is composed of a series of diamond-shaped sections, each of which includes a main path and two side paths. The main path and side paths deliver the data via different beams with well-separated collision domains. Compared to MPTCP, RDC makes better use of the multi-beam capacity of the intermediate nodes. However, the formation of the RDC topology requires five nodes for each diamond section, which makes the chain infeasible if there are not enough nodes for a section, and also limits the further capacity improvement if there are more than five nodes available.

In 2014, Loch et al. proposed a corridor topology for Orthogonal Frequency-Division Multiple Access (OFDMA) wireless networks [24]. To fully exploit the frequency channels of each node, a corridor involving many nodes is built to bridge the sender and the receiver, as shown in Fig. 1 (c). In the corridor, the intermediate nodes having the same number of hops from the sender comprise a stage, and each node simultaneously communicates with multiple nodes belonging to its last and next stages.

Inspired by their work, a fence routing scheme was proposed by our previous work [25]. In this scheme, a transmission corridor is created between the sender and the receiver, which is composed of a main path and many side paths. Different

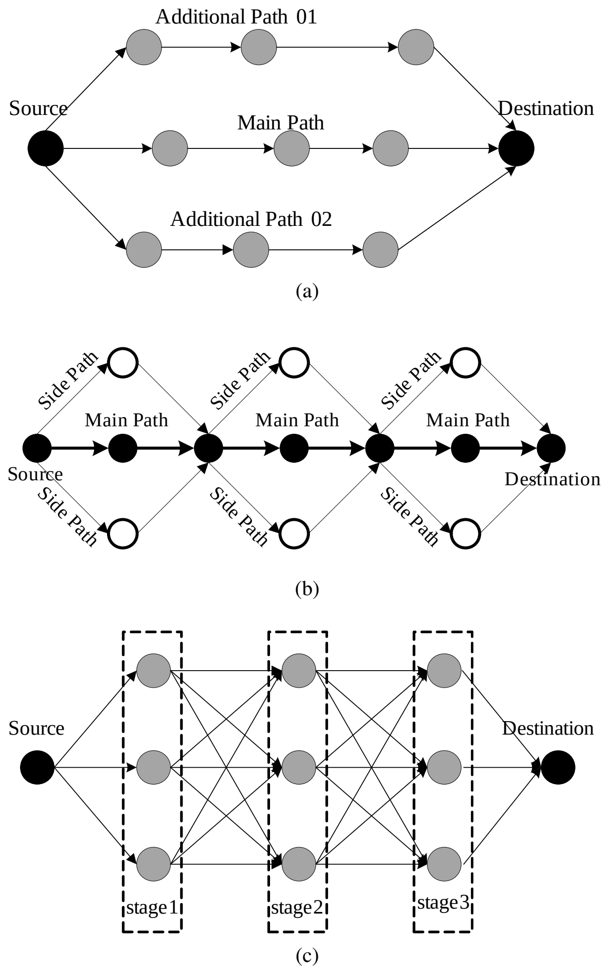


Fig. 1: Multi-Path Topologies
(a) Multipath TCP; (b) RDC topology; (c) Corridor Topology

from the RDC and the corridor topologies, fence routing does not require an identical number of nodes for each stage, thereby providing more flexibility to the topology formation.

III. A Novel Network Coding Scheme: Adaptive Batch Coding (ABC)

The proposed adaptive batch coding scheme allows the communication entities to transmit messages successfully without retransmission when random loss happens. The ABC scheme first estimates the path quality based on the data collected from the network layer, and determines the coding rate. According to the coding rate, redundant symbols are generated through a bit-wise exclusive-OR (XOR) operation upon the original data. Once the batch coding is finished, the source node attaches a unique ID to each encoded symbol and dispatches all the symbols through the main path as well as the side paths. Each node, including the source and all the intermediate nodes, collects the transmission quality of all of its outgoing links and allocates traffic according to link quality.

The decoder at the receiver side shares the same generator matrix with the encoder at the sender side. For each block (batch), if the received symbols are sufficient enough for decoding, the decoder starts to decode and retrieves all of

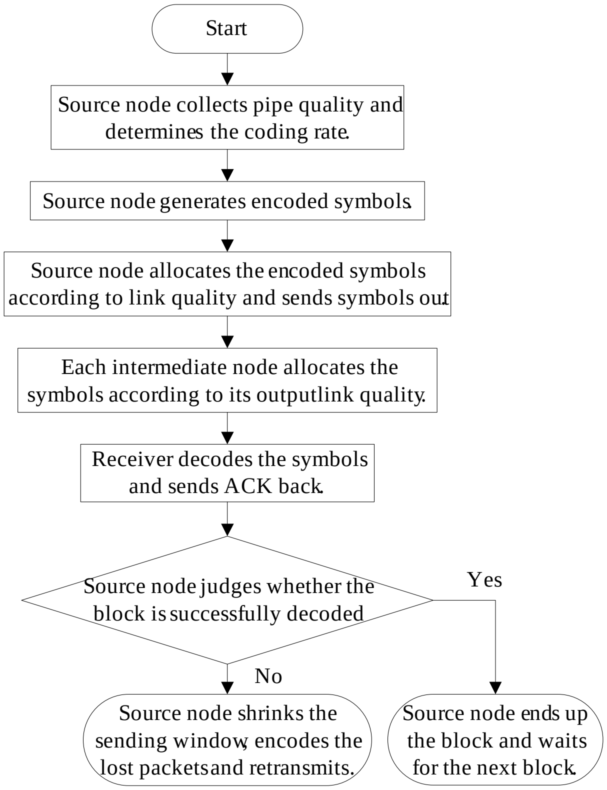


Fig. 2: Flowchart of the Proposed ABC Protocol

the original packets successfully. Note that the decoder does not necessarily wait for all of the symbols of a batch due to the redundancy. However, if there are so many lost symbols that the received symbols are not sufficient for successful decoding, a Negative Acknowledgement (NACK) is sent back to the source, indicating the IDs of the decoding-failed packets. The source then shrinks the window size, encodes the failed packets, and retransmits them. Note that a large buffer is possessed by the sender and receiver to combat the channel condition variation. The schematic of the proposed ABC scheme is shown in Fig. 2.

