COded Taking And Giving (COTAG): Enhancing Transport Layer Performance over Indoor Millimeter Wave Access Networks

Zongshen Wu*, Chin-Ya Huang†, *Member, IEEE*, and Parameswaran Ramanathan*, *Fellow, IEEE*
*Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI, 53706 USA
†Department of Electronic and Computer Engineering,
National Taiwan University of Science and Technology, Taipei, 106, Taiwan
E-mail: zongshen.wu@wisc.edu, chinya@mail.ntust.edu.tw, parmesh.ramanathan@wisc.edu

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9839289

***Abstract*—Millimeter wave (mmWave) access networks have the potential to meet the high-throughput and low-latency needs of immersive applications. However, due to the highly directional nature of the mmWave beams and their susceptibility to beam misalignment and blockage resulting from user movements and rotations, the associated mmWave links are vulnerable to large channel fluctuations. These fluctuations result in disproportionately adverse effects on performance of transport layer protocols such as Transmission Control Protocol (TCP). To overcome this challenge, we propose a network layer solution, *COded Taking And Giving (COTAG)* scheme to sustain low-latency and high-throughput end-to-end TCP performance in dually connected networks. In particular, COTAG creates network encoded packets at the network gateway and each access point (AP) aiming to adaptively take the spare bandwidth on each link for transmission. Further, if one link bandwidth drops due to user movements, COTAG actively abandons the transmission opportunity by conditionally dropping packets. Consequently, COTAG actively adapts to link quality changes in mmWave access network and enhances the TCP performance without jeopardizing the latency of immersive content delivery. To evaluate the effectiveness of the proposed COTAG, we conduct experiments using off-the-shelf APs and network simulations. The evaluation results show that COTAG improves end-to-end TCP performance significantly on both throughput and latency.**

***Index Terms*—mmWave Links, TCP, Dual Connectivity, Immersive Content, Network Coding.**

I. Introduction

The demand for immersive content such as augmented reality (AR) and virtual reality (VR) is growing rapidly. These applications require high-throughput and low-latency network connectivity to support their rich, dynamic, and interactive content. They also need wireless connectivity for untethered access. In this paper, we assume that the wireless connectivity is primarily provided using millimeter wave (mmWave) based communication link. Note that, mmWave links have the potential to support high throughput because there is a large amount of spectrum available in this frequency band, e.g., Federal Communications Commission (FCC) has allocated 7 GHz (57 GHz – 64 GHz) of contiguous unlicensed spectrum for commercial and public use.

As the mmWave band brings the channel capacity to a new high level, there are also challenges. One challenge with the mmWave band is that the beams are substantially more directional than the conventional sub-6 GHz band. Furthermore, link bandwidth degrades more severely at the mmWave band than at the sub-6 GHz band [1], [2]. For example, the 60 GHz signal attenuates 15 to 20 dB for the first-order reflected path as compared to the line-of-sight (LOS) path. Although adaptive beam management can alleviate deterioration in channel conditions, detecting a change in the channel conditions as well as reforming the beams takes time. The delays incurred in adapting the beams adversely affect the end-to-end latency and the resulting end-to-end throughput, which in turn degrades AR/VR experience. Additionally, at certain frequencies especially around the 60 GHz band, the mmWave channels are highly susceptible to human blockages. Finally, although dual connectivity reduces end-to-end outage probability [3], the link quality imbalance in dual connectivity can impact upper-layer end-to-end throughput and latency.

AR/VR applications often use Transmission Control Protocol (TCP) to ensure reliable, sequenced delivery of packets from server to client. Due to the above characteristics of mmWave band and due to changes in user pose, location, and direction during AR/VR applications, TCP's performance often falls well short of its performance in a comparable fully wired network. This is because human-induced channel quality deteriorations often causes the data rate to fall substantially. As a result, the bits in transit from the TCP sender to the TCP receiver experience severe congestion, which is sensed by the TCP sender after a certain network delay. Even after the lower layer protocols have largely restored the data rates, the TCP sender cannot sense this increase. Instead, in order to preserve network fairness, it slowly strives to increase the sending rate. While the sending rate is slowly increasing, the end-to-end throughput is substantially below the available bandwidth.

Although there are several potential solutions to this problem, it is very difficult to avoid the inherent delay for solutions at the transport layer to work effectively. The motivation of this paper is to enhance the transport layer end-to-end performance for indoor applications like AR/VR in dual connectivity mmWave network by alleviating impacts of channel fluctuations and link quality imbalance. We propose a network layer technique called ***COded Taking And Giving (COTAG)*** in this paper. COTAG assumes that there is a dual wireless

connectivity to end user, through either two mmWave links or one mmWave link and one sub-6G link. An advantage of dual connectivity is that there will be no complete loss of connectivity unless both links are blocked simultaneously. COTAG encodes packets into linear combinations at the gateway of indoor wireless network and forwards these combinations of packets through the dual connectivity for decoding at the mobile device. During the forwarding, if spare bandwidth is available, COTAG also generates extra combinations to fully use the bandwidth. COTAG adaptively gives up transmission opportunity when one link does not have enough bandwidth and generates redundancy when there is adequate bandwidth. In this way, COTAG adapts to the changes in link quality and alleviates adverse effects of channel fluctuations and link quality imbalance to enhance TCP performance.

