

Buffer size estimation of TP LINK TL-PA211KIT HomePlug AV adapters

Radika Veeravalli, Jason But, Grenville Armitage
Centre for Advanced Internet Architectures, Technical Report 130417A
Swinburne University of Technology
Melbourne, Australia
rveeravalli@swin.edu.au, jbut@swin.edu.au, garmitage@swin.edu.au

Abstract—We previously investigated aspects of the Netgear XAVB2001 HomePlug adaptors and found that their large internal buffers contributed to poor network behaviour under certain conditions. In this report we investigate the TPLink TL-PA211KIT HomePlug adaptor. We determined that the TPLink devices buffer about 546 Ethernet frames internally, which was observed to increase RTTs by up to 240-250ms (median 150-170ms) when using NewReno and CUBIC (loss-based) TCPs. Such increases in delay could result in poor performance for competing real-time traffic such as multimedia or gaming.

I. INTRODUCTION

This report extends on the work presented in [1] in which we investigated the buffer sizes and TCP performance of the NetGear XAVB2001 HomePlug adaptor. In this report we estimate the buffer size of the TP-Link TL-PA211KIT adaptor, and compare it with results for the NetGear adaptor.

HomePlug network devices allow consumers to deploy bridged Ethernet-like services across electrical mains powerlines, thereby allowing the provision of networking infrastructure where installation of new cables would either be expensive or difficult [2], [3]. The bandwidth afforded by the HomePlug technology is typically lower than the Ethernet infrastructure used to connect these devices to computers on the network. As such, the HomePlug link will typically form a bottleneck in the local network.

In order to minimise packet losses during bursts of traffic, bottleneck devices usually implement a form of buffering. As we observed in [1], these buffers can have a detrimental impact on the performance of the network under loaded conditions. Following on, we are motivated to explore the buffering capabilities of the TPLink adaptors.

Two methods to estimate the adaptor buffer size were used. One of these methods involved deploying SIFTR

[4] at the sender host. Through SIFTR, we were able to obtain exact values of $cwnd$, leading to more accurate estimates for buffering in the network. We found that the TPLink adaptors exhibited packet-based buffering with buffers of about 546 packets.

The rest of this report is structured as follows. Some background information and motivation for this work is provided in Section II, while a brief summary of our testbed configuration is provided in Section III. Section IV discusses the estimation of the buffer size of the TPLink adaptor, we discuss these results in Section V. We describe our conclusions in Section VI.

II. BACKGROUND INFORMATION

HomePlug AV Technology provides a system whereby two or more HomePlug adaptors are able to communicate over shared powerline infrastructure. Data packets are encoded on carrier frequencies for transmission over the powerline media. The achievable throughput is variable due to the noisy nature of the powerline environment [2], [3], [1].

HomePlug adaptors allow people to deploy data networking into environments where alternate means of communications is difficult to install (eg. older homes are difficult to re-wire or renters are unable to install networking infrastructure).

HomePlug adaptors typically bridge Ethernet packets onto the powerline media. As the throughput of the HomePlug link is often lower than the Fast Ethernet links to connected devices, the HomePlug link typically forms a local bottleneck for communications [1]. As such, the buffering capabilities of the HomePlug devices can have a significant impact on communications between two devices located in the same powerline environment. If excessive buffering (“buffer bloat”) is present in the HomePlug devices, this can lead to increased average queuing delays in the network. This can be problematic

to competing real-time application flows such as VoIP or online games.

In our previous work [1], we found that the NetGear adaptors implemented buffers of about 2720 Ethernet frames. Coupled with the implementation of Ethernet flow control, these buffers induced multi-second Round Trip Times (RTTs) in a network under the heavily loaded conditions generated by loss-based TCP Congestion Control (CC) algorithms. These results motivated us to see if other HomePlug devices displayed similar buffering characteristics.

III. EXPERIMENTAL TESTBED CONFIGURATION

Figure 1 depicts our experimental testbed, the configuration of the hosts and network topology are exactly the same as in [1]. The adaptors were deployed in different rooms to approximate a typical home environment. For this report, we are using the NewReno [5], CUBIC [6] and CDG [7] CC algorithms deployed on the sender host.

