VISIBLE: Virtual Congestion Control with Boost ACKs for Packet Level Steering in LWIP Networks

Thomas Valerrian Pasca Santhappan, Bheemarjuna Reddy Tamma, and Antony Franklin A.
Indian Institute of Technology Hyderabad, India
Email:[cs13p1002, tbr, antony.franklin]@iith.ac.in

***Abstract*—Tightly coupled LTE—Wi-Fi networks have emerged as a promising solution for improving capacity and coverage of wireless networks. Different architectures which realize this integration include LTE—Wi-Fi radio level interworking with IPSec tunnel (LWIP) and LTE—Wi-Fi Aggregation (LWA). A major issue with these architectures is that they do not exhibit expected performance when TCP is employed, i.e., TCP throughput decreases compared to using either LTE or Wi-Fi for transmission. Also, Multipath TCP (MPTCP) is inefficient while aggregating LTE and Wi-Fi links, especially when the link rates are incomparable. In this paper, we propose *VIrtual congeStion control wIth Boost acknowLedgEment* (VISIBLE) mechanism for LWIP networks which encompasses an efficient packet level traffic steering technique for steering Downlink traffic across LTE and Wi-Fi links of LWIP node and Boost ACK technique to reduce the number of duplicate ACKs (DUP-ACKs) delivered to the TCP sender. Unlike MPTCP, VISIBLE+LWIP uses both LTE and Wi-Fi links efficiently even if their link rates are incomparable and reduces unnecessary DUP-ACKs. We have developed VISIBLE+LWIP framework in NS-3 and compared its performance with the state-of-the-art MPTCP algorithms. We could observe that the proposed VISIBLE mechanism has doubled the throughput of basic LWIP and outperformed throughput of MPTCP by 37%. Also, it has enhanced the throughput by 30% as compared to basic LWA.**

I. Introduction

Smartphones are gaining momentum due to sophisticated Apps which are high data demanding. As per Cisco VNI forecast, mobile data requirement will grow 7x by 2021 compared to that in 2016 [1]. To address this ever growing demand, mobile operators seek a chunk of licensed spectrum, which is highly expensive, but on the other hand, unlicensed band has a large chunk of bandwidth which is free to use. Hence, operators are showing keen interest to deploy access networks that operate on unlicensed band to cater the traffic demand. LTE—Wi-Fi interworking will facilitate operators to use unlicensed band to serve indoor mobile users. 3GPP proposed various LTE—Wi-Fi interworking solutions from Rel.8 to Rel.12 which realize interworking at Evolved Packet Core (EPC). But, it is more challenging for an EPC based interworking solution to promptly react to variations in the channel conditions of LTE and Wi-Fi. To take a quick decision, finer control over LTE and Wi-Fi links is required. A finer control can be achieved only if the decision-making entity is closer to end devices. This has pushed the interworking decision making entity all the way from the EPC to Small cell eNodeB (SeNB) in which both LTE and Wi-Fi are integrated tightly at radio level. This radio level interworking is standardized by 3GPP in two ways: (a) LTE—Wi-Fi Aggregation (LWA) which integrates LTE and Wi-Fi protocol stacks at Packet Data Convergence Protocol (PDCP) layer and (b) LTE—Wi-Fi radio level interworking with IPSec tunnel (LWIP) [2] which realizes this interworking at IP layer. Among tighter level of interworking architectures, LWIP does not require any modifications at User Equipment (UE) and hence it can be deployed even for the existing UEs.

LWIP could be realized in two ways: collocated LWIP and non-collocated LWIP. In collocated LWIP, SeNB and Wi-Fi AP are located in the same node and tightly integrated at radio level. In case of non-collocated LWIP, an IPSec tunnel is setup between LWIP node and UE through Wi-Fi AP, which makes the communication over Wi-Fi link secure (refer Figure 1).

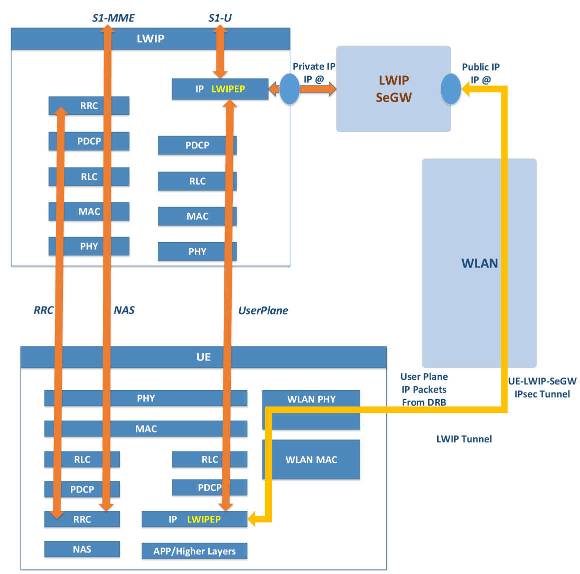

Fig. 1: 3GPP LWIP architecture in Release 13. [2]

Tighter interworking architectures like LWIP can harvest maximum benefit of link aggregation only when they do steering at the packet level. Packet steering in LWIP context refers to dynamically forwarding some packets of a downlink flow that arrive at LWIP node (i.e., SeNB) through LTE link and the rest through Wi-Fi link. But naive steering like round-robin at packet level across multiple links, especially when data rates of links are incomparable, creates problems for TCP traffic, as and when the packets received Out-of-Order (OOO) at the TCP