The rest of this paper is organized as follows. The related work is introduced in Section II. The proposed approach and algorithms are described in Section III. The evaluation results demonstrating the effectiveness of the proposed mechanism are presented in Section IV. Finally, the paper is summarized in Section V.

II. Related Work

MPTCP. Transmission Control Protocol (TCP) is the most common transport layer scheme for reliable, sequenced delivery of data. There are many variants of TCP. The recent variants such as NewReno [4], CUBIC [5], Compound [6], and BBR [7] are designed for networks with high bandwidth delay products (BDP). However, they are not well suited for dealing with large bandwidth changes due to fluctuations in the wireless channel quality. Moreover, although TCP variants for wireless networks have been studied for dealing with packet errors and single channel fluctuations [8], further enhancement is needed for mmWave based dual connectivity networks to mitigate impacts from not only packet loss, but also imbalance in link quality. The TCP variant most related to the solution in this paper is Multipath TCP (MPTCP) [9].

MPTCP is a transport layer protocol designed to better utilize the bandwidth available across multiple paths in order to increase the end-to-end throughput. It tends to work well when the multiple paths from the source to the receiver are mostly edge disjoint. If there are many shared links among the paths, then the effectiveness is diminished. The effectiveness of MPTCP has been evaluated in delivering across multiple interfaces of an end host in [10]. [10] shows that the effectiveness of MPTCP is not satisfactory when the bandwidth of mmWave link fluctuates significantly over time.

To alleviate this problem, MuSher [11] adjusts the packet forwarding ratio according to the transport layer throughput and buffer conditions across the multiple paths in MPTCP. The adaptive packet scheduling at the transport layer in Musher enhances MPTCP performance in dual connectivity network with mmWave (IEEE 802.11ad) and sub-6G (IEEE 802.11ac) links. As compared to Musher, this paper addresses a network layer enhancement to the dual connectivity network for AR/VR scenario, where the frequent user movements, rotations and blockages result in significant channel fluctuations.

SRNC. Spare-bandwidth Rate-adaptive Network Coding (SRNC) is an approach to enhance TCP performances in a wireless mesh based network with gigabit links. The scheme proposed in this paper is inspired by the concept of network coding in SRNC. Due to the mesh networks, there are many paths with possibly many links with available bandwidth to provide many opportunities for network coding in [12]. Since packets from TCP flows destined for different end hosts are cross-coded, the decoding also occurs in the wireless mesh network. The end host is delivered unencoded packets and it is completely oblivious to the network coding that occurred in the mesh. Such cross-coding is not applicable to scenarios considered in this paper.

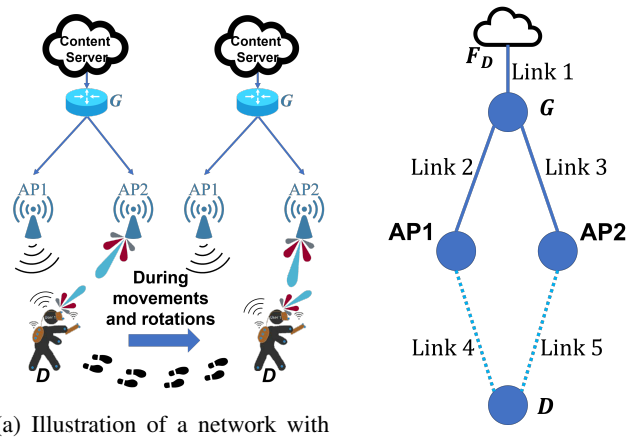
Other Related Work. Challenges of mmWave communications have been addressed at all layers of network stack. At physical layer, Sur et al. propose MUST to predict the best beam and to redirect user traffic over conventional WiFi when there is blockage in the mmWave link [13]. Although MUST focuses on the physical layer, it needs support from higher layers of the protocol to redirect user traffic. [3] discusses the challenges and tradeoffs of Ultra-Reliable Low Latency Communication (URLLC) considering massive MIMO and multi-connectivity. Using mmWave links cooperatively with LTE, 5G new radio (NR) are also being investigated in [10], [14]. Moreover, relaying video with mmWave based data plane and sub-6G based control plane is proposed in [15] to mitigate impacts of link failures in mmWave based relay networks. Non-Orthogonal Multiple Access (NOMA) supports the multi-user wireless networks for 5G NR in both power and code domains [16]. Besides the improvement achieved by NOMA for multi-user cases, enhancement at the higher layer is also important for more robust dual connectivity which may be supported by different base stations connected with the same user equipment.