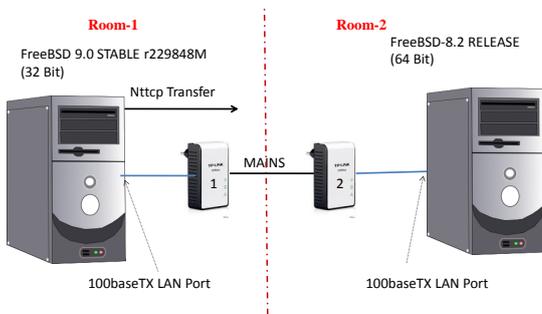


Figure 1. Hosts connected by a pair of TL-PA211KIT adaptors

We estimated the buffer size of the TPLink adaptors by using `nttcp` [8] to generate eight TCP transfers of 1GB at three different MTU values – 600, 1200 and 1500 bytes. Traffic at the source and destination was captured by `tcpdump` [9]. FreeBSD’s SIFTR [4] logged `cwnd`, RTT and number of bytes in flight as recorded in the TCP control block on a per-packet basis. The TCP control block estimate of RTT is a smoothed value, we also used SPP [4] on the captured `tcpdump` files to determine accurate RTT samples for each packet pair. We used `tcpstat` [10] to determine the average throughput of the HomePlug link at different stages of the flow.

We also ran some TCP performance experiments by using `nttcp` to generate 1GB of TCP traffic for each TCP CC variant with `rwnd` set to {50KB, 100KB,

200KB, 600KB, 1MB, 5MB, 10MB¹}. Each of these 21 combinations was repeated 5 times.

IV. BUFFER ESTIMATION

We use two independent techniques to estimate the buffer size of the TPLink adaptor. The first of these is based on measurement of RTT variations and throughput. The second technique takes advantage of our use of SIFTR and the availability of all `cwnd` values captured at the TCP sender. For both approaches, we used data collected from the same experiment outlined above.

A. Method 1

This method of buffer estimation is inspired by the Q-Find method [11], where the buffer size of the link is estimated as the product of throughput and the additional delay due to buffering. The proposed Q-Find method is designed for passive monitoring to a generic remote destination. Its accuracy is limited by the fact that the true RTT cannot be measured, instead it is sampled using periodic ping packets. We do not utilise the ping packets as we are able to capture packets at both the sender and receiver, and subsequently measure true RTTs using SPP. SPP provides an RTT estimate for each packet pair, resulting in many RTT samples for the duration of a TCP flow.

We calculate the buffering delay as the difference between the maximum and minimum observed RTT. These values are determined over the course of a long-lived TCP flow that transfers 1 GB of data. The receiver window is set to 10 MB to ensure that throughput is limited by network congestion and not `rwnd`.

RTT_{min} is defined as the path idle RTT. We estimate this by measuring the RTT during the three way handshake. At this time, the link is unloaded, and we expect the buffers to be empty. As no data is conveyed in the SYN/SYN-ACK packets, the serialisation delay is also minimal.

We expect the maximum RTT to be seen when the buffer is full, immediately prior to the collapse of `cwnd`². This occurs multiple times over the course of a trial, leading to many candidates for RTT_{max} . Selection of RTT_{max} is further complicated due to the time varying nature of actual throughput as conditions on the power-line are not stable. To compensate, we determine a value for RTT_{max} and throughput for each `cwnd` cycle. We eliminate outliers in the RTT measurement by selecting

¹Due to an `nttcp` limitation, `rwnd` was set to 9999KB

²as identified from SIFTR log files

the median of the ten RTT measurements immediately preceding a $cwnd$ collapse.

We also estimate the average throughput of the powerline network for the corresponding $cwnd$ cycle. $tcpstat$ is used to sample the $tcpdump$ file captured at the sender in one second intervals. The mean Ethernet layer throughput of all samples corresponding to a $cwnd$ cycle is used as the measurement of throughput.