Assuming that the original message batch is $M = [m_1, m_2, \cdots, m_k]$, and each packet contains h bits, then, a (n, k) batch encoder generates the encoded block C through the following computation:

$$C = G * M = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nk} \end{bmatrix} * \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_k \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad (1)$$

where G is a pre-defined generator matrix, $g_{ij} \in [0, 1]$, and c_i is the encoded symbol with a size of h bits ($i \in [1, 2, \cdots, n]$, $j \in [1, 2, \cdots, k]$). The coding rate is $r = k/n$.

The generator matrix is designed according to the propagation features of each link. In the wireless backhaul network, due to the factors of nodes' mobility, line-of-sight (LoS) condition, jamming, etc, the quality of the wireless backhaul channel varies from link to link and from time to time [3, 4]. Since the fence routing protocols always pick the best link in each stage to comprise the main path, we set the background for our network coding design as, 1) the main path provides higher capacity and lower error rate, and the transmission condition is relatively stable, and 2) the side paths have lower capacity, higher error rate, and more unstable conditions.

On one hand, the multi-hop transmission accumulates the packet errors; on the other hand, the multi-path transmission aggravates the out-of-order problem, since the link qualities of the main path and side paths are different. Apparently, adding slight redundancy to resist random loss is beneficial. However, when congestion occurs, adding too much redundancy may aggravate congestion. Therefore, the proposed ABC scheme employs the following strategies:

- A systemic coding method is adopted for fast decoding, which significantly reduces the computational complexity and decoding delay.
- The task of the redundancy information is to resist random loss, thus, the amount of redundancy is generally low, and the coding rate is dynamically adjusted according to the link error rate.
- The generation of the redundancy is based on the routing characteristics and the path quality.

Thus, the generator matrix of the proposed ABC scheme is defined as:

$$G = \begin{bmatrix} I_k \\ G^o_{1\times k} \\ G^p_{l\times k} \end{bmatrix} = \left[\begin{array}{cccc} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ \hdashline 1 & 1 & \dots & 1 \\ \hdashline g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{l1} & g_{l2} & \cdots & g_{lk} \end{array}\right] \tag{2}$$

The generator matrix is composed of three parts: 1) an identical matrix, I_k, with a size of $k \times k$; 2) an *overall checking submatrix*, $G^o_{1\times k}$, which is an $1 \times k$ all-one vector. The overall checking submatrix generates slight redundancy to provide comprehensive protection against the random loss potentially happened to every packet in a batch; and 3) the *partial checking submatrix*, $G^p_{l\times k}$, which is optional and has a size of $l \times k$. The pattern of $G^p_{l\times k}$ is related to routing characteristics and is designed in such a way that the symbols traveling through "poor" paths are provided extra protections.

Therefore, the encoded symbols obtained by (1) is:

$$\begin{aligned} C &= \begin{bmatrix} C^s & C^o & C^p \end{bmatrix} \\ &= \begin{bmatrix} c^s_1 & c^s_2 & \dots & c^s_k & c^o_1 & c^p_1 & c^p_2 & \dots & c^p_l \end{bmatrix} \end{aligned} \tag{3}$$

There are three parts in the encoded block C:

- The first part, C^s, includes the *systemic symbols*, which are identical to the original packets, i.e., $c^s_i = m_i$ ($i = 1, 2, \dots, k$). A systemic coding strategy decreases the decoding complexity. If no congestion happens, the decoder submits the packets to the upper layer without waiting for the redundant symbols.
- The second part includes an *overall checking symbol* (OCS), i.e., $c^o_1 = m_1 \oplus m_2 \oplus \cdots \oplus m_k$, where $\oplus$ is the bit-wise XOR operation. Apparently, the OCS can resist single packet loss within a batch. In the proposed ABC scheme, the OCS is mandatory to be employed.
- The last part includes the *partial checking symbols* (PCS). The PCSs are optional, and are obtained by the bit-wise XOR operations implemented on particular packets. If some side paths persistently yield worse quality, more redundancy can be added to the packets travelling through these paths to provide extra protection. However, the PCSs are effective only when the transmission situations are so stable that the source node is able to determine which packets are travelling through the side paths.

An example based on the RDC topology as shown in Fig.1 (b) is presented. Assuming that the side paths undertake $1/3$ traffic approximately, the generator matrix covering 4 original packets within a batch can be designed as:

$$G = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hdashline 1 & 1 & 1 & 1 \\ \hdashline 0 & 0 & 1 & 1 \end{array}\right] \tag{4}$$

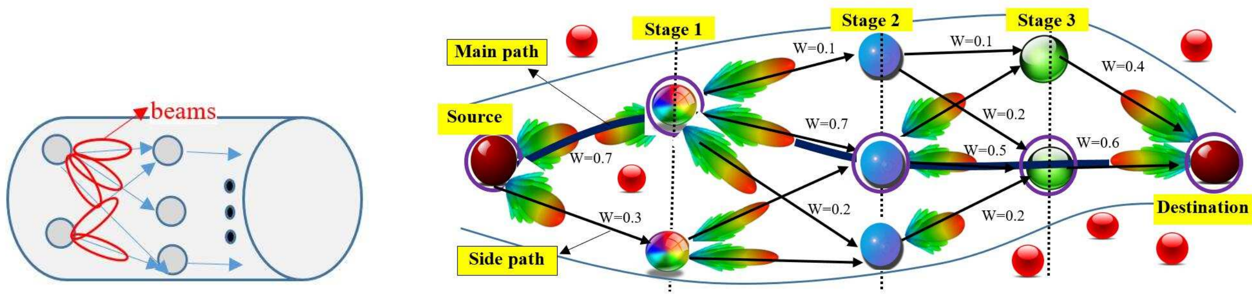
Thus, the encoded symbols are $C = [c^s_1, c^s_2, c^s_3, c^s_4, c^o_1, c^p_1,]$, where the OCS is $c^o_1 = m_1 \oplus m_2 \oplus m_3 \oplus m_4$, and the PCS is $c^p_1 = m_3 \oplus m_4$. If segments m_1 and m_2 travel through the main path, and segments m_3 and m_4 travel through the side paths, the PCS c^p_1 provides extra protection for the side path symbols. However, if the traffic allocation varies significantly from stage to stage, it would be difficult to decide which packets should be provided extra protections. Under this circumstance, the PCS should be abandoned to use.