III. Coded Taking And Giving (COTAG) Scheme

A. Overview

For simplicity of presentation, consider a simple topology formed by an immersive content server: a network gateway G, one sub-6G AP (AP1), one mmWave AP (AP2), and a mobile device D (e.g., AR/VR headset) (see Fig. 1a and 1b). Assume that the mobile device moves in a small geographic area in the vicinity of these two APs as a part of the immersive experience. In the immersive experience, the user moves from one place to another. During this process, the user induces changes in pose, location and direction, which result in mmWave channel quality variations.

Gateway G is on the route from the content server to the device D. It forwards the packets belonging to the traffic flows F_D that are associated with the delivery of the immersive content. Specifically, the traffic flows F_D may be comprised of multiple TCP flows, multiple flows can help achieve higher throughput because it is easier to deal with slower growth in the congestion window of each flow after packet losses or



(a) Illustration of a network with sub-6G and mmWave access

(b) Simple topology

Fig. 1: AR/VR dual connectivity network topology

reordering effects. Aiming to provide low-latency and high TCP throughput to the mobile device, when the network gateway G receives packets of F_D, it forwards them to the two APs. Device D is simultaneously connected to both APs. It may receive packets from both APs at the same time. COTAG is a network layer solution that complements the different strategies implemented at the physical and link layers. The key aspect of the proposed scheme is that the forwarding of the packets at G and the two APs does not occur in the traditional fashion. Instead, they forward using concepts of *coded forwarding*, *coded taking*, and *giving* as described below.

B. COTAG Illustration

Fig. 2 shows how sequenced packets at gateway G are delivered to the application using COTAG. For simplicity of explanation, this example assumes that G, AP1, AP2, and D are tightly synchronized. In reality, as detailed in later sections of this paper, the COTAG is self-clocking and synchronization between the nodes is not needed. COTAG introduces three concepts: *Coded Forwarding*, *Coded Taking*, and *Giving*.

Coded Forwarding: Instead of forwarding packets in its queue, each node (G, AP1, and AP2) forwards "network coded" packets, which are linear combinations of a cohort of packets from the same TCP flow. To create a cohort of packets, COTAG a priori selects a "buffering time" denoted by T. In Fig. 2, the value of T is 12. All packets arriving at G in the interval $[(k-1)T, kT)$, k = 1, 2, ..., are buffered until time kT. These packets form cohort "k". In Fig. 2, all packets that arrive at G in the interval [0, 12) form cohort 1. They are not forwarded until time 12. Similarly, packets which arrive at G in the interval [12, 24) form cohort 2, and so on. At time 12, G starts forwarding packets from cohort 1. The first packet in the queue from cohort 1 is a_1. Therefore, G linearly combines all cohort 1 packets from the "a" flow (a_1, a_2, a_3, and a_4) to generate a network coded packet $A_{1,1}$ and forwards it to AP1. The second packet in the queue from cohort 1 is a_2. Therefore, G once again generates a network coded packet $A_{1,2}$ by combining a_1, a_2, a_3, and a_4 and forwards it to AP2. After $A_{1,2}$ is transmitted, the third packet in the queue is b_1. Therefore, G linearly combines b_1 and b_2 and forwards it to AP2. Gateway G proceeds similarly until all packets in the queue have been considered or it is time for the next cohort. Similarly, nodes AP1 and AP2 also buffer incoming packets and forward linearly coded packets from the same cohort.

Coded Taking: G, AP1, and AP2 are not done with forwarding packets in a cohort after they have been through the buffer once during the time interval. For example, at time 18, G is done considering all the six packets in cohort 1 once. However, the time allocated for transmitting cohort 1 does not end till time 24. Between time 18 and 24, G forwards redundant packets. Specifically, it reconsiders the first, second, third, and so on packets in cohort 1, generates appropriate network coded packets and forwards them until either the time expires or a specified redundancy limit is achieved. Since a_1 is the first packet in this cohort, at time 18, G forwards $A_{1,5}$, which is a linear combination of a_1, a_2, a_3, and a_4. In this example, for cohort 1, $A_{1,5}$, $A_{1,6}$, $A_{1,7}$, and $B_{1,3}$ are redundant packets; there are six packets in cohort 1 but G forwarded a total of 10 packets for cohort 1 to AP1 and AP2. AP1 and AP2 also follow the same strategy, although the example does not show this situation.