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

receiver lead to generation of DUPlicate ACKnowledgements (DUP-ACK). These unnecessary DUP-ACKs adversely affect the TCP congestion window growth and thereby lead to poor TCP performance [3]. DUP-ACKs are predominant when links are heterogeneous, where one link rate is much faster/slower than the other link. This problem is addressed to some extent in LWA architecture, where PDCP reordering procedure ensures packets are delivered in-sequence to the TCP layer at the receiver using Dual Connectivity (DC) [4] reordering procedure. But, during the reordering procedure, if the packets are buffered for longer duration, it has adverse effect on growth of TCP congestion window (for instance, TCP-New Reno, where congestion control is dependant on RTT) and hence results in poor throughput. In case of LWIP, this becomes even more challenging as packet steering is done at IP layer which lacks any packet reordering procedure. In this paper, we address the problem of handling packet level steering in TCP over the LWIP architecture by proposing a virtual congestion control mechanism (VIrtual congeStion control wIth Boost acknowLedgEment -VISIBLE). The proposed mechanism not only improves the throughput of a flow by reducing number of unnecessary DUP-ACKs delivered to the TCP sender but also sends Boost ACKs in order to grow the congestion window to reap in aggregate benefits of heterogeneous links. These Boost ACKs are pseudo ACKs for the actual TCP packets which are already in the queue of LWIP node. VISIBLE mechanism is implemented at LWIP node in such a way that it does not disturb the semantics of TCP.

II. Related Work

In this section, we discuss existing works related to two major problems that are being addressed: (1) Reducing trigger for spurious retransmissions and (2) Reducing OOO delivery of packets. First some existing works to control spurious retransmissions are summarized. Reducing spurious retransmissions involves the TCP sender to precisely differentiate congestion loss from OOO packet delivery. DOOR [5] detects OOO delivered packet by an additional ordering information in the ACK to avoid redundancy. It adds one byte TCP option field known as ACK Duplication Sequence Number (ADSN) to TCP ACK header. When the receiver sends the first ACK for TCP data segment, the ADSN option is initially marked as zero. It increments ADSN number when it sends a DUP-ACK for the same sequence number. Extension of DOOR is TCP-DOOR-TS [6] which uses TCP timestamp mechanism. The sender keeps track of sending time of the packet with respect to receiving time and relatively calculates the time stamp of every packet with the previously received one for detection of OOO packet. TCP receiver sets a field known as ooo_option_bit and informs the same to the sender. This method needs an option field to be explicitly set for its working. Eifel algorithm [7] explains about TCP robustness against spurious retransmissions. It has the facility of backward compatibility. It eliminates retransmission ambiguity and restores the transmission with the next unsent segment.

The forthcoming works include reordering at the receiver side. Delayed ACK [8] introduces a waiting time before the receiver generates a DUP-ACK. This delay in ACK generation provides an opportunity for the receiver to check the necessity of generating a DUP-ACK. The drawback of this method is that when an ACK is delayed in *slow start phase*, it may negatively affect the growth of TCP congestion window. Other re-ordering techniques such as Reordering Robust-TCP [9] and TCP-Packet Reordering [10] target to prevent persistent packet re-ordering from contrivedly activating congestion response by deferring packet retransmission and congestion response till the occurrence of packet loss.

Khadraoui *et al.* [11] studied the performance of TCP in a testbed of LWA and observed that LWA performs poorly even after packet reordering. Also it was found that PDCP reordering timer has negative impact on TCP and hence proposed a network coding technique to enhance the performance of LWA. A well known solution for aggregating multiple links/paths is Multipath TCP (MPTCP). The power of multiple sub-flows is used to harvest the aggregated bandwidth of links present. MPTCP receiver aggregates data from multiple sub-flows, reorders and then delivers it to higher layers. The problem with MPTCP is that it is not efficient in utilizing multiple links when they have incomparable data rates. Also, decision taken by MPTCP is based on the entire path into consideration even if the problem resides only with the last hop, which is true more often for wireless networks.

A. Contributions of the paper

The main contributions of this paper are as follows:

- We have designed a virtual congestion control mechanism (VISIBLE) with the objective of improving throughput of TCP flows by integrating efficient packet steering and ACK boosting techniques.
- We have analytically modelled the performances of different LWIP packet steering techniques.
- We have implemented the proposed VISIBLE mechanism in NS-3 and compared its performance with state-of-the-art aggregating solutions.

III. Proposed Work

Proposed VISIBLE mechanism addresses the problems faced by downlink TCP flows in LWIP networks. It is realized at LWIP node and includes two major components: (1) Packet steering technique and (2) Virtual congestion control mechanism. Packet steering takes care of forwarding the incoming packets of LWIP node into queues of LTE and Wi-Fi (i.e., RLC queue of LTE stack and MAC queue of Wi-Fi stack) at appropriate rates. Virtual congestion control mechanism helps the TCP sender to grow its congestion window by resolving DUP-ACK problem with the help of LWIP node, thereby improving throughputs of TCP flows.

A. LWIP packet steering techniques

The major cause for OOO packet delivery in LWIP networks is due to "speed of slowest link". This problem arises when packets of a TCP flow are split across two interfaces and a packet that is first sent through one interface arrives later than the ones that are sent through the other interface. This limits TCP throughput to the speed of slowest link. On one hand, steering all the packets onto one link avoids OOO delivery but it is inefficient in aggregating multiple links. On the other hand, steering packets in inappropriate fraction will cause more OOO deliveries. A better packet steering technique is required to split packets of the incoming flow across LTE and Wi-Fi links. Here we present two packet steering techniques for LWIP networks.

1) **Lowest RTT First (L-RTT):** This technique resembles MPTCP's default scheduler which first fills the congestion window of the link with the lowest RTT and then the link with higher RTT. In LWIP context, the transmit queue of interface with the lowest RTT is filled first before the other interface gets a packet. Note that RTT here factors delay of only the last (wireless) hop, not end-to-end path.
2) **Queue Depletion Rate (Q-Depl):** The rate of decrease in the length of each queue is used as the factor for steering the traffic across LTE and Wi-Fi links. The queue depletion rate is comparable to available data rate of an interface.