Since a generator matrix yielding systemic coding is shared between the sender and receiver, the computational complexity of both sides is low. For each batch, the maximum computation of the sender is $(n - k) * (k - 1) * h$ times of bit-wise XOR operation (to obtain $(n - k)$ redundant symbols). For the receiver, if no original packets lost, there is no decoding computation involved. However, if packet loss happened and k intact symbols were collected, the decoder retrieves the lost packets through a maximum of $(n - k) * (k - 1) * h$ times of bit-wise XOR operation.

IV. ABC BASED TRANSPORT LAYER PROTOCOL

A natural solution to form a MBSA-oriented routing is to use a pipe to deliver packets, as shown in Fig. 3 (a). The i-th stage nodes can concurrently communicate with the $(i + 1)$-th stage nodes via different beams. By controlling the beam appropriately, the interference among beams are low enough for high-quality concurrent communications.

To build such a transmission pipe, we proposed a fence routing topology in our previous work [25], as shown in Fig. 3 (b). The transmission pipe consists of multiple stages, each of which has a flexible number of nodes, depending on how many nodes are available. Each beam has a weight value that measures the link quality. There is a main path in the pipe, which consists of high-quality links and defines the trajectory



(a) (b)

Fig. 3: Data Forwarding in a Pipe-Shaped Path
(a) Pipe Transmission (b) Data Forwarding in a Pipe

of the pipe. To build a fence routing topology, each node maintains a beam table, as shown in Table I. The parameters in beam table indicate the communication quality of each link, which are used to determine the coding pattern of the ABC scheme.

The proposed ABC based transport layer protocol is based on the fence routing topology. To achieve a smooth pipe streaming, the following strategies are adopted: (1) automatic batch size control, (2) postponed retransmission and window size adjustment, and (3) multi-beam traffic allocation. The details are elaborated in the following.

A. Batch Size Adjustment

The proposed ABC scheme controls the data transmission in a time-slot pattern. Assuming that the source node generates N packets in each second, and a *Standard Window Size* (SWS), b, is pre-selected, then, the time slot length is $T = b/N$, which means that a coding block, i.e., a batch, will be sent out by the sender within T seconds. Note that the window/batch time duration is proportional to the SWS — the more packets we pack in a batch, the longer window time would be. This is different from TCP, where the window time duration is determined by the Round Trip Time (RTT). Following that, the sender estimates the path quality and chooses appropriate parameters for ABC coding scheme, i.e., the batch size k and the coded block size n. To determine these coding parameters, two factors should be considered.

1) Coding Rate: Since the aim of redundant coding is to resist the random loss, the coding rate r should be proportional to the rate of the loss occurrences. Assuming that there are Q paths from the source to the destination, and for the i-th path, the symbol loss rate of the j-th stage is $e^S_{i,j}$, then, the loss rate of the i-th path is

$$e^R_i = 1 - \prod_{j=1}^{D}(1 - e^S_{i,j}), \tag{5}$$

where $i \in [1, 2, \cdots, Q]$, and D is the number of stages from the source to destination. Then, the coding rate is chosen as

$$r = \gamma \times [1 - \max(e^R_1, e^R_2, \cdots, e^R_Q)], \tag{6}$$

where γ is defined as the *risk factor*. If a conservative transmission strategy is expected, a smaller γ should be chosen. Thus, less raw packets and more redundancy will be included in a batch. On the contrary, if an aggressive transmission strategy is needed, a larger γ should be adopted to send more raw packets (with the price of less loss-resistance capability).

2) Block Size: Since the bandwidth of each link/beam changes with time, the capacity of the entire pipe varies. Thus, the amount of data sent by the source in each window should be dynamically adjusted according to the pipe capacity. To achieve this goal, for the i-th time slot, the source node collects the bandwidth information of each stage and computes the capacity evaluation factor as follows:

$$\kappa_i = \frac{\min(C^i_1, C^i_2, \cdots, C^i_D)}{C_E}, \tag{7}$$

where C^i_j is the bandwidth of the j-th stage in the i-th time slot ($j \in [1, 2, \cdots, D]$), and C_E is the expected bandwidth for the entire pipe, which are computed as:

$$C^i_j = \sum_{l=1}^{L} C^i_{j,l},$$
$$C_E = \frac{\sum_{l=1}^{i-1}\min(C^l_1, C^l_2, \cdots, C^l_D)}{i-1},$$

where $C^i_{j,l}$ is the bandwidth of the l-th link of the j-th stage at the i-th time slot. Then, the recommended window size of the i-th time slot, w'_i, is:

$$w'_i = \kappa_i b \tag{8}$$

where b is the Standard Window Size.

If there is no NACK, the real window size, w_i, is equal to the recommended window size, i.e., $w_i = w'_i$. Therefore, for the i-th time slot, the sender takes w_i original packets, generates $\frac{1-r_i}{r_i} w_i$ redundant symbols, and sends all $\frac{w_i}{r_i}$ symbols within a time slot T. This strategy allows the sender to dynamically adjust the sending rate according to the pipe's capacity.

B. Postponed Retransmission

The redundancy is only used for random loss resistance. If congestion happens, there might be so many missing symbols that the receiver cannot successfully decode the batch. In this situation, the lost packets will be retransmitted.

TABLE I: Beam Table

Beam #	Facing direction	Link quality	Queue size	Dest. node	Traffic type	Traffic rate	Traffic Capacity
Beam 1	North East [20°, 60°]	RSSI=-70dbm	90 % full	B	Video-VBR	1.5Mbps	2Mbits
Beam 2	South East [−20°, −60°]	RSSI=-60dbm	75 % full	F	Audio-CBR	0.6Mbps	600Kbits
Beam 3	East [−20°, 20°]	RSSI=-55dbm	50 % full	H	Data-ABR	1Mbps	1.6Mbits

1) Negative ACK: Since most batches can be successfully decoded even though random loss happens, using a *Negative ACK* (NACK) mechanism is more efficient than using ACK for feedback. Therefore, at the receiver side, if sufficient symbols (not necessarily all of the symbols) of a batch have been received, the receiver begins to decode, submits the decoded packets to the application layer, and ignores the late-arriving symbols of the same batch. However, if the receiver is unable to collect enough symbols for a successful decoding within a certain time, a NACK will be sent back to the sender, indicating the IDs of the fail-decoded packets. To fasten the feedback process, the NACK packets travel through the main path. Note that the NACK packets are tiny (<100 bytes), and can be easily piggybacked in reverse packet.