Giving: Sometimes, depending on the link bandwidth and the number of packets in the cohort, G, AP1, and AP2 may not have adequate time to even consider all packets in the cohort at least once. In this case, "Giving" is invoked. Unlike in conventional forwarding, G, AP1, and AP2 does not forward all packets that are queued. If the time for a cohort runs out before all the packets are forwarded, the nodes simply discard the remaining packets and transition to handling the next cohort. For example, AP2 receives six packets of cohort 1 in the interval [12, 24). However, due to low data rate of the mmWave link between AP2 and D, AP2 is only able to forward two of these packets, in a network coded fashion, in the interval [24, 36). At time 36, AP2 discards all cohort 1 packets and starts forwarding packets from cohort 2.

C. COTAG Mechanism

Specifically, the proposed scheme COTAG has a parameter T that represents the scheme interval. The COTAG scheme applies interval parameter T to arrange arrived packets in cohorts for Coded Forwarding, Coded Taking and Giving. All packets of F_D that arrive at G in the time interval $[(k-1)T, kT)$, are queued and forwarded only in the interval $[kT, (k+1)T)$, for integer $k \geq 1$. Let us refer to the packets of F_D which arrive in the time interval $[(k-1)T, kT)$, as cohort k packets.

Operations at Gateway G: Forwarding at G in the interval $[kT, (k+1)T)$ occurs in the following way with the description in Algorithm. 1.

- **Coded Forwarding.** All cohort k packets are forwarded to the two APs in accordance to scheduling policy; some are forwarded to one AP, while the others are forwarded the other AP. However, they are not forwarded in the conventional way. Instead, when the scheduler is ready to

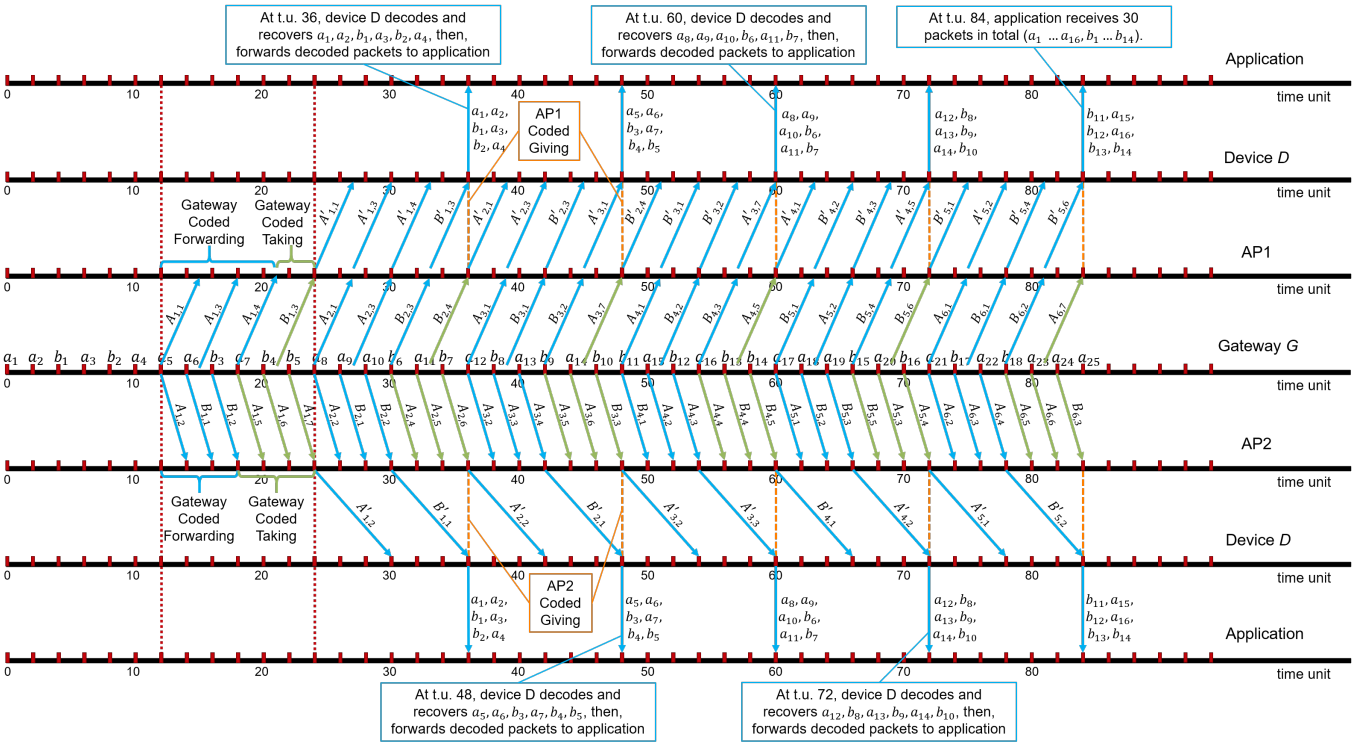


Fig. 2: Illustration of cases with COTAG

Algorithm 1 Algorithm executed by the network **Gateway** in **COTAG**.