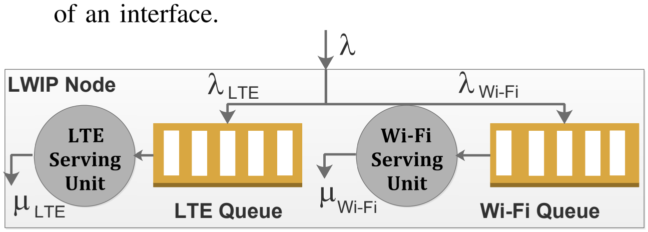

Fig. 2: Packet steering model at LWIP node

B. Analysis of packet steering techniques

In this section, the performance of L-RTT and Q-Depl packet steering techniques is measured in terms of their average queue lengths. In our analytical model, the packet arrival rate to LWIP node is assumed to follow Poisson distribution with mean arrival rate of λ. μ_{LTE} is the serving rate of LTE and μ_{WiFi} is the serving rate of Wi-Fi, which are exponentially distributed. The packet steering technique steers the incoming packets across LTE and Wi-Fi queues at rates λ_{LTE} and λ_{WiFi}, respectively. We have considered $\lambda = \lambda_{LTE} + \lambda_{WiFi}$. Figure 2 shows the packet steering model considered for the analysis.

1) Queue length in L-RTT: Lowest RTT aims at first filling the queue of the interface with the lowest RTT. For instance, let us assume that LTE link has the lowest RTT, then in LWIP context L-RTT involves filling LTE queue before Wi-Fi queue gets a packet. The queue sizes of LTE and Wi-Fi interfaces are represented as N and M, respectively. Here, the total length of the system corresponds to number of packets in the queue and that in the serving unit. So, when L-RTT is employed the total length of the system can be written as $L_{L-RTT} = L^L_{L-RTT} + L^W_{L-RTT}$. Here L^L_{L-RTT} and L^W_{L-RTT} correspond to the lengths of LTE and Wi-Fi queues, respectively when L-RTT packet steering is employed. In L-RTT, packet steering rate λ_{LTE} should be sufficient for filling the LTE queue first, which can be obtained by equating L^L_{L-RTT} to N. Using M/M/1/N/FIFO queue model, L_{L-RTT} can be represented as

$$\begin{aligned} L_{L-RTT} = &\left(\frac{\lambda_{LTE}}{\mu_{LTE} - \lambda_{LTE}} - \frac{(N+1) \times \left(\frac{\lambda_{LTE}}{\mu_{LTE}}\right)^{N+1}}{1 - \left(\frac{\lambda_{LTE}}{\mu_{LTE}}\right)^{N+1}} \right) \\ &+ \left(\frac{\lambda_{WiFi}}{\mu_{WiFi} - \lambda_{WiFi}} - \frac{(M+1) \times \left(\frac{\lambda_{WiFi}}{\mu_{WiFi}}\right)^{M+1}}{1 - \left(\frac{\lambda_{WiFi}}{\mu_{WiFi}}\right)^{M+1}} \right) \end{aligned} \quad (1)$$

2) Queue length in Q-Depl: Packet steering using queue depletion rate corresponds to the serving rates μ_{LTE} and μ_{WiFi}. Th incoming packets are steered onto LTE and Wi-Fi links in the ratio of the interface serving rates.

$$\lambda_{LTE} = \frac{\mu_{LTE}}{\mu_{LTE} + \mu_{WiFi}} \times \lambda; \quad \lambda_{WiFi} = \frac{\mu_{WiFi}}{\mu_{LTE} + \mu_{WiFi}} \times \lambda$$

Average queue length of L_{Q-Depl} based system is given by

$$\begin{aligned} L_{Q-depl} = &\frac{2 \times \lambda}{\mu_{LTE} + \mu_{WiFi} - \lambda} \\ &- \left[\frac{(N+1)\left(\frac{\lambda}{\mu_{LTE}+\mu_{WiFi}}\right)^{(N+1)}}{1 - \left(\frac{\lambda}{\mu_{LTE}+\mu_{WiFi}}\right)^{(N+1)}} \right. \\ &\left. + \frac{(M+1)\left(\frac{\lambda}{\mu_{LTE}+\mu_{WiFi}}\right)^{(M+1)}}{1 - \left(\frac{\lambda}{\mu_{LTE}+\mu_{WiFi}}\right)^{(M+1)}} \right] \end{aligned} \quad (2)$$

Figure 3 shows CDF of average queue length of the system when λ is varied from 0 to $(\mu_{LTE} + \mu_{WiFi})$. It can be clearly observed that if L-RTT is employed for packet steering, then a packet has to suffer a longer queuing delay compared to Q-Depl. This is because L-RTT is filling in one queue first which results in increase in its average queue length. At queue length 100, the first queue is completely filled, and packets are then steered onto the second queue, which reduces the rate of increase in the total queue length of the system. L-RTT can become comparable to Q-Depl only when one of the links serving rate becomes zero. Hence, in a system with heterogeneous links, Q-Depl will result in better aggregation benefits than L-RTT.