2) Window Time Control: Once receiving a NACK, two operations are performed by the sender: a) The sender encodes the failed packets as a single batch and retransmits it; and b) the window size is dramatically shrunk. Assuming that there are p packets needed to be retransmitted, the sender encodes these p packets as a new batch, postpones for λT seconds, and retransmits. Here λ is defined as *Congestion Adjustment Factor (CAF)*. The reason of holding the retransmission for some time is to allow the network clearing up the queues in the bottleneck nodes, thus relieving the congestion. Note that the CAF can be less than 1, which means that the sender waits for less than one window time before the retransmission.

Once the lost packets have been retransmitted, the sender doubles the window size until reaching the recommended size computed by (8). Assuming that a NACK claiming p lost packets is received by the sender in the j-th window, the size of the i-th window, w_i, is ($i = 1, 2, \cdots, j, j+1, \cdots$):

$$w_i = \begin{cases} w'_i, & for\ i \leq j \\ 0, & for\ j < i < j + \lambda T \\ p, & for\ i = j + \lambda T + 1 \\ \min(2w_{i-1}, w'_i), & for\ i > j + \lambda T + 1 \end{cases} \tag{9}$$

Fig. 4 exemplifies a scenario of window size variation, for which the CAF is $\lambda = 1$. It is assumed that there is no NACK in the first 6 windows, and the window sizes computed by (8) are 7, 8, 7, 9, 6, 7, respectively. However, a NACK claiming two lost packets is received by the sender during the 6th window. Thus the size of the 7th window is decreased to 0. Following that, the two lost packets are retransmitted in the 8th window, i.e., $w_8 = 2$. Assuming that the recommended sizes from the 9th to 12th windows are 7, 7, 7, and 8, respectively. Then according to (9), the actual sizes of these four windows are: $w_9 = \min(4, 7) = 4$, $w_{10} = \min(8, 7) = 7$, $w_{11} = \min(14, 7) = 7$, $w_{12} = \min(14, 8) = 8$.

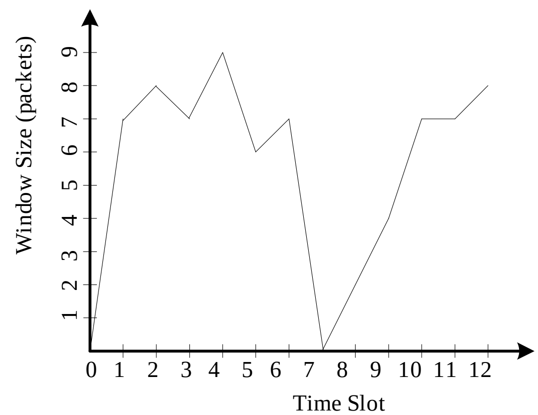


Fig. 4: Window Size Variation

C. Traffic Allocation

In both the RDC topology and the fence topology, the main path has the best transmission quality, i.e., the highest bandwidth and the lowest lost rate, while the transmission quality of the side paths varies significantly. Therefore, in the proposed scheme, each node, including the sender and all the intermediate nodes, allocates the symbols among all of its outlet links according to a capacity-and-reliability criterion.

First, each node collects path quality parameters of all of its outgoing links, including 1) the path type, which indicates whether the outgoing link belongs to the main path or the side path; 2) bit error rate of the link, e; and 3) link bandwidth, W. All the information can be collected from routing layer, as shown in Table I.

Meanwhile, each node records the time that an outgoing link finished the transmission of the last symbol. Using this information, the time that the next symbol will be finished sending through the same link can be calculated as:

$$t_{i,p} = t_{i-1,p} + \frac{B}{W_p} \tag{10}$$

where $t_{i,p}$ is the time that the i-th symbol will be finished transmission via the p-th link, $t_{i-1,p}$ is the time that the $(i-1)$-th symbol was finished sending through the p-th link, B is the packet size, and W_p is the data rate of the p-th link. Here $p \in [1, 2, \cdots, P]$, and P is the quantity of the outgoing links.

For symbol i, the optimal outgoing link, L_i^{opti}, is the one that yields the minimum transmission time, i.e.,

$$L_i^{opti} = L_{i,j} \ni t_{i,j} = \min(t_{i,1}, t_{i,2}, \cdots, t_{i,P}) \tag{11}$$

In such a way, the symbols are primarily allocated based on link quality. In addition, the following rules should also be followed when allocating traffic: 1) The checking symbols

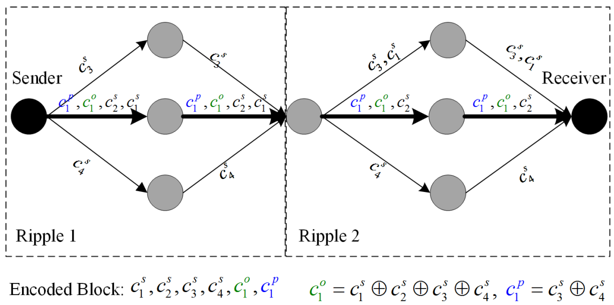


Fig. 5: Traffic Allocation upon RDC Topology

should always travel through the main path since they are the *skeleton keys* for all (or partial) original packets. 2) If the error rate of an outgoing link is larger than the coding rate r, the link is marked as invalid, and no symbol is allocated to it.

Fig. 5 shows an example of traffic allocation on a 2-ripple RDC topology with a generator matrix of (4). Assume that in ripple 1, the two side paths undertake 30% payload; while in ripple 2, the two side paths undertake 50% payload. The PCP packet, $c_1^p = c_3^s \oplus c_4^s$, provides extra protection for the side path symbols, c_3^s and c_4^s. Fig. 6 shows another prototype with the block size of 7 upon a 6-stage fence routing topology. Assume that the link quality significantly varies from stage to stage, thus, only the OCP is adopted as redundancy. Note that the OCP always travels through the main path, while the systemic symbols are dynamically allocated by each node.