In time interval $[kT, (k+1)T)$:

1: **if** $t == kT$, **then**
2: [Coding:] Discard all packets in F_D which arrive in $[(k-2)T, (k-1)T)$.
3: [Coding:] With n_k packets arrived in $[(k-1)T, kT)$, create n_k random linear network encoded packets in F_D arriving in $[(k-1)T, kT)$.
4: [Coding:] Attach a label k to each encoded packet.
5: [Coding, Give:] Mark these network encoded packets as "High priority".
6: **end if**
7: [Coding:] Buffer all incoming packets in F_D.

On each outgoing link:

Transmit (as permitted by the packet scheduler) network encoded packets :

8: **if** the total number of transmitted packets of all outgoing links is less than n_k, **then**
9: [Coding:] Forward the "High priority" network encoded packets.
10: **else**
11: [Take:] Generate a random linear combined packet of packets in F_A.
12: Attach a label k, mark it as "Low priority", and forward it.
13: **end if**

forward a packet from F_D, it forwards a random linear combination, i.e.. network coding [17], [18], of all cohort k in the queue. Before forwarding, the packet header is modified to indicate that it is from cohort k and it is also tagged "high priority". It proceeds in this fashion either until all cohort k packets have been forwarded once or until the interval expires. Cohort k packets are not discarded from the queue immediately after forwarding; instead they are used for coded taking.

- **Coded Taking.** If additional time is available until $(k+1)T$, G also transmits redundant packets to both APs in accordance with scheduling policy. Note that, this additional time is present when the sum of the bandwidth available for F_D from G to the two APs exceeds the current TCP throughput for F_D, a common occurrence when the bottleneck is at the mmWave links. The redundant packets are obtained by random linear combination of cohort k packets. Prior to forwarding, the packet header is modified to indicate that it is from cohort k. These packets are also tagged as "low priority". At time $(k+2)T$, all cohort k packets are discarded.

Operations at Each AP: Algorithm. 2 describes the operations at each AP. Cohort k packets are queued and not forwarded until the first cohort $(k+1)$ packet arrives. Forwarding occurs with following mechanism.

- **Giving.** Forwarding of cohort k packets occurs immediately after the arrival of the first cohort $k+1$ packet and *only until* the first cohort $k+2$ packet. As soon as the first cohort $k+2$ packet arrives, the remaining cohort k packets in the queue are discarded, even if they were never forwarded. This important aspect of the proposed schemed is called *Giving*, because unlike conventional routers, it may give away packets without forwarding when bandwidth is not available.

Algorithm 2 Algorithm executed by the **mmWave Access Point** in **COTAG**.

When a packet in F_D arrives on incoming link l:

if buffer is full, **then**
  if incoming packet is marked as "Low priority", **then**
    [Give:] Discard the incoming packet.
  else if all packets in buffer are marked as "High priority", **then**
    [Give:] Discard the incoming packet.
  else
    [Give:] Discard the latest received "Low priority" packet in the buffer.
    Enqueue incoming packet.
  end if
else
  Enqueue incoming packet.
end if
Let L' be the smallest unprocessed label of all F_D packets in buffer.
Let L_l be the largest label of all F_D packets received on incoming link l.
if $L' \leq \min\{L_l : \text{incoming } l\} - 1$, **then**
  $k = L'$.
  [Coding:] Discard all packets in F_D in buffer with label smaller than k.
end if

On the outgoing link of each AP:

Transmit (as permitted by packet scheduler) linearly combined packets in F_D with label k in buffer no more than number of received packets of label k.
if the total number of packets sent by all outgoing links is less than n_h, **then**
  [Coding:] Duplicate and forward an unsent "High priority" packet with label k.
  [Give:] Conditionally forward the unsent "Low priority" packet with label k.
else
  [Take:] Create a random linearly combined packet of packets in F_D in buffer.
  [Take:] Attach a label k, mark it as "Low priority", and forward it.
end if

- **Coded Forwarding.** High priority cohort k packets are forwarded to mobile device D over the mmWave link in accordance with scheduling policy.
- **Coded Taking.** If additional time is available until first cohort $k+2$ packet arrives, AP also forwards linear combination of cohort k packets in the queue.

Operations at Mobile Device D: Algorithm. 3 describes "Algorithm Mobile Device", the operations executed by the mobile device D. Similar to AP_i, the mobile device D also follows the label value of the network encoded packets of F_D to determine which packets to decode. The mobile device D decodes packets with label k when each incoming link has received a packet in F_D with label greater than k, the smallest unprocessed label in buffer. If packets are successfully decoded, the unencoded packets are forwarded to the application layer for immersive content streaming. Otherwise, undecoded packets are discarded.