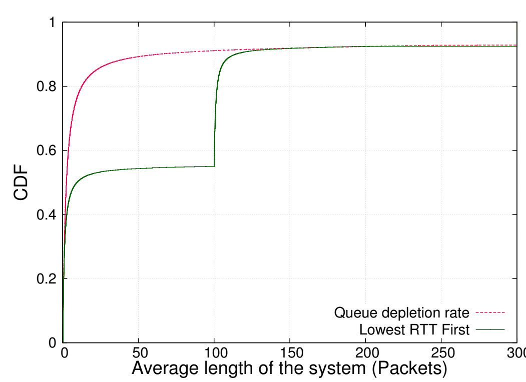

Fig. 3: Queue length of the system for L-RTT & Q-Depl

C. VIrtual congeStion control wIth Boost acknowLdgEment: VISIBLE mechanism

VISIBLE mechanism is employed at LWIP node which allows peeking into TCP header to collect necessary information for its operation. VISIBLE mechanism employs Q-Depl for steering the traffic across LTE and Wi-Fi links, and Boost ACK mechanism to minimize DUP-ACKs sent which trigger spurious retransmissions. Boost ACK is constructed by changing the ACK number field in the TCP DUP-ACK which makes the TCP sender understand that the packets are indeed delivered to the receiver successfully. Boost ACKs improve throughput of TCP flows, which otherwise would not have happened if DUP-ACKs are dropped at LWIP node or sent as they are to the TCP sender. VISIBLE not only reduces the delivery of DUP-ACKs to the TCP sender by boosting the DUP-ACKs but also holds the DUP-ACKs for some time in order to let the sender congestion window to grow. VISIBLE uses LTE interface of LWIP for both uplink and downlink TCP transmissions, but Wi-Fi interface is used only to send downlink TCP packets for which UE sends corresponding TCP ACKs through LTE. This ensures that there are no collisions in Wi-Fi link and thereby leads to efficient utilization of Wi-Fi link's capacity.

Illustration of VISIBLE: Figure 4 shows an illustration for working of VISIBLE mechanism. For a set of TCP segments transmitted by TCP sender, LWIP node steers IP packets containing TCP segments across LTE and Wi-Fi links based on packet steering algorithm implemented. In this example, Wi-Fi is shown as bottleneck link, and so it takes longer time to deliver a packet. In due course of time, the packets sent through LTE interface reach the receiver for which the receiver generates DUP-ACKs. LWIP node on reception of a DUP-ACK runs VISIBLE mechanism. VISIBLE transforms the DUP-ACK into a Boost ACK and sends that to the TCP sender or holds the DUP-ACKs for a while in order to let the ACKs flow in-sequence. Once the packet through Wi-Fi interface gets delivered to the receiver, LWIP node sends an ACK which acknowledges all the previous successfully delivered packets. Hence, the congestion window of the TCP sender is prevented from reducing unnecessarily which helps in significantly improving throughputs of TCP flows.

In the second part of Figure 4, a TCP segment with sequence number 150 is lost on LTE link. LWIP node again performs boosting and holding on reception of DUP-ACKs. As the DUP-ACKs arrival to LWIP node continues even after boost and holding phases, LWIP node concludes that the packet is actually lost and does a retransmission from its local buffer. Hence, TCP segment loss that occurred at the link level (LTE/Wi-Fi) is salvaged locally by the LWIP node.

The proposed VISIBLE is presented in Algorithm 1. Table I lists out the notations used in presenting VISIBLE mechanism. Following are the main features that VISIBLE mechanism encompasses (1) Rate of boosting DUP-ACKs (2) Holding DUP-ACKs and (3) Reducing packet losses. These main features are presented below in detail.

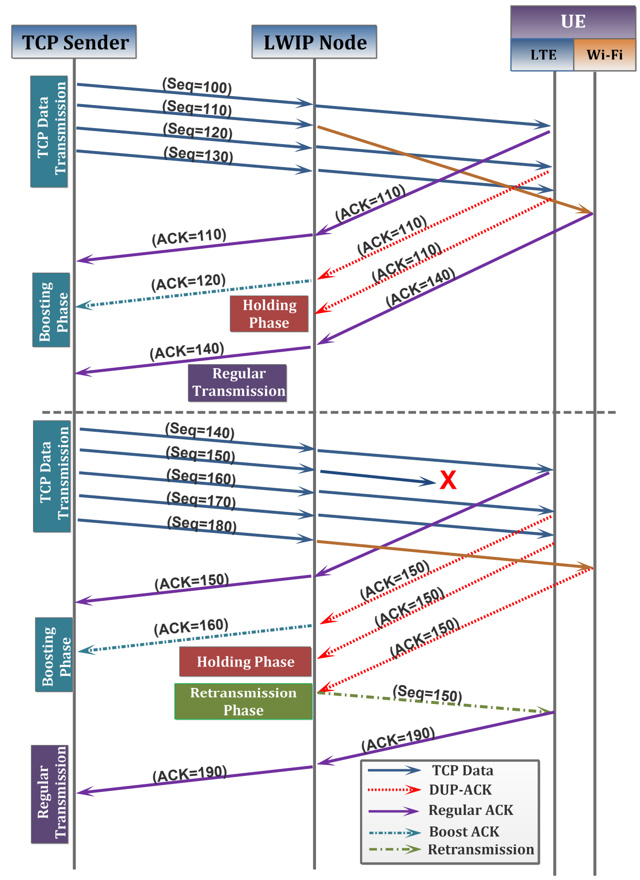

Fig. 4: Virtual congestion control procedure in VISIBLE

1) Boosting DUP-ACKs: During Boost ACK phase, a received DUP-ACK is transformed into a regular ACK by changing ACK number field of TCP header to a new ACK number. Note that this action is done only when the packet with sequence number corresponding to the new ACK number is already received by LWIP node from the TCP sender. The ACK with new ACK number conveys the sender that the packet got already delivered at the receiver, which makes the sender to grow its congestion window. Boost ACK phase is followed by a skipped ACK (also called as ACK holding) period in order to align the boosted ACK (new ACK) to become in sequence with actual received ACK from the receiver. The rate of boosting ACKs is a function of available buffer space in queues of LTE and Wi-Fi interfaces. Boosting of ACKs should be stopped well before the buffer space gets filled up. This ensures that the actual packets sent by the sender for the boosted ACKs will not get lost in the LWIP node due to buffer overflow phenomenon. The rate of boosting the ACKs is also reduced as the queue starts filling up. This is done in order to reduce the rate of growth of sender's congestion window so that it can sustain longer time without reduction due to packet losses induced by the full buffer.

