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

Thomas Valerrian Pasca, Bheemarjuna Reddy Tamma, and Antony Franklin.

Department of Computer Science and Engineering
Indian Institute of Technology - Hyderabad
Outline

1. Introduction
2. Challenges in Packet Level Steering
3. Literature Review
   - Reducing Spurious Retransmissions
   - Packet Reordering
4. Packet Steering Techniques
5. Proposed Solution
   - Outline of the proposed solution
   - VISIBLE Algorithm
6. Performance Comparison
Cisco: Mobile data traffic will grow 7x by 2021 compared to that in 2016 [1]
Mobile operators need to significantly improve network capacity to meet the user demand
Utilizing unlicensed band efficiently has gained operators interest

Wi-Fi offloading is a sweet spot for addressing the bandwidth crunch
We focus on LWIP for harvesting the benefits of unlicensed band
Radio Level Interworking Architectures

Figure: LWIP

Figure: LWA
Motivation

- Tighter LTE and Wi-Fi interworking architectures can harvest maximum benefit of link aggregation with packet level steering.
- Packet level steering may lead to Out-of-Order (OOO) delivery of packets at the receiver due to link heterogeneity.
- TCP receiver generates DUPLICATE ACKnowledgements (DUP-ACK) for OOO packets received.
- The unnecessary DUP-ACKs adversely affect the TCP congestion window growth and then lead to poor TCP performance.
- Efficient packet level steering and avoiding OOO delivery are necessary to reap in full benefits of LWIP.
Major issues with packet level steering in LWIP can be classified under two major categories:

1. Problem with packet reordering
2. Spurious Retransmissions
Reducing Spurious Retransmissions

- Enabling 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
- DOOR adds one byte TCP option field ACK Duplication Sequence Number (ADSN) into TCP ACK header
- Sender tracks the packet sending time and receiving time and relatively calculates the time stamp of every packet for detection of OOO packet.
- TCP receiver sets OOO option bit to inform the sender
Packet Reordering Techniques

1. Delayed ACK [8] introduces a waiting time before the receiver generates a DUP-ACK.

2. Delay in ACK generation provides opportunity for the receiver to minimize the necessity for generating DUP-ACK.

3. Delaying ACK in slow start phase will negatively affect TCP growth.

Packet Steering Techniques in LWIP

Lowest RTT First (L-RTT)
1. Employed as MPTCP’s default scheduler
2. First fills the congestion window of the link with the lowest RTT and then the link with higher RTT.

Queue Depletion Rate (Q-Depl)
1. The rate of decrease in the length of the queue is used as a factor for steering the traffic
2. Queue depletion rate is comparable to available data rate of a radio link
Figure: Packet steering model at LWIP node

Lowest RTT First:

\[
L_{L-RTT} = \left( \frac{\lambda_{LTE}}{\mu_{LTE} - \lambda_{LTE}} \right) - \frac{(N + 1) \times (\frac{\lambda_{LTE}}{\mu_{LTE}})^{N+1}}{1 - (\frac{\lambda_{LTE}}{\mu_{LTE}})^{N+1}} \right) + \left( \frac{\lambda_{WiFi}}{\mu_{WiFi} - \lambda_{WiFi}} \right) - \frac{(M + 1) \times (\frac{\lambda_{WiFi}}{\mu_{WiFi}})^{M+1}}{1 - (\frac{\lambda_{WiFi}}{\mu_{WiFi}})^{M+1}} \right)
\]  (1)

Queue with faster depletion rate:

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

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

\[
L_{Q-depl} = \frac{2 \times \lambda}{\mu_{LTE} + \mu_{WiFi} - \lambda} - \left[ \frac{(N + 1) \times (\lambda)}{(\mu_{LTE} + \mu_{WiFi})}^{N+1}}{1 - (\lambda)} \right] + \frac{(M + 1) \times (\lambda)}{(\mu_{LTE} + \mu_{WiFi})}^{M+1)}{1 - (\lambda)} \right]
\]  (2)
Comparison of different Packet Steering Techniques

**Figure**: Queue-cdf

- CDF of average queue length of the system when $\lambda$ is varied from 0 to $\mu_{LTE} + \mu_{WiFi}$
- L-RTT is filling the queue with lowest RTT which results in increase in its average queue length compared to Q-depl
Objectives of the Proposed Solution - VISIBLE

1. Enable growth of TCP congestion window
2. Reduce the triple DUP-ACK delivery to TCP sender
3. Support TCP packet retransmission at last hop

The three objectives are achieved through three different phases of the VISIBLE algorithm.