Algorithm 3 Algorithm executed by the **Mobile Device** in **COTAG**.

When a packet in F_D arrives on incoming link l:

if buffer is full, **then**
  if incoming packet is marked as "Low priority", **then**
    [Give:] Discard the incoming packet.
  else if all packets in buffer are marked as "High priority", **then**
    [Give:] Discard the incoming packet.
  else
    [Give:] Discard the latest received "Low priority" packet in the buffer.
    Enqueue incoming packet.
  end if
else
  Enqueue incoming packet.
end if
Let L' be the smallest unprocessed label of all F_D packets in buffer.
Let L_l be the largest label of all packets in F_D received on incoming link l.
if $L' \leq \min\{L_l : \text{incoming } l\} - 1$, **then**
  $k = L'$.
  [Coding:] Discard all packets in F_D in buffer with label smaller than k.
  [Coding:] Apply decoding algorithm on packets in F_D with label k.
  if the decoding algorithm succeeds, **then**
    [Coding:] Transmit the recovered unencoded packets on the outgoing links as per routing algorithm.
  else
    [Coding:] Discard undecoded F_D packets with label k.
  end if
end if

IV. Evaluation

The performance of COTAG is evaluated using a network simulator. The network simulator uses measurements from a wearable VR setup for link bandwidth fluctuations. The setup measures the link bandwidth fluctuations caused by changes in user poses, locations, and directions while participating in an immersive experience on an Oculus Quest 2 headset. Although the Oculus Quest 2 headset does not support IEEE 802.11ay mmWave wireless link, the wearable setup includes 802.11ay devices deployed to capture the effects of user movements on the mmWave link bandwidth. The simulations explore the impact of user movements on transport layer performance during the immersive VR experience. We compare the performances of MuSher and COTAG for different network delays and packet error rates (PER) in the network simulator.

A. Experimental Setup

Wearable real-time VR mmWave channel measurement system: We first build the wearable mmWave link system to measure the mmWave channel capacity at real time during the use of VR headset.

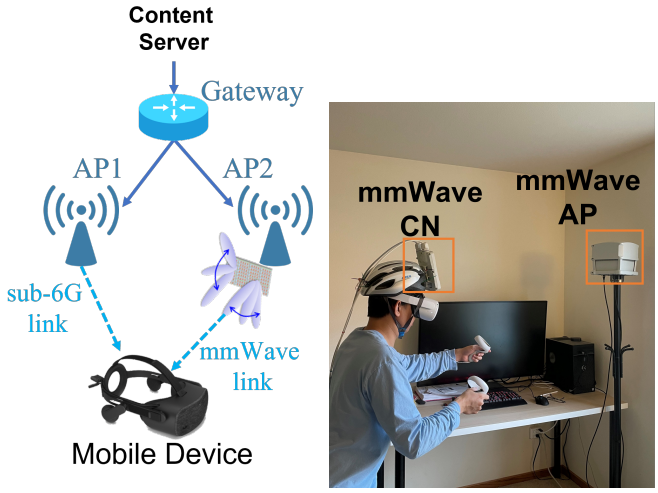
1) VR scenario setup: The dual connectivity scenario of mmWave network simulations are shown in Fig. 3a. This dual connectivity involves one sub-6G connection and one

mmWave connection with modern beamforming techniques. These two connections are supported by two different APs, which are connected to the same gateway. The mobile device associates with these two connections to establish dual link connectivity and receives packets from both mmWave and sub-6G APs simultaneously. Since the user movements are fairly small, the sub-6G connection seems to provide a relatively stable channel bandwidth.

Fig. 3b shows a picture of a user in an immersive VR experience. The setup augments Oculus Quest 2 headset with additional mmWave devices because the current headset does not support mmWave connections. The mmWave link channel quality fluctuations are obtained from measurements while the user is in the immersive experience. The user wears both the VR headset and the mmWave client node (CN) on the head and the mmWave channel capacity between the AP and the CN is measured in real time while the user is using VR applications. To measure the network-layer link bandwidth fluctuations in real-time which the user is participating in a VR experience, we use a well-known technique called packet trains [19].

2) Packet train based measurement of link bandwidths: Packet train is an extension of a concept called packet pair [19]. A packet pair measurement system is comprised of a sender and a receiver. The network route from the sender to the receiver goes through the link whose bandwidth is to be measured. The packet pair technique assumes that desired link has the smallest bandwidth among all the links in the route. The packet pair sender transmits two back-to-back measurement packets. The technique relies on the observation that often these two packets get queued back-to-back waiting for transmission on the desired link and then get transmitted back-to-back on the desired link. When this happens, the spacing between the two packets (called the time dispersion) is inversely proportional to the link bandwidth. Due to different measurement errors, the link bandwidth may not be correctly estimated one packet pair. However, a good estimate can be obtained from repeated packet pair measurements. When the link bandwidth is high, as in the case of mmWave links, the time dispersion is very small and the receiver clock may not be accurate enough to measure time dispersion accurately.