2) Holding DUP-ACKs: During holding DUP-ACK phase, the actual DUP-ACKs when received at LWIP node are not sent to the TCP sender instead they are dropped by LWIP node. Dropping too many DUP-ACKs leads to *timeout* at the

TCP sender and also increases RTT, hence rate of regulating these DUP-ACKs is very crucial. We have controlled DUP-ACKs by taking into account various parameters, specially the holding rate depends on (a) available buffer space in queues of both LTE and Wi-Fi interfaces, (b) Link trip times of both LTE and Wi-Fi links, (c) Number of packets held currently, and (4) Holding time elapsed for DUP-ACKs. Link trip time of an interface in LWIP node corresponds to the time elapsed between sending a TCP packet to UE till getting the ACK for the same packet. The packet holding time and number of packets held are reduced by the factor of available packets in the buffer.

3) Handling packet losses: If a packet intended to the receiver gets lost on a wireless link (LTE/Wi-Fi), then the packet is retransmitted from the LWIP node instead of asking the actual sender to retransmit the packet. ϑ denotes the fraction of buffer size till boost and holding of ACKs can be done. α and β are weight fractions considered in an exponential moving average function. Typically α is low which emphasizes that when number of DUP-ACKs are increased, the holding threshold should be increased slowly. β takes a high value signifying that when the number of DUP-ACKs decreases the number of packets held should also be decreased rapidly, so that it reduces TCP timeouts happening because of longer holding. Trigger_Local_ReTx function retransmits the packet locally from LWIP node with RNTI ($\mathcal{R}$) and bearer id ($\mathcal{BI}$).

TABLE I: NOTATIONS USED IN VISIBLE MECHANISM

Parameter	Symbol
DUP-ACK Packet Counter (i^{th} flow)	$\mathcal{P}_i$
Last Received Ack Number at LWIP	$\mathcal{A}_i^r$
Last Sent Ack Number from LWIP	$\mathcal{A}_i^s$
Packet Sending Time	$\mathcal{T}_i$
Sequence Number of Packet Sent	$\mathcal{S}_i$
Radio Network Temporary Identifier (j^{th} user)	$\mathcal{R}_j$
Bearer ID (j^{th} user)	$\mathcal{BI}_{j,i}$
Number of DUP-ACKs Held	$\mathcal{PH}_i^a$
DUP-ACK Holding Time	$\mathcal{H}_i$
Timestamp of Last Holden Packet	$\mathcal{TH}_i$
Retransmission Counter at LWIP	$\mathcal{RT}_i$
Maximum number of Retransmissions from LWIP	$\mathcal{RT}^{max}$
Current Time	$\mathcal{CT}$
Available Buffer Size in LTE and Wi-Fi	$\mathcal{B}_{LTE}^a, \mathcal{B}_{Wi-Fi}^a$
Total Buffer Size of LTE and Wi-Fi	$\mathcal{B}_{LTE}^s, \mathcal{B}_{Wi-Fi}^s$
Initial sequence number	$\mathcal{I}_i$
Number of Flows in the System	$\mathcal{N}$
Boost Fraction	φ

Algorithm 1 : VISIBLE

TCP ACK Packet Received:

Update the TCP state information for all flows

if DUP-ACK of i^{th} flow && $\mathcal{A}_i^r > \mathcal{I}_i$ && $\mathcal{B}_{LTE}^a > \vartheta \times \mathcal{B}_{LTE}^s$ && $\mathcal{B}_{Wi-Fi}^a > \vartheta \times \mathcal{B}_{Wi-Fi}^s$ && $\varphi + \mathcal{P}_i < \frac{1}{\mathcal{N}} \times min(\frac{\mathcal{B}_{LTE}^a}{\mathcal{B}_{LTE}^s}, \frac{\mathcal{B}_{Wi-Fi}^a}{\mathcal{B}_{Wi-Fi}^s})$ && $\mathcal{RT}_i$==0 **then** ▷ Boost Acknowledgement Phase

  $\mathcal{TH}_i \leftarrow \mathcal{CT}$; $\mathcal{H}_i \leftarrow 0$; $\mathcal{P}_i \leftarrow \mathcal{P}_i + 1$

  $\mathcal{A}_i^r \leftarrow \mathcal{A}_i^r + (MSS \times \mathcal{P}_i)$

  $Modify\_ACK\_Number(Packet, \mathcal{A}_i^r)$; $\mathcal{A}_i^s \leftarrow \mathcal{A}_i^r$

else if DUP-ACK of i^{th} flow && $\mathcal{RT}_i == 0$ && $\mathcal{A}_i^r > \mathcal{I}_i$ &&$\mathcal{P}_i < \mathcal{PH}_i^a \times min(\frac{\mathcal{B}_{LTE}^a}{\mathcal{B}_{LTE}^s}, \frac{\mathcal{B}_{Wi-Fi}^a}{\mathcal{B}_{Wi-Fi}^s})$ && $\mathcal{H}_i < (\frac{1}{\mathcal{N}} \times ((LTE_{LTT} + WiFi_{LTT})/2) \times min(\frac{\mathcal{B}_{LTE}^a}{\mathcal{B}_{LTE}^s}, \frac{\mathcal{B}_{Wi-Fi}^a}{\mathcal{B}_{Wi-Fi}^s})$ && $\mathcal{B}_{LTE}^a > \vartheta \times \mathcal{B}_{LTE}^s$ && $\mathcal{B}_{Wi-Fi}^a > \vartheta \times \mathcal{B}_{Wi-Fi}^s$ **then** ▷ Holding Acknowledgement Phase