V. Performance Evaluations

Since the proposed transport control is a novel scheme based on MBSA features, the existing network simulation tools such as OPNET/Riverbed, NS-3, etc., are difficult to use. We therefore built a customized simulation tool on Matlab, which can be easily converted to C/C++ codes for faster execution.

The following 4 transport layer models were simulated for performance comparison: 1) the proposed ABC-based protocol, 2) TCP, 3) MPTCP, and 4) Decongestion control based on rateless codes, as shown in Table II. For all models, MBSAs are equipped on each node. Since mobile wireless systems are subject to both frequency-selective fading (due to multipath effect) and to time-selective fading (due to shadowing) [26], and the random packet loss is caused by various reasons, normal distributions are adopted for both link capacity and link packet loss rate in our simulations. Specifically, for the main path, the link capacity is set to 2M bps and the random symbol loss rate is 1‰; for each side path in each second, the link capacity is $N(1, 0.1^2)$ Mbits/s and the random symbol loss rate is $N(1, 0.1^2) \times 0.01$, where $N(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. Note that the simulations were implemented for the aim of performance analyses and comparison in general mesh networks, thus the link parameters were set as moderate values. The proposed scheme can be implemented in 5G networks if the simulation platform is provided.

The performance of the proposed ABC scheme depends on 1) SWS, which decides the number of raw packets encoded in each batch, and 2) CAF, which decides how long the sender will postpone for retransmission. In the following, the impacts of the parameters on transmission performances are simulated and analyzed, and a comprehensive performance comparison is conducted finally.

A. Impacts of the Standard Window Size (SWS)

As discussed in Section IV, the value of time slot (T) depends on SWS (b), i.e., $T = b/N$, where N is the data generating rate (in packets number per second). Once the coding scheme has been determined, the larger the value of SWS is, the more raw packets would be encoded in a batch, yielding a higher coding rate and a lower resistance capability to random loss. Figs. 7 shows the trends of good throughput, delay, and retransmission ratio with the variations of SWS. Note that the good throughput present in this work is the raw packets, and the delay we calculated includes the waiting time at the receiver for the retransmitted packets. The experiments were conducted using the simulation model I (shown in Table II), running on the fence topology as shown in Fig.6. The CAF is set to 0.2, and the data sending rate is 2Mbits/s.

The experimental results show that when SWS is small, the delay and the retransmission rate are high and the good throughput is low. This is because that for a small SWS, much bandwidth is occupied by redundant data. Note that in our simulations, the data generating rate is set to a relatively high value compared to the network capacity. Therefore, too much redundancy in the pipe causes network congestion, which decreases good throughput and increases delay and retransmission rate. With the increase of the SWS, the throughput increases, the delay and retransmission rate decrease, which are the consequences of the higher coding efficiency. The best performance is achieved when the SWS is 10. However, when the SWS continues to increase, the performance deteriorates. This is because that the increase of the SWS causes the decrease of redundant ratio, which consequently results in a lower loss-resistance capacity. Therefore, the receiver has to rely more on retransmissions to retrieve the randomly lost packets. As a consequence, retransmission and window shrink happen more frequently, which causes lower throughput and higher delay and retransmission rate. The ABC-based congestion control eventually reach a relatively stable throughput of around 1.94Mbps and stable delay at around 250ms, as shown in Figs. 7.

Thus, the best strategy is that an appropriate redundancy rate is adopted, which provides loss-resistance capability for the random loss while leaves the congestion loss to be healed by retransmission. To achieve this goal, an appropriate Standard Window Size value should be chosen according to the pipe bandwidth and random loss rate.

B. Impacts of the Congestion Adjustment Factor (CAF)

Once a NACK is received by the sender, the transmission of all the packets will be postponed for λT seconds. Apparently, the larger the CAF value is, the better the congestion can be relieved since the network gets more time to clear up its queues. The performance variations with CAF for model I (based on the fence topology in Fig.6) are shown in Fig. 8.

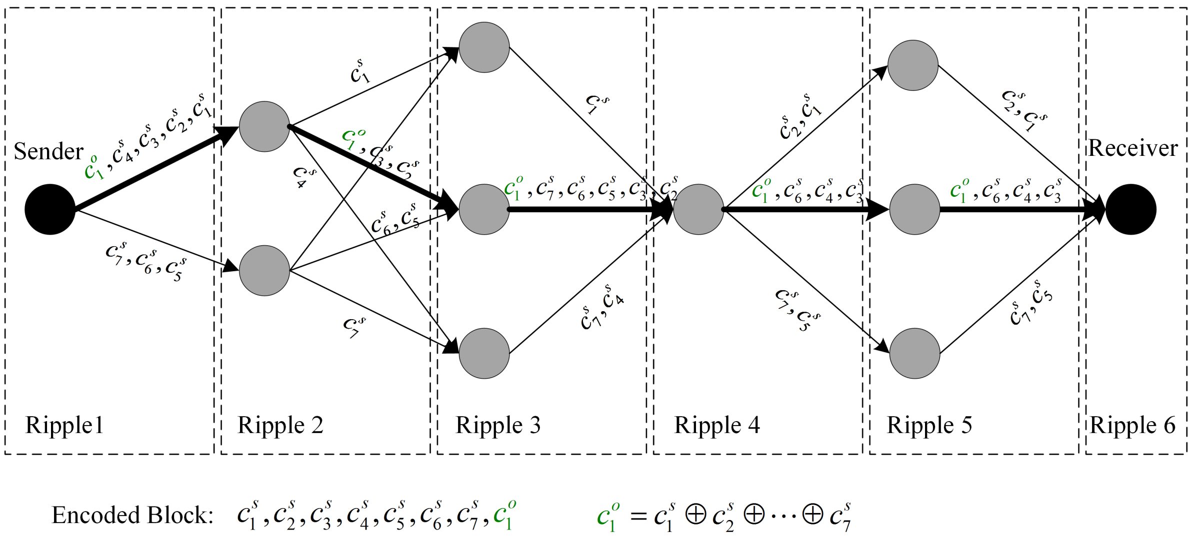


Fig. 6: Traffic Allocation upon Fence Topology

TABLE II: Simulation Models

	Transmission Scheme	Coding Method	Window batch Time	Window Size Adjustment	Traffic allocation
Model I	Proposed ABC scheme	$G = \begin{bmatrix} I_k \\ 1_{1\times k} \end{bmatrix}$	$T = b/N$	Automatically adjusted according to route quality and shrunk upon restransmission request	According to link quality
Model II	TCP protocol	N/A	One RTT	TCP sliding window control algorithm	Even allocation
Model III	MPTCP protocol	N/A	One RTT	TCP sliding window control algorithm	According to path quality
Model IV	Rateless Coding scheme	LT coding scheme	Transmission time of a coding block, flexible	N/A	Even allocation