To overcome this problem, one can use a train of measurement packets, instead of a pair. This ideas is illustrated in Fig. 4. One packet train is a group of UDP packets having the same size. As shown in the figure, ideally, all the packets in the train are queued and then later transmitted back-to-back on the mmWave link. Once the receiver receives a packet train, it calculates the bottleneck bandwidth along the path according to the train length and the time dispersion in the bottleneck bandwidth. If L_{train} is the length of one packet train and the UDP packet size is S_{packet}, then relationship to link bandwidth can be shown to be $BW_{bottleneck} = (L_{train} \cdot S_{packet})/t_{dispersion}$. The errors in measurement depend on the length of the packet train and the length of each packet in the train. Long packet trains and long packets tend to have smaller errors. However, they also increase the overhead of measurement. Through targeted simulations where one could quantify the errors, we chose the following paramters: L_{train} = 30 packets, S_{packet} = 1500 Bytes. Since the link bandwidth fluctuates, we chose to repeat the packet train measurement once every 10 ms. Note that, overhead of this measurement is only 36 Mbps for an approximately 1 Gbps link. The trace of channel measurements from this setup was input to simulations in the form of data rates.



(a) Dual connectivity topology (b) Augmented VR setup

Fig. 3: mmWave channel measurement setup

B. Performance Evaluation

We set different network delays and packet error rates (PER) and measure the short-term TCP throughput and latency of MuSher and COTAG during the burst of immersive data transmitting in real mmWave channels. In simulations, TCP-CUBIC [5] is used to assure reliable, sequenced delivery of packets with the packet size of 128 bytes. The channel bandwidth observed at the network layer is set in the form of data rate value and the bandwidth changes according to the packet train measurement interval, which is 10 ms, reflecting the network-layer channel quality with the impacts at MAC and physical layers. Results are shown in Fig. 5 and 6.

Results in Fig. 5a are based on simulations of MuSher and COTAG with zero PER in the dual connectivity network and with different network delays in the back end network. The error bars in the results show the range of 90% confidence intervals. In this case, the impacts are the fluctuations of mmWave channel bandwidth, the bandwidth unbalance between the mmWave channel and sub-6G channel and the delay in the back end network. It shows that as the delay in the network grows, the impact of the channel fluctuations

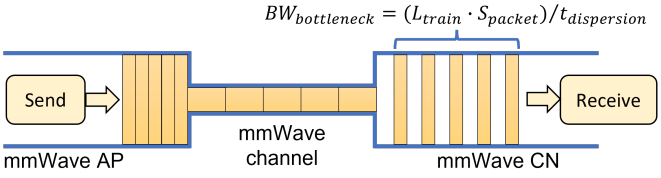
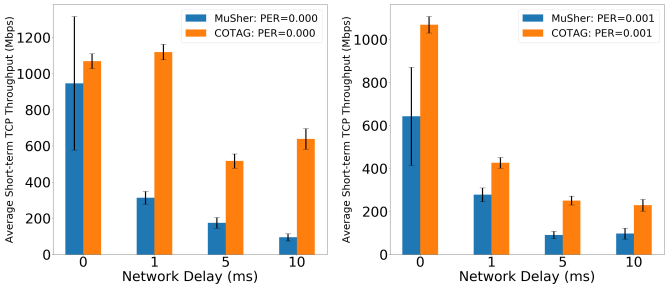


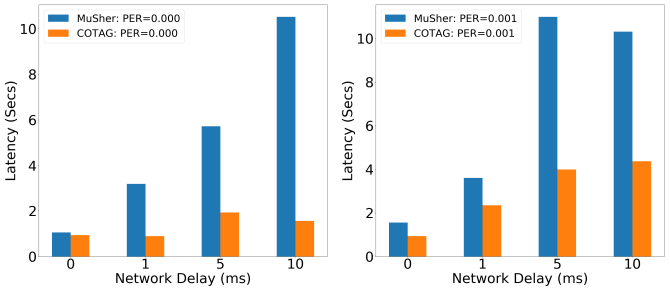
Fig. 4: Packet train illustration



(a) Results with zero PER (b) Results with 0.001% PER

Fig. 5: Short-term throughput with different PER

and bandwidth unbalance degrades TCP performance more severely. Simulations in Fig. 5b shows the average short-term throughput of MuSher and COTAG with 0.001% PER in wireless channels. The results show that with additional impact from PER, TCP performance is more vulnerable as the network delay increases. These results in Fig. 5 show that in real VR application scenario, COTAG significantly enhances TCP's short-term throughput.