  if $\mathcal{TH}_i == 0$ **then**

    $\mathcal{TH}_i \leftarrow \mathcal{CT}$

  end if

  $\mathcal{H}_i \leftarrow \mathcal{H}_i + \mathcal{CT} - \mathcal{TH}_i$; $\mathcal{TH}_i \leftarrow \mathcal{CT}$; $\mathcal{P}_i \leftarrow \mathcal{P}_i + 1$

  $return$ ▷ Stops the DUP-ACKs

else if DUP-ACK of i^{th} flow && $\mathcal{RT}_i < \mathcal{RT}^{max}$ && $\mathcal{B}_{LTE}^a > \vartheta \times \mathcal{B}_{LTE}^s$ && $\mathcal{B}_{Wi-Fi}^a > \vartheta \times \mathcal{B}_{Wi-Fi}^s$ **then** ▷ Retransmission Phase

  $\mathcal{P}_i \leftarrow \mathcal{P}_i + 1$; $\mathcal{RT}_i \leftarrow \mathcal{RT}_i + 1$

  $Trigger\_Local\_ReTx(\mathcal{A}_i^r, \mathcal{R}_j, \mathcal{BI}_{i,j})$

else ▷ Regular Transmission

  if $Get\_ACK\_Number(Packet) == \mathcal{A}_i^r$ **then**

    $\mathcal{P}_i \leftarrow \mathcal{P}_i + 1$

  else

    if $\mathcal{P}_i > 0$ **then**

      if $\mathcal{P}_i > \mathcal{PH}_i^a$ **then**

        $\mathcal{PH}_i^a \leftarrow (1 - \alpha) \times \mathcal{PH}_i^a + \alpha \times \mathcal{P}_i$

      else

        $\mathcal{PH}_i^a \leftarrow (1 - \beta) \times \mathcal{PH}_i^a + \beta \times \mathcal{P}_i$

      end if

      $\mathcal{P}_i \leftarrow 0$

    end if

    $\mathcal{H}_i \leftarrow$0; $\mathcal{TH}_i \leftarrow$0; $\mathcal{RT}_i \leftarrow$0;

    $\mathcal{A}_i^r \leftarrow Get\_ACK\_Number(Packet)$

    $\mathcal{A}_i^r \leftarrow Get\_ACK\_Number(Packet)$

  end if

end if

$Send\_to\_S1U\_Socket(Packet)$

IV. Performance Evaluation

The performance of the proposed VISIBLE mechanism is evaluated in NS-3 with experimental parameters shown in Table II. We have developed LWIP module and LWA module in NS-3. Our simulation setup consists of an LWIP node and a set of associated UEs. Figure 5 shows the simulation setup. Each UE is having two downlink TCP flows from a Remote Server (RS). Bulk-send application is used to send TCP traffic. On receiving TCP segments from RS, VISIBLE+LWIP steers them across LTE and Wi-Fi links. UE on reception of each TCP packet generates a TCP ACK, which is sent in uplink to RS through LTE uplink and LWIP node. On reception of a TCP ACK, LWIP node runs Algorithm 1. In our setup, TCP New Reno is chosen as the underlying congestion control mechanism. This is because unlike TCP Cubic, the growth of congestion window of TCP New Reno is dependant on RTTs. Hence, holding ACK packet for longer duration will adversely affect the sending rate by the TCP sender. The

TABLE II: EXPERIMENTAL PARAMETERS

Parameter	Value
Number of LWIP Node and UEs	1, [1 to 4]
LTE SeNB bandwidth	10 MHz, FDD
LTE and Wi-Fi Tx power	20, 16 dBm
LTE antenna model	Isotropic antenna model
LTE path loss model	Friis propagation loss model
LTE SeNB scheduler	Proportional fair
Wi-Fi frequency, bandwidth	2.4 GHz and 5 GHz, 20 MHz
Wi-Fi standard	IEEE 802.11 a/b/g
Wi-Fi rate control algorithm	Adaptive auto rate fallback
Application	TCP BulkSend Application
TCP congestion control algorithm	TCP New Reno
Buffer size of LWIP Node	40 packets (per interface)

problem becomes more challenging if a packet is held for a longer time which could lead to TCP timeouts. We have evaluated VISIBLE mechanism to observe its performance in various scenarios as follows.