The buffer size estimation is given by:

$$BufferSize(bytes)_i = (RTT_{max,i} - RTT_{min}) * EthernetRate_i \quad (1)$$

and

$$BufferSize(packets)_i = \frac{BufferSize(bytes)_i}{MTU} \quad (2)$$

As this estimation is made for each cycle, a number of estimates for buffer size are made. Outlier estimates are removed by taking the median buffer size estimate for the trial. Our results indicate that the buffer size in bytes for the TPLink devices appear proportional to the MTU size, indicating that the buffers are packet based.

Figure 2 plots the CDF of buffer sizes estimated across all trials and all MTU values. These results indicate that the TPLink adaptor buffers can queue 540-550 packets.

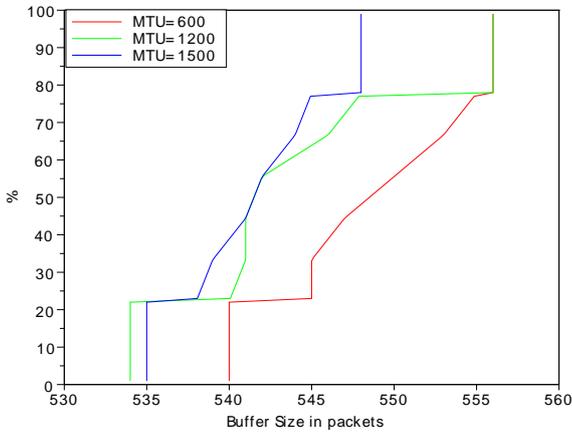


Figure 2. CDF of Buffer Size estimated by Method 1

B. Method 2

The $cwnd$ value in the TCP stack at the sender is a measure of the number of unacknowledged bytes in transit. As the idle path delay-bandwidth product is low, and the HomePlug link is the only bottleneck in

our network, it is fair to assume that the $cwnd$ value is a measure of the number of bytes queued in the TPLink adaptor. As $cwnd$ collapse occurs when buffers overflow and packets are lost, we take the $cwnd$ value immediately prior to collapse as a reasonable estimate for buffer size.

By deploying SIFTR at the TCP sender, we obtained a log of all $cwnd$ values for the duration of all trials. We sampled $cwnd$ immediately prior to collapse for all cycles, then took the median value to eliminate any outliers. The CDF of estimated buffer sizes across all trials and all MTU values are plotted in Figure 3. From our results, the buffer size estimate is about 546 packets.

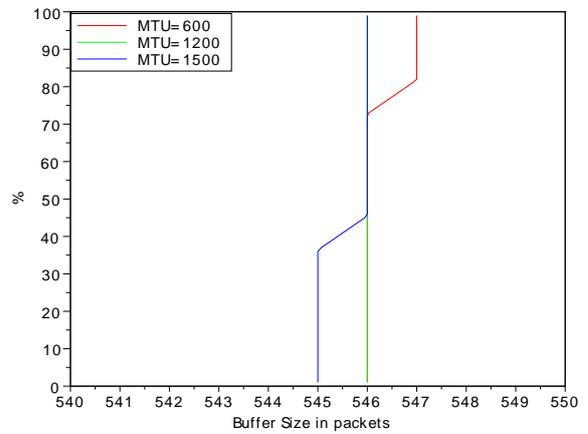


Figure 3. CDF of Buffer Size estimated by Method 2

The range of estimated buffer sizes is much smaller using this method, this can be attributed to the exact measure of $cwnd$. In contrast, the estimates of average throughput and difference in RTT for each $cwnd$ cycle used in Method 1 are not exact, this leads to wider margins of error in the buffer size estimate.

The median buffer size estimates for both approaches are presented in Table I. Both methods have independently produced similar estimates for the buffer size in the TPLink adaptor.

Table I
MEDIAN BUFFER SIZE ESTIMATES FOR TPLINK POWERLINE ADAPTER

MTU in Bytes	Buffer Size (packets)	
	Method 1	Method 2
600	549	546
1200	542	546
1500	542	546

V. DISCUSSION

In our previous work we explored the bufferbloat introduced into the network by the Netgear XAVB2001 HomePlug adaptor. We found that these adaptors internally buffer about 2,720 Ethernet frames. In comparison, buffering on the TPLink TL-PA211Kit HomePlug adaptors is about 546 Ethernet frames. Unlike the Netgear adaptors, the TPLink adaptors do not implement Ethernet Flow Control, a feature which resulted in the Netgear buffers being concatenated to the NIC buffers of the source host, further increasing the effective buffer bloat in the network.