In the experiments, the data generating rate is 2Mbits/s, the SWS is 7 and the time slot is $T = 7 \times \frac{2000\times8}{2\times10^6} = 0.056$ s.

It is shown that both a too small and a too large CAF values yield degraded throughput performance. This is because that when the CAF value is low, e.g., 0, the postponement time is too short for the network to relieve the congestion. On the other hand, if the CAF value is too large, the waiting time might be longer than the congestion relief time, potentially wasting bandwidth. Thus, an optimal CAF value should provide appropriate time for the network to relieve congestion without wasting too much time. In our simulation, a CAF value of 0.4 yields the best throughput performance.

Similarly, the delay and retransmission performances are worse when a too small CAF value is adopted. A too large CAF value also increases delay and retransmission rate. This is because that the sender holds the retransmission for longer time. The receiver may mistakenly think that the retransmitted packets were lost again and send more NACKs. Experimental results show that a CAF value of 0.2 yields the best delay performance and a CAF value of 0.4 provides the lowest retransmission rate.

C. Performance Comparisons

In this subsection, the transmission performances of the proposed scheme are compared to the-state-of-art transport layer protocols, i.e., Models II, III, IV in Table II. Figs. 9, 10 and 11 show the comparison results conducted upon the fence topology as shown in Fig.6, where the CAF is 0.2, and the SWS values of 7 and 10 are adopted respectively for the ABC scheme.

In the MBSA network, both the out-of-order problem and the packet loss are aggravated by TCP, thus, the average window size is low and the retransmission rate is high (sometimes retransmission happens upon late-arriving packets). Furthermore, each node distributes symbols randomly among its outlets. Since the transmission quality varies from link to link, some packets may suffer severe congestion at some links. Once a time-out event occurs, the source node shrinks the window size and decreases the sending rate for the entire pipe. These facts dramatically decrease the good throughput and increase the delay.

The MPTCP uses separate paths to transmit data and controls the window size of each path independently, thus increasing the good throughput and decreasing delay compared to TCP, as shown in Figs. 9-11. However, there are three problems in MPTCP: 1) Since the side paths have unstable transmission quality, packets may suffer severe delay at side path nodes, thus increasing the overall delay. 2) The payload distribution is implemented according to the average quality of each path, which is not efficient since the link qualities are different within each path; 3) the formation of the MPTCP topology needs more nodes and yields less flexibility.

For the rateless coding scheme, the redundancy coding

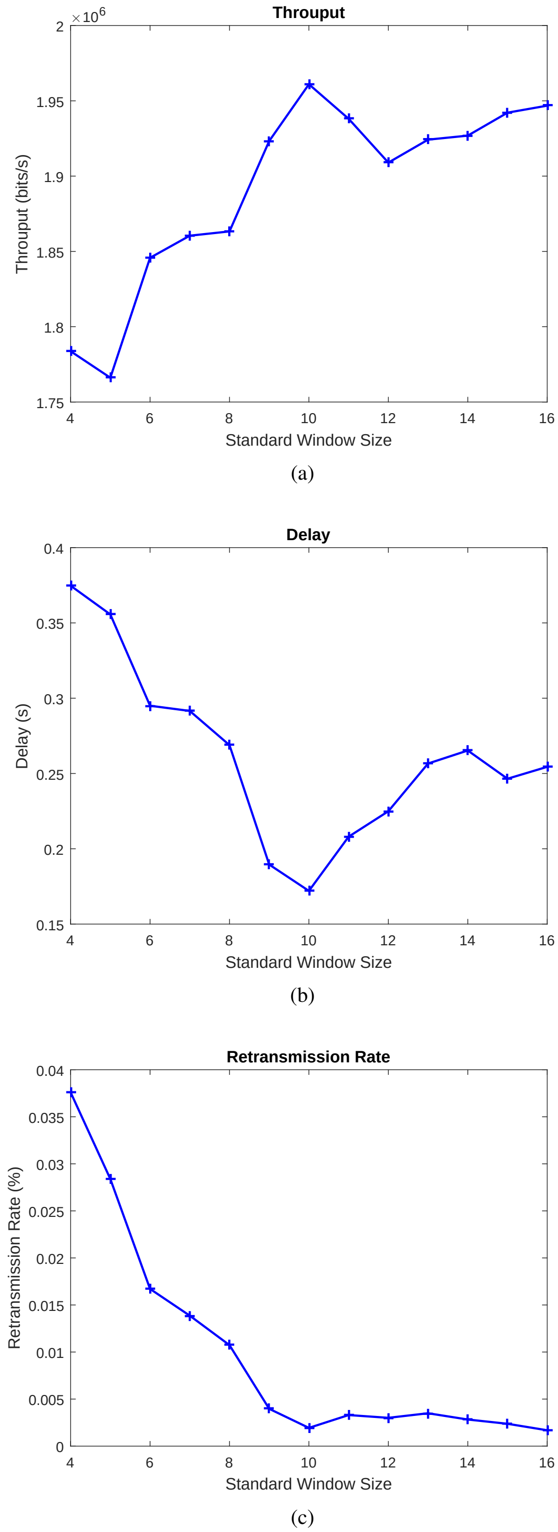


Fig. 7: Performances with SWS variations
(a) Throughput (b) Delay (c) Retransmission Rate