1. Boosting Phase
2. Holding Phase
3. Retransmission Phase
Virtual congestion control with Boost acknowledgement (VISIBLE)

- The objective includes High throughput, Reduction in OOO delivery and increased reliability
- Combination of packet steering technique and reordering technique to achieve higher throughputs
- No change to the TCP semantics
- No split in TCP session, Single congestion window at the sender
VISIBLE Algorithm I

TCP ACK Packet Received:
Update the TCP state information for all flows

1: if DUP-ACK of $i^{th}$ flow && $A^r_i > I_i$ && $B^a_{LTE} > \vartheta \times B^s_{LTE}$ && $B^a_{Wi-Fi} > \vartheta \times B^s_{Wi-Fi}$ && 
\[ \varphi + P_i < \frac{1}{N} \times \min\left( \frac{B^a_{LTE}}{B^s_{LTE}}, \frac{B^a_{Wi-Fi}}{B^s_{Wi-Fi}} \right) \text{ & & \(RT_i = 0\)} \text{ then} \]
▶ Boost Acknowledgement Phase

2: $T \mathcal{H}_i \leftarrow CT; \mathcal{H}_i \leftarrow 0; \mathcal{P}_i \leftarrow P_i + 1$
3: $A^f_i \leftarrow A^f_i + (MSS \times P_i)$
4: Modify_ACK_Number(Packet, $A^f_i$); $A^s_i \leftarrow A^f_i$

5: else if DUP-ACK of $i^{th}$ flow && $RT_i = 0$ && $A^r_i > I_i$ && $P_i < PH^a_i \times \min\left( B^a_{LTE}, B^a_{Wi-Fi} \right)$ && 
\[ \mathcal{H}_i < \left( \frac{1}{N} \times \left( (LTE_{LTT} + WiFi_{LTT}) / 2 \right) \right) \times \min\left( \frac{B^a_{LTE}}{B^s_{LTE}}, \frac{B^a_{Wi-Fi}}{B^s_{Wi-Fi}} \right) \text{ & & \(B^a_{LTE} > \vartheta \times B^s_{LTE}\) && \(B^a_{Wi-Fi} > \vartheta \times B^s_{Wi-Fi}\)} \text{ then} \]
▶ Holding Acknowledgement Phase

6: if $T \mathcal{H}_i = 0$ then
7: $T \mathcal{H}_i \leftarrow CT$
8: end if
9: $\mathcal{H}_i \leftarrow \mathcal{H}_i + CT - T \mathcal{H}_i; T \mathcal{H}_i \leftarrow CT; \mathcal{P}_i \leftarrow P_i + 1$
10: return
11: else if DUP-ACK of $i^{th}$ flow && $RT_i < RT^{max}$ && $B^a_{LTE} > \vartheta \times B^s_{LTE}$ && $B^a_{Wi-Fi} > \vartheta \times B^s_{Wi-Fi}$ then
▶ Stops the DUP-ACKs

12: $P_i \leftarrow P_i + 1; RT_i \leftarrow RT_i + 1$
13: Trigger_Local_ReTx($A^f_i$, $RT_i$, $B I_{i,j}$)
14: else
▶ Regular Transmission
**VISIBLE Algorithm II**

15: if $\text{Get\_ACK\_Number}(\text{Packet}) == A_i^r$ then
16: \[ P_i \leftarrow P_i + 1 \]
17: else
18: if $P_i > 0$ then
19: if $P_i > PH_i^a$ then
20: \[ PH_i^a \leftarrow (1 - \alpha) \times PH_i^a + \alpha \times P_i \]
21: else
22: \[ PH_i^a \leftarrow (1 - \beta) \times PH_i^a + \beta \times P_i \]
23: end if
24: \[ P_i \leftarrow 0 \]
25: end if
26: \[ H_i \leftarrow 0; TH_i \leftarrow 0; RT_i \leftarrow 0; \]
27: \[ A_i^r \leftarrow \text{Get\_ACK\_Number}(\text{Packet}) \]
28: \[ A_i^r \leftarrow \text{Get\_ACK\_Number}(\text{Packet}) \]
29: end if
30: end if
31: $\text{Send\_to\_S1\_U\_Socket}(\text{Packet})$
Experiment Setup

**Table:** EXPERIMENTAL PARAMETERS

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

Figure: Holding time of ACK packets

Figure: Number of DUP-ACK held

Figure: Lengths of LTE and Wi-Fi queues

Figure: Congestion window behaviour

Figure: RTT of TCP flows observed
Performance Comparison with VISIBLE Algorithm

Figure: Throughputs of MPTCP and LWIP+VISIBLE

Figure: Throughputs of LWA, LWIP, and LWIP+VISIBLE
Performance comparison with MPTCP

1. Two downlink flows are generated between RS and UE in full mesh mode of MPTCP.
2. Various congestion control algorithms of MPTCP viz., Coupled, Uncoupled and Link Increase Algorithm (LIA).
3. When LTE and Wi-Fi link rates are incomparable, then MPTCP suffers from "the speed of the slowest link" problem and hence fails to achieve the aggregated throughput.
4. LWIP+VISIBLE has improved the throughput due to its Boost ACKs mechanism.
5. When IEEE 802.11g is used (here LTE and Wi-Fi link rates are comparable), then MPTCP gets the full aggregation benefit.
6. 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.
VISIBLE has successfully aggregated multiple links even if they are of incomparable rates

LWIP node incorporated packet steering technique based on queue depletion rate

VISIBLE supports growth of congestion window of the sender to grow by sending Boost ACKs in a controlled fashion from LWIP node

VISIBLE has out performed MPTCP based LTE-Wi-Fi integration by 37% and LWA architecture by 30%
Acknowledgements

This work was supported by the project "Converged Cloud Communication Technologies"

Ministry of Electronics and Information Technology
Government of India
References I

References II

Performance Comparison

THANK YOU
For Further Queries Contact US

cs13p1002@iith.ac.in

www.thomasvalerrianpasca.in

https://github.com/ThomasValerrianPasca/