Table II summarises the comparison between the Netgear and TPLink adaptors. While the buffers in the Netgear adaptor are about 5 times larger than those in the TPLink adaptor, the effect of concatenation with the NIC buffers has resulted in a larger impact on the induced RTT in the network for all loss based CC algorithms. In contrast, the CDG CC algorithm ensures that the buffer size has no impact on the induced RTT.

Table II
COMPARISON BETWEEN NETGEAR AND TPLINK HOMEPLUG ADAPTORS

CC Algorithm	Netgear		TPLink	
	RTT _{median}	RTT _{max}	RTT _{median}	RTT _{max}
NewReno	~2,300ms	~2,500ms	~150ms	~240ms
CUBIC	~2,400ms	~2,500ms	~170ms	~250ms
CDG	10ms	20ms	10ms	20ms

Despite the smaller buffer size, we would still argue that the TPLink adaptors exhibit a significant amount of buffer bloat. The median induced RTT of 150-170ms (peak RTT 240-250ms) will be experienced by any real-time traffic sharing the HomePlug link with the TCP flow.

More information about the performance of the NewReno, CUBIC and CDG CC algorithms over the TPLink HomePlug links can be found in Appendix A.

VI. CONCLUSIONS

In this report, we examine the internal buffering of the TPLink TL-PA211KIT HomePlug powerline adaptors. Our previous experiments with the Netgear XAVB2001 adaptors [1] displayed poor network performance under certain scenarios, we became interested in examining whether other HomePlug adaptors exhibited similar behaviour.

We found that like the Netgear adaptors, the TPLink devices form a bottleneck for traffic flowing from an Ethernet LAN into the HomePlug powerline link. We experimentally determined that the TPLink devices buffer approximately 546 Ethernet frames, a figure about five times smaller than the Netgear buffers of about 2,720 Ethernet frames. Ethernet Flow Control deployed by the Netgear device added internal NIC buffering to any buffering performed by the adaptor.

Despite these smaller buffers, the TPLink devices still induce median RTTs of ~150ms (NewReno) or ~170ms (CUBIC) in a loaded network, a delay which is experienced by all flows sharing the link. While this is an order of magnitude smaller than the multi-second delay induced by the Netgear devices, this is still a significant delay to any real-time traffic sharing the link with a TCP flow.

APPENDIX A: PERFORMANCE OF CC ALGORITHMS OVER TPLINK

To get a better idea of how various congestion control algorithms performed over the TPLink HomePlug link, we ran various experiments with the NewReno [5], CUBIC [6] and CDG [7] Congestion Control (CC) algorithms deployed on the sender host. All CC algorithms behaved as expected over a simple bottleneck link. In this Appendix, we quantify the impact of the adaptor buffers in our testbed.

Figure 4 plots the impact of `rwnd` on median throughput for all CC algorithms. All values of `rwnd` achieve approximately the same throughput, CDG was able to achieve nearly 90% of the throughput of the loss based CC algorithms. These results are in line with the performance of all CC algorithms with the Netgear HomePlug devices [1].

Figure 5 plots the median RTT for the three CC algorithms for varying configured `rwnd` values. For `rwnd` values lower than 600KB, the induced RTT for the two loss-based CC algorithms are equal and grows linearly. This behaviour is expected as the TPLink buffer is not being completely filled. In either case, this value is significantly higher than that observed at lower `rwnd` values with no discernable improvement in throughput. We note that the more aggressive nature of CUBIC has resulted in a slightly higher induced RTT (approximately 170ms vs 150ms) due to its tendency to keep the buffers full more often.

In contrast, since CDG does not cause the buffer to be fully utilised, we note that the induced RTT remains roughly constant at 10ms.

