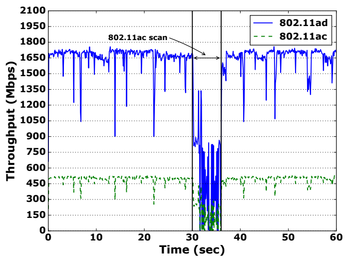
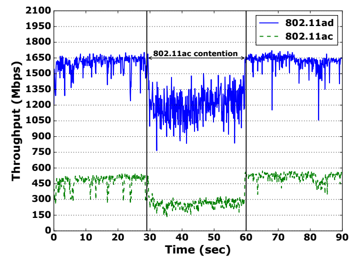
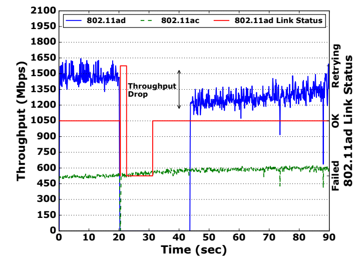


(a) Network scan: Throughput timeline.

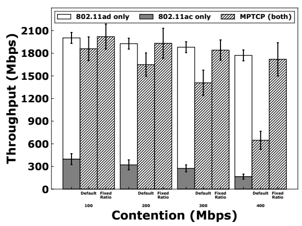


(b) 802.11ac contention: Timeline showing throughput drop during contention.

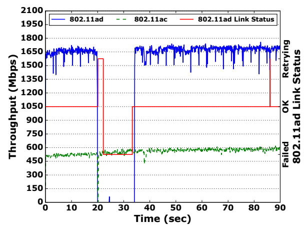


(c) 802.11ad blockage: 802.11ad throughput drop after re-connection.

Figure 2: Performance issues.



(a) *minRTT* vs. *FixedRatio.*



(b) 802.11ad Blockage: Improved recovery time.

Figure 3: *AMuSe* evaluation.

4 *AMUSE*: DESIGN & IMPLEMENTATION

***AMuSe*-SCAN** arbitrates the network scan requests generated from the user space and disables the scheduling of packets to the subflow where the request has been made for the duration of the network scan. However, disabling future scheduling alone may not be enough to prevent packets from being held-up in the TCP queues or at any of the buffers in the lower layers of the network stack. We thus adopt a two-step approach, where *AMuSe*: (1) stops the assignment of packets to subflow about to undertake scanning and (2) waits for the subflow-level *send-queue* to be emptied out. We observed that with *AMuSe*'s network scan management solution applied the 802.11ad throughput remains unaffected during the scan interval. We repeated the measurements several times and observed **2.2x** performance gain on average.

***AMuSe*-CONTENTION** leverages our findings regarding the existence of a unique MPTCP throughput-optimal ratio, for given subflow throughputs. The reaction to contention is to set the packet-assignment ratio to match the ratio of the throughput of 802.11ad and 802.11ac flows, accounting for the drop in 802.11ac throughput due to contention. We test our solution under different amounts of contention (from 100 Mbps to 400 Mbps). Fig. 3(a) shows the expected sum, after accounting for 802.11ac throughput reduction in the presence of contention, and MPTCP performance under the default and *FixedRatio* scheduler, which uses the optimal ratio for a given amount of contention. In all cases, *FixedRatio*'s throughput is close to the expected sum while the default scheduler's throughput can be much lower.

***AMuSe*-BLOCKAGE** reduces the delay in resuming traffic over the 802.11ad subflow by resetting the pf flag to allow for traffic to be scheduled on the 802.11ad subflow. However, we found that this alone was not enough to resume the traffic flow on the 802.11ad interface. When the 802.11ad link is blocked, the subflow-level *cwnd* is cut to 1, with packets in flight also equal to one. As a result, the scheduler is unable to schedule any new packets on 802.11ad subflow since the *cwnd* is reported as being full. In order to overcome this, *AMuSe* uses the TCP's window recovery mechanism to restore the *cwnd* to the value just before loss the event. In contrast to Fig. 2(c), where MPTCP resumed traffic on the 802.11ad subflow after a 12 s delay, with *AMuSe* engaged (Fig. 3(b)) MPTCP starts using the 802.11ad interface in less than 1s after link re-establishment. This is a vast reduction in delay for resuming traffic on the subflow. In a dynamic environment, where such blockage events will occur quite frequently, *AMuSe*'s gains would translate into a significant improvement in user-experience.

5 ACKNOWLEDGEMENTS

This work has been supported in part by NSF grant CNS-1553447, the ERC project SEARCHLIGHT grant no. 617721, the Ramon y Cajal grant RYC-2012-10788, and the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919).

REFERENCES

[1] Kien Nguyen, Mirza Golam Kibria, Kentaro Ishizu, and Fumihide Kojima. 2017. Feasibility Study of Providing Backward Compatibility with MPTCP to WiGig/IEEE 802.11ad. In *Proc. of IEEE Vehicular Technology Conference Fall (VTC-Fall).*

[2] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. 2012. How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP. In *Proc. of 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI).*

[3] Swetank Kumar Saha, Abhishek Kannan, Geunhyung Lee, Nishant Ravichandran, Parag Kamalakar Medhe, Naved Merchant, and Dimitrios Koutsonikolas. 2017. Multipath TCP in Smartphones: Impact on Performance, Energy, and CPU Utilization. In *Proc. of ACM MobiWac.*

[4] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2017. WiFi-Assisted 60 GHz Wireless Networks. In *Proc. of ACM MobiCom.*