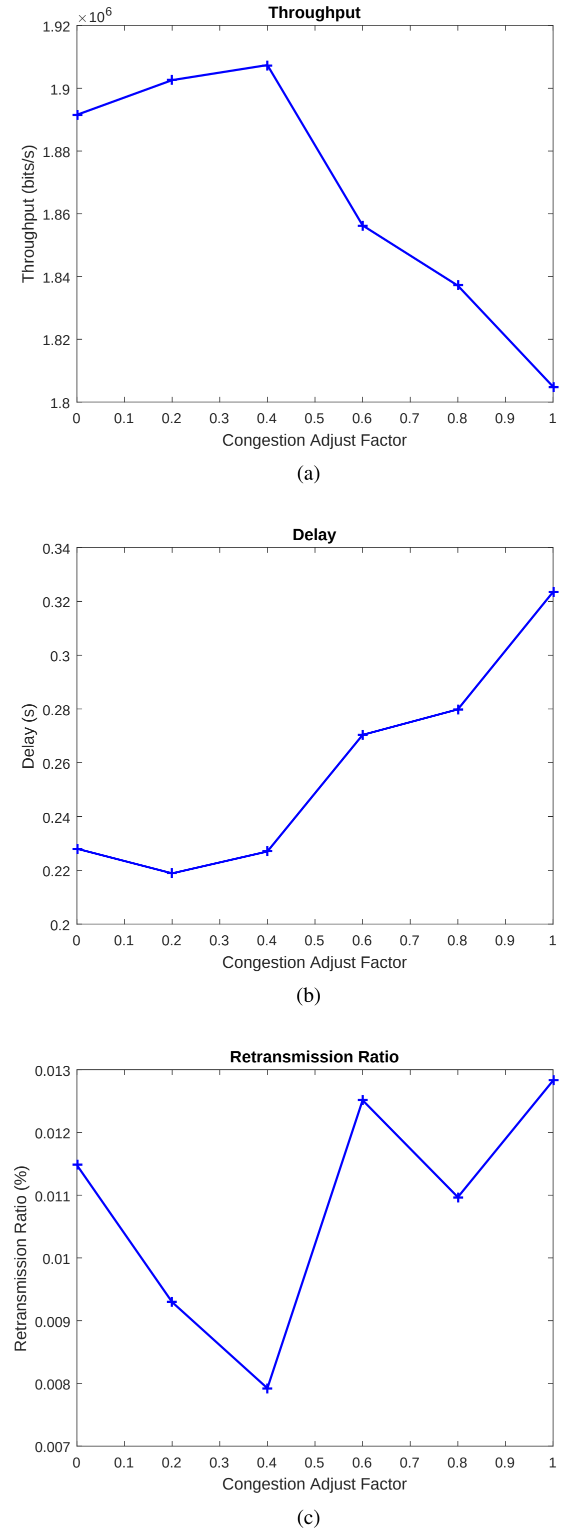


Fig. 8: Performances with CAF variations
(a) Throughput (b) Delay (c) Retransmission Rate

effectively compensates for the random packet loss. However, there are two problems when used in the MBSA networks: 1) The sender stops the encoding and the transmission of a block only when a positive ACK is received. However, the multi-hop network may cause late arrival of the ACKs, thereby wasting bandwidth; 2) if the packet loss is caused by network congestion instead of random loss, the congestion cannot be efficiently relieved since the protocol does not shrink the window size.

The proposed scheme adopts appropriate redundancy to

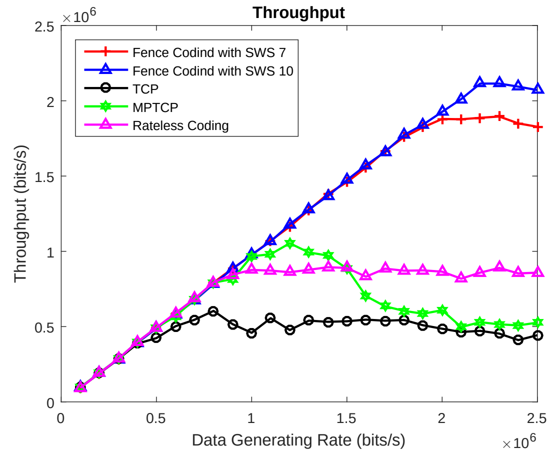


Fig. 9: Throughput comparisons

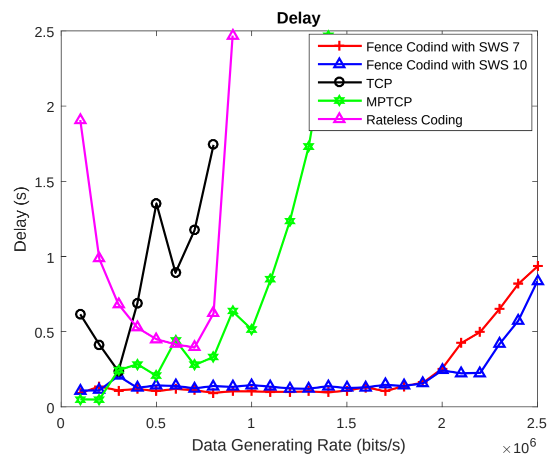


Fig. 10: Delay performance

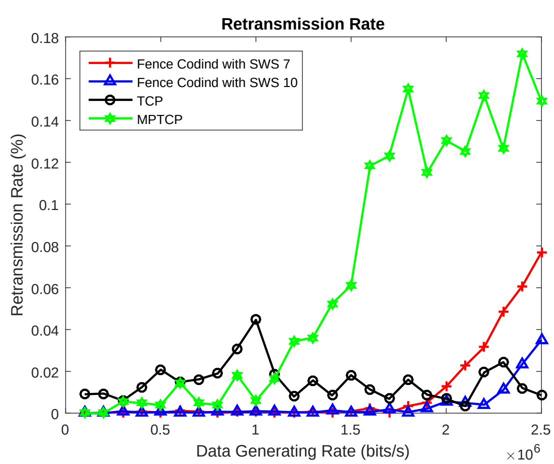


Fig. 11: Retransmission comparisons

compensate for random loss, making retransmission happens only upon congestion. In a multi-hop wireless network, a lower retransmission rate significantly increases good throughput and reduces delay, as shown in Figs. 9, 10 and 11.