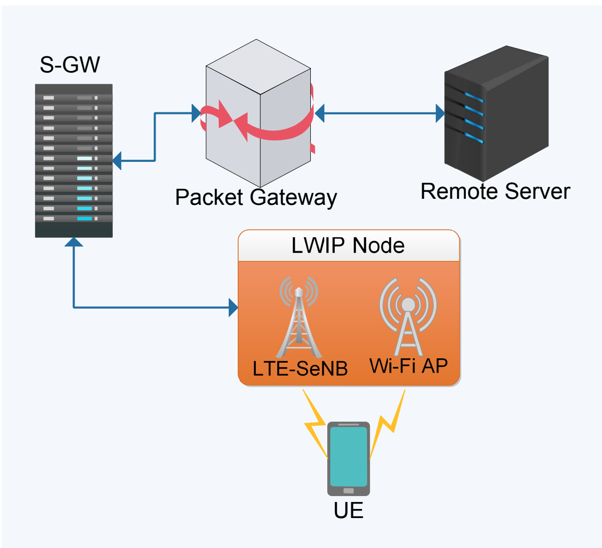

Fig. 5: Experimental setup

A. Performance of different phases of VISIBLE mechanism

Various phases of VISIBLE mechanism are evaluated in this section. In this experiment, for simplicity, we have considered a UE associated with an LWIP node which is having one downlink TCP flow from RS. Figure 6 shows the ACK number received at LWIP node and various operations performed on those ACK packets. When LWIP node receives DUP-ACKs, then Boost ACK phase gets triggered and VISIBLE boosts the ACK numbers. Further on receiving DUP-ACKs, holding phase is triggered. The DUP-ACKs are held by LWIP node either till the threshold time to hold these DUP-ACKs is met or till the number of DUP-ACKs threshold is met. Figure 7 shows the duration for which the DUP-ACKs are held and Figure 8 shows the number of DUP-ACKs that are held by VISIBLE. The threshold in both cases (viz., time to hold the DUP-ACKs and number of DUP-ACKS to be held) resembles a saw tooth pattern. Figure 9 shows the queue lengths of LTE and Wi-Fi interfaces. Retransmission of packets is triggered when DUP-ACKs count exceeds threshold packets to be held or the threshold holding time. Figure 8 shows the DUP-ACKs exceeding the threshold.

B. Performance of basic LWIP vs LWIP+VISIBLE

The growth of congestion window in case of basic LWIP (i.e., LWIP without employing VISIBLE mechanism at LWIP node) is heavily degraded by DUP-ACKs received which could be observed from Figure 10. In LWIP+VISIBLE, the congestion window grows faster which is not only due to avoiding DUP-ACKs but also due to boosting ACKs which helps the congestion window of the sender to grow faster. Figure 11 shows RTTs of both basic LWIP and LWIP+VISIBLE. The RTT for LWIP is constant, whereas in the case of LWIP+VISIBLE RTTs have an increase and decrease pattern i.e., this pattern coincides with congestion window's growth. When the congestion window increases, RTT goes higher because of holding of TCP ACKs for longer duration by LWIP node before the fast retransmit phase of TCP gets triggered (refer Figures 7 and 8).

C. Performance of LWIP+VISIBLE vs MPTCP

We have used open source MPTCP module [12] for NS-3 to perform this experiment. The simulation setup contains LTE and Wi-Fi networks (no interworking as in LWIP) connected to RS, and both RS and UE are MPTCP capable. Two downlink flows are generated between RS and UE in full mesh mode of MPTCP. The performance of LWIP+VISIBLE is compared with various congestion control algorithms of MPTCP viz., Coupled, Uncoupled and Link Increase Algorithm (LIA). Figure 12 shows the throughput improvement of LWIP+VISIBLE compared to MPTCP algorithms. LWIP+VISIBLE has improved throughput of the network by 55% as compared to MPTCP when IEEE 802.11b is used as Wi-Fi link. This is because when LTE and Wi-Fi link rates are incomparable, then MPTCP suffers from "the speed of the slowest link" problem, thereby MPTCP fails to achieve the aggregated throughput. LWIP+VISIBLE has improved the throughput by leveraging the potential of Boost ACKs. When IEEE 802.11g is used (here LTE and Wi-Fi link rates are comparable), then MPTCP gets the aggregation benefit. LWIP+VISIBLE also achieves comparable performance with MPTCP. When IEEE 802.11a is used, then LWIP+VISIBLE improves network throughput by 12% as compared to MPTCP.

D. Performance of different link aggregation architectures

In this experiment, an LWIP node is associated with 4 UEs and each UE is having two downlink flows. Here, we compare the performances of LWA, basic LWIP, and LWIP+VISIBLE. Figure 13 shows the throughputs achieved when different link aggregation architectures are used. LWA has achieved 50% throughput improvement compared to basic LWIP because of PDCP reordering procedure which it implements. But LWIP+VISIBLE has outperformed LWA by 30% this is due to boosting of ACKs which leads to better growth in congestion window and thereby improves the network throughput. VISIBLE mechanism (VISIBLE+LWIP) has almost doubled the throughput of basic LWIP.