REFERENCES

- [1] R. Veeravalli, G. Armitage, J. But, and T. Nguyen, "Interactions between TCP and Ethernet flow control over Netgear XAVB2001 HomePlug AV links," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 130121A, 21 January 2013. [Online]. Available: <http://caia.swin.edu.au/reports/130121A/CAIA-TR-130121A.pdf>
- [2] M. E. Hazen, "The Technology Behind HomePlug AV Powerline Communications," *IEEE Computer*, vol. 41, no. 6, pp. 90–92, June 2008. [Online]. Available: <http://dx.doi.org/10.1109/MC.2008.2053>
- [3] "HomePlug AV White Paper," White Paper, HomePlug Powerline Alliance, August 2005. [Online]. Available: http://www.homeplug.org/tech/whitepapers/HPAV-White-Paper_050818.pdf
- [4] L. Stewart, "Newtcp-tools," accessed 18th Jan, 2013. [Online]. Available: <http://caia.swin.edu.au/urp/newtcp/tools.html>
- [5] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," RFC 5681 (Draft Standard), September 2009. [Online]. Available: <http://tools.ietf.org/html/rfc5681>
- [6] I. Rhee, L. Xu, and S. Ha, "CUBIC for Fast Long-Distance Networks draft-rhee-tcpm-cubic-02.txt," IETF Draft, August 2008. [Online]. Available: <http://tools.ietf.org/html/draft-rhee-tcpm-cubic-02>
- [7] D. A. Hayes and G. Armitage, "Revisiting TCP Congestion Control using Delay Gradients," in *IFIP Networking*, Valencia, Spain, 9-13 May 2011. [Online]. Available: <http://caia.swin.edu.au/cv/dahayes/content/networking2011-cdg-preprint.pdf>
- [8] "FreshPorts – benchmarks/nttcp," accessed 18th Jan, 2013. [Online]. Available: <http://www.freshports.org/benchmarks/nttcp>
- [9] "TCPDUMP/LIBPCAP public repository," accessed 18th Jan, 2013. [Online]. Available: <http://www.tcpcap.org/>
- [10] P. Herman, "Tcpcat," 16 June 2009, accessed 15th Mar, 2013. [Online]. Available: <http://www.frenchfries.net/paul/tcpstat/>
- [11] M. Claypool, R. Kinicki, M. Li, J. Nichols, and H. Wu, "Inferring queue sizes in access networks by active measurement," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science, C. Barakat and I. Pratt, Eds. Springer Berlin Heidelberg, 2004, vol. 3015, pp. 227–236. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24668-8_23

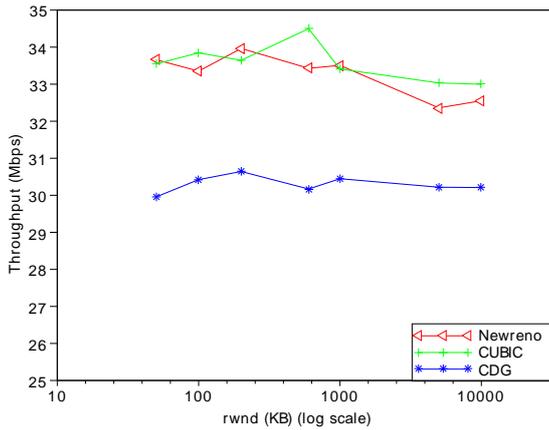


Figure 4. Median throughput versus receiver window size (adjusted y-axis scale)

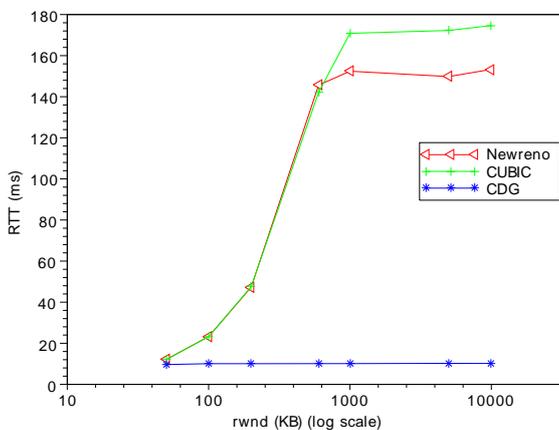


Figure 5. Median RTT versus receiver window size