VI. Conclusion

In this research, we proposed a novel transport layer protocol based on Adaptive Batch Coding, which differentiates the packet loss and deals with it using different strategies. For the random loss, the ABC-based protocol uses slight redundancy to compensate for the lost packets, and the coding scheme is automatically adjusted according to transmission quality. When the number of lost packets exceeds decoding threshold, the situation is diagnosed as congestion, therefore the lost packets are retrieved by retransmission, and the window size is shrunk. Simulation results show a significant performance improvement over TCP, MPTCP and de-congestion control schemes. Further work will focus on the dynamic network coding for multicast flows based on queueing theory and machine learning.

References

[1] M. N. Hamdy. (2015, Aug.). Multi beam antennas planning-limitations & solutions. CommScope. Hickory, NC, USA. [Online]. Available: http://www.commscope.com/Docs/Multi_Beam_Antenna_Planning_AN-109688-EN.pdf

[2] H. T. Chou, "An effective design procedure of multibeam phased array antennas for the applications of multisatellite/coverage communications," *IEEE Trans. Antennas. Propag.*, vol. 64, no. 10, pp. 4218-4227, Oct. 2016.

[3] J. Gan, Z. Guo, K. Sandlund, et al., "LTE in-band relay prototype and field measurement," in Proc. *IEEE VTC 2012 Spring*, 2012, pp. 1-5.

[4] Y. Qian, H. Asplund, J. Berg, and Z. Guo, "Non-Line-of-Sight 2.6GHz relay backhaul channel performance: field test and analysis," in Proc. *IEEE VTC 2012 Fall*, 2012, pp. 1-5.

[5] R. Taori and A. Sridharan, "Point-to-multipoint in-band mmwave backhaul for 5G networks," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 195-201, Jan. 2015.

[6] G. Zhang, T. Q. S. Quek, A. Huang, M. Kountouris, and H. Shan, "Delay modeling for heterogeneous backhaul technologies," in Proc. *IEEE VTC 2015 Fall*, 2015, pp. 1-6.

[7] D. C. Chen, T. Q. S. Quek, and M. Kountouris, "Backhauling in heterogeneous cellular networks: modeling and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3194-3206, Jun. 2015.

[8] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756-769, Dec 1997.

[9] S. V. Jansi Rani, P. Narayanasamy, and J. J. Shiny, "A cross layer approach for improving TCP performance using channel access information," in Proc. *ICCSP 2013*, 2013, pp. 587-591.

[10] R. Cheng and D. Deng, "Congestion control with dynamic threshold adaptation and cross-layer response for TCP Vegas over IEEE 802.11 wireless networks," *Int. J. Commun. Syst.*, vol. 27, no 11, pp. 2918-2930, Nov. 2014.

[11] R. Adams, "Active queue management: a survey," *IEEE Commun. Surv. Tut.*, vol. 15, no. 3, pp. 1425-1476, Oct. 2012.

[12] W. Ge, Y. Shu, L. Zhang, L. Hao, and O. W. W. Yang, "Measurement and analysis of TCP performance in IEEE 802.11 wireless network," in Proc. *CCECE 2006*, 2006, pp. 1846-1849.

[13] C. Chen, S. Y. Oh, P. Tao, M. Gerla, and M. Y. Sanadidi, "Pipeline network coding for multicast streams," in Proc. *ICMU 2010*, 2010, pp. 1-7.

[14] M. Luby, "LT codes," in Proc. *FOCS 2002*, 2002, pp. 271-280.

[15] Y. Wu, S. Kumar, F. Hu, Y. Zhu, and J. D. Matyjas, "Cross-layer forward error correction scheme using Raptor and RCPC codes for prioritized video transmission over wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 1047-1060, Jun. 2014.

[16] Y. Cui, L. Wang, X. Wang, H. Wang, and Y. Wang, "FMTCP: a Fountain code-based multipath transmission control protocol," in Proc. *ICDCS 2012*, 2012, pp. 366-375.

[17] L. Chen, T. Ho, M. Chiang, S. H. Low, and J. C. Doyle, "Congestion control for multicast flows with network coding," *IEEE Trans. Inf. Theory*, vol. 58, no. 9, pp. 5908-5921, Sept. 2012.

[18] Z. I. Kiss, Z. A. Polgar, C. Vinti, M. Varga, A. B. Rus, and V. Dobrota, ”Network coding-based congestion control at network layer: protocol design and evaluation,” *Int. J. Comput. Netw. Commun.*, vol. 3, no. 1, pp. 119-138, Jan. 2011.

[19] C. Zhang, Y. Chen, and C. Li, ”TCP adaptation with network coding and opportunistic data forwarding in multi-hop wireless networks,” *PeerJ Computer Science*, 2:e89, pp. 1-19, Oct. 2016.

[20] R. Hou, K. S. Lui, and J. Li, ”Joint congestion control and scheduling in wireless networks with network coding,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 7, pp. 3304-3317, Sept. 2014.

[21] H. M. Ruiz, M. Kieffer, and B. Pesquet-Popescu, TCP and network coding: equilibrium and dynamic properties, *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 1935-1947, Aug. 2016.

[22] TCP Extensions for Multipath Operation with Multiple Addresses, IETF RFC 6824 (Experimental), 2013.

[23] K. Bao, F. Hu, and S. Kumarl, AI-augmented, Ripple-Diamond-Chain shaped, rateless routing in wireless mesh networks with multi-beam directional antennas, *IEEE Transactions on Mobile Computing*, to be published.

[24] A. Loch, M. Hollick, A. Kuehne, and A. Klein, ”Practical OFDMA for corridor-based routing in wireless multihop networks,” in Proc. *LCN 2014*, 2014, pp. 1-9.

[25] L. Hu and F. Hu, ”Bio-inspired fence routing for multi-beam smart antennas,” *IEEE Transactions on Mobile Computing*, to be published.

[26] A. M. Tulino, G. Caire, S. Shamai, and S. Verdu, ”Capacity of channels with frequency-selective and time-selective fading,” *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1187-1215, Mar. 2010.