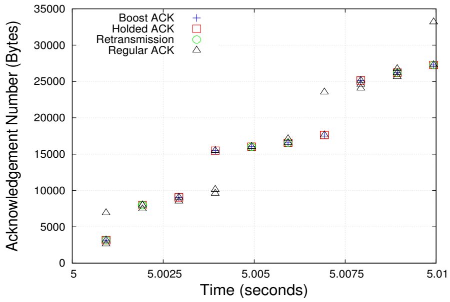

Fig. 6: Different phases of VISIBLE in LWIP network

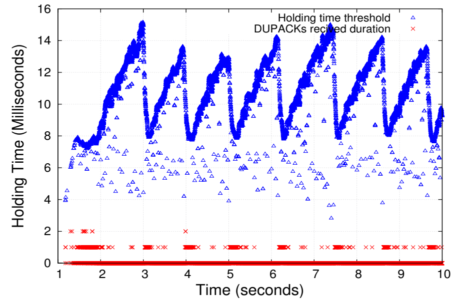

Fig. 7: Holding time of ACK packets in VISIBLE+LWIP

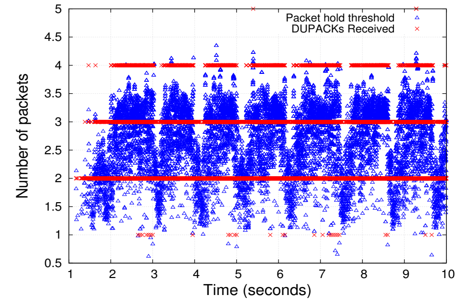

Fig. 8: Number of DUP-ACK held at LWIP node

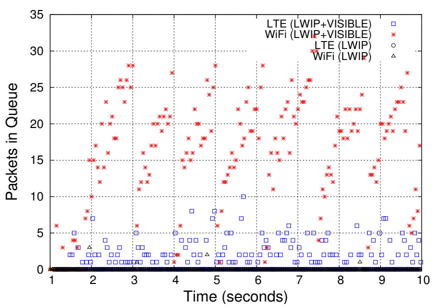

Fig. 9: Lengths of LTE and Wi-Fi queues

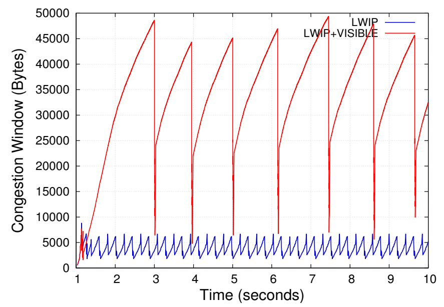

Fig. 10: Congestion window growth of basic LWIP and LWIP+VISIBLE

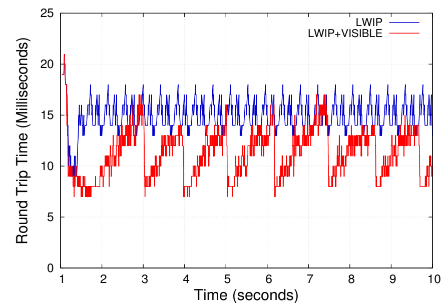

Fig. 11: RTT of TCP flow in case of basic LWIP and LWIP+VISIBLE

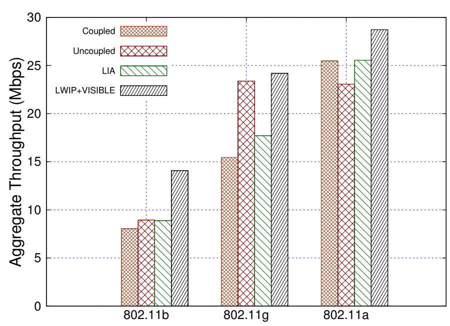

Fig. 12: Throughputs of MPTCP and LWIP+VISIBLE

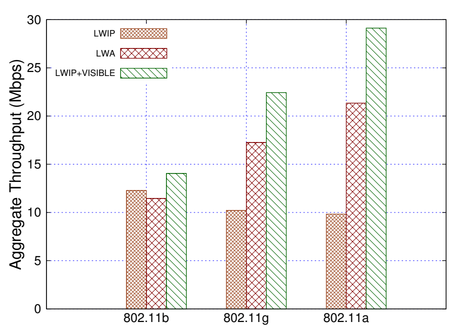

Fig. 13: Throughputs of LWA, LWIP, and LWIP+VISIBLE

V. Conclusions

In this work, we proposed VISIBLE mechanism for improving TCP performance in LWIP networks. The most crucial challenge is to let the congestion window of the sender to grow, which is achieved by sending Boost ACKs in a controlled fashion from LWIP node. The proposed VISIBLE mechanism has successfully aggregated multiple links even if they are of incomparable rates. LWIP node incorporated packet steering technique based on queue depletion rate and Boost ACKs. The proposed VISIBLE mechanism has out performed MPTCP based LTE—Wi-Fi integration by 37% and LWA architecture by 30%.

Acknowledgements

This work was partially supported by the projects ”Converged Cloud Communication Technologies” and ”Low Latency Network Architecture and Protocols for 5G Systems and IoT”, MeitY and SERB, Govt. of India.

References

[1] Cisco. (2017) A White Paper on Global Mobile Data Traffic Forecast Update. [Online]. Available: https://goo.gl/zY4nKI
[2] 3GPP, “LTE-WLAN Aggregation and RAN Controlled LTE-WLAN Interworking,” Tech. Rep. 36.300, 2016.
[3] P. Thomas *et al.*, “Architectural challenges and solutions for collocated lwip: Network layer perspective,” in *Proc. of NCC*. IEEE, March 2017.
[4] 3GPP. (2015) Study on Small Cell enhancements for E-UTRA.
[5] F. Wang and Y. Zhang, “Improving tcp performance over mobile ad-hoc networks with out-of-order detection and response,” in *Proc. of MobiHoc*. ACM, 2002, pp. 217–225.
[6] H. Mulugeta *et al.*, “Performance improvement of tcp using tcp-door-ts algorithm in mobile ad hoc networks,” in *Proc. of ICCT*. IEEE, 2011.
[7] R. Ludwig *et al.*, “The eifel algorithm: making tcp robust against spurious retransmissions,” *ACM SIGCOMM CCR*, vol. 30, no. 1, 2000.
[8] V. Paxson, “End-to-end internet packet dynamics,” in *ACM SIGCOMM CCR*, vol. 27, no. 4, 1997, pp. 139–152.
[9] M. Zhang, B. Karp, S. Floyd, and L. Peterson, “RR-TCP: a reordering-robust TCP with DSACK,” in *Proc. of ICNP*. IEEE, Nov 2003.
[10] S. Bohacek *et al.*, “A new tcp for persistent packet reordering,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, April 2006.
[11] Y. Khadraoui *et al.*, “Tcp performance for practical implementation of very tight coupling between lte and wifi,” in *Proc. of VTC*. IEEE, Sept 2016, pp. 1–6.
[12] MPTCP. [Online]. Available: www.github.com/mkheirkhah/mptcp