(a) Results with zero PER (b) Results with 0.001% PER

Fig. 6: Data burst delivery latency with different PER

Moreover, the user movements and rotations can cause changes in the VR headset viewpoint during the use of immersive VR applications. In this case, a burst of data needs to be delivered for the new viewpoint. If the data is not delivered quickly, the user experience may be degraded. Thus, we evaluate the latency in delivering a burst of data. With the same PER and network delay setting, Fig. 6 presents the average latency for transmitting 1 Gb sequenced data when there is a data burst in the immersive application. In these simulations, COTAG has much smaller latency that MuSher.

V. Conclusion

Dual connectivity transmission based on mmWave and sub-6G wireless links is considered for delivering immersive content to AR/VR applications. To better enhance the robustness of dual connectivity transmission on network layer, we propose a COded Taking And Giving (COTAG) scheme in the presence of user movements, rotations and blockages. The simulation results show that considering network latency, packet errors, and link quality unbalance and fluctuations in dual connectivity, when traditional transport layer solutions are affected, COTAG can enhance TCP's short-term throughput and latency for short data bursts.

Acknowledgments

This work was partially supported by National Science Foundation grant CNS 1703389, U.S.A and by grant MOST 108-2221-E-011-058-MY3 of the Ministry of Science and Technology, Taiwan, R.O.C.

References

[1] S. Sun, T. Rappaport, M. Shafi, P. Tang, J. Zhang, and P. Smith, "Propagation Models and Performance Evaluation for 5G Millimeter-Wave Bands," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8422–8439, Sep. 2018.

[2] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of Millimeter Wave Communications for Fifth-Generation (5G) Wireless Networks–With a Focus on Propagation Models," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6213–6230, Dec. 2017.

[3] P. Popovski, Č. Stefanović, J. J. Nielsen, E. de Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A. Bana, "Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC)," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, Aug. 2019.

[4] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC Editor, RFC 2582, Apr. 1999.

[5] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-friendly High-speed TCP Variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, Jul. 2008.

[6] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A Compound TCP Approach for High-speed and Long Distance Networks," in *Proceedings of International Conference on Computer Communications (INFOCOM)*, Apr. 2006, pp. 1–12.

[7] D. Scholz, B. Jaeger, L. Schwaighofer, D. Raumer, F. Geyer, and G. Carle, "Towards a Deeper Understanding of TCP BBR Congestion Control," in *2018 IFIP Networking Conference (IFIP Networking) and Workshops*, May 2018, pp. 1–9.

[8] H. Haile, K.-J. Grinnemo, S. Ferlin, P. Hurtig, and A. Brunstrom, "End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks," *Computer Networks*, vol. 186, p. 107692, Feb. 2021.

[9] R. Barik, M. Welzl, S. Ferlin, and O. Alay, "LISA: A Linked Slow-start Algorithm for MPTCP," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[10] M. Z. Michele Polese, Rittwik Jana, "TCP in 5G mmWave Networks: Link Level Retransmissions and MP-TCP," in *IEEE INFOCOM WKSHPS*, Mar. 2017, pp. 1–6.

[11] S. K. Saha, S. Aggarwal, R. Pathak, D. Koutsonikolas, and J. Widmer, "MuSher: An Agile Multipath-TCP Scheduler for Dual-Band 802.11ad/ac Wireless LANs," in *The 25th Annual International Conference on Mobile Computing and Networking*, Oct. 2019, pp. 1–16.

[12] C.-Y. Huang and P. Ramanathan, "Network Layer Support for Gigabit TCP Flows in Wireless Mesh Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2073–2085, Oct. 2015.

[13] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "WiFi-Assisted 60 GHz Wireless Networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking*, Oct. 2017, pp. 28–41.

[14] G. Yang, M. Xiao, M. Alam, and Y. Huang, "Low-Latency Heterogeneous Networks with Millimeter-Wave Communications," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 124–129, Jun. 2018.

[15] Z. Li, Y. Shu, G. Ananthanarayanan, L. Shangguan, K. Jamieson, and V. Bahl, "Spider: Next generation Live Video Analytics over Millimeter-Wave Networks," Microsoft, Tech. Rep. MSR-TR-2020-17, May 2020.

[16] L. Dai, B. Wang, Y. Yuan, S. Han, I. Chih-lin, and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, Sep. 2015.

[17] D. Vukobratovic, A. Tassi, S. Delic, and C. Khirallah, "Random Linear Network Coding for 5G Mobile Video Delivery," *Information*, vol. 9, no. 4, Apr. 2018.

[18] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[19] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet Dispersion Techniques and Capacity Estimation," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 963–977, Dec. 2004.