series of signals to the sender in order to shrink the size of the sender's TCP congestion window. It then monitors the IoT device's future traffic so that when an IoT device follows TCP congestion control, FR-WARD relabels the suspicious connection as benign. This allows FR-WARD to avoid throttling a benign connection, and thus, FR-WARD avoids resource penalties on benign traffic. On the other hand, if the IoT device fails to follow TCP congestion control mechanisms and still transmits at the same (high) rate, FR-WARD relabels the connection as malicious and begins throttling the traffic. In this case, FR-WARD allows trace amounts of DDoS traffic to be sent to the destination, but only for a very short duration. In fact, FR-WARD can relabel a suspicious connection as malicious and begin throttling within the length of a couple round trip times; in most cases, this additional delay in throttling will be negligible.

However, FR-WARD must also prevent smarter DDoS attacks than ones that blindly send TCP DDoS traffic. If an attacker knows a source network utilizes FR-WARD, the attacker may design a DDoS attack correspondingly. For example, a compromised device could be programmed to follow TCP congestion control. When this occurs, if FR-WARD only sends signals to initially stop the DDoS attack, the compromised device could quickly return to a high sending rate. In order to prevent this, FR-WARD sends multiple rounds of signals to the sender in order to continuously restrict its TCP congestion window to a reasonable size. This bounds the duration of DDoS traffic from the policed network to a negligible amount without penalizing the traffic from benign IoT devices. Denoting s as the number of signals FR-WARD will send each round trip time, or RTT, we can calculate s as $s = \frac{\log_2(\frac{W}{A}+1)}{\lfloor \frac{D}{RTT} \rfloor}$, where D is the negligible amount of time for a receiver to be under DDoS (an external host may notify FR-WARD of a specific allowable DDoS time, or FR-WARD can generically bound successful DDoS time by a reasonable amount), W is the current window size of the sender, and A is the amount of allowed segments each RTT. By this definition of s, after FR-WARD detects a malicious agflow, when FR-WARD sends s signals to the sender each RTT, and the sender follows TCP congestion control, FR-WARD guarantees the DDoS attack has ended after D seconds. Since s may not be a whole number, FR-WARD sends $\lceil s \rceil$ signals for the first m RTTs and $\lfloor s \rfloor$ signals the next $\lfloor \frac{D}{RTT} \rfloor - m$ RTTs where m is the smallest integer such that $\lceil s \rceil * m + \lfloor s \rfloor * (\lfloor \frac{D}{RTT} \rfloor - m) \geq s * \lfloor \frac{D}{RTT} \rfloor$.

Furthermore, since FR-WARD shapes the traffic of all suspicious connections similarly, it must maintain minimal impact on benign devices but maximal impact on malicious devices. Whether a suspicious connection attempts to send benign traffic or DDoS, the receiver cannot handle transmission rates higher than A for a longer period than D. By definition, after FR-WARD shapes the suspicious connection, benign traffic sends segments at the quickest allowable rate, but malicious traffic sends segments at a manageable rate.

Figure 1 examines how many more retransmissions D-WARD requires than FR-WARD for a benign IoT device. In our simulation, a benign IoT device attempts to send a 15-MB file outside of the network. We observed the number of retransmissions each DDoS defense system required of the IoT device over two main parameters: the sender's congestion window size, W, at the time D-WARD or FR-WARD detects a malicious agflow, and the pre-set fraction, f, of traffic that D-WARD or FR-WARD allows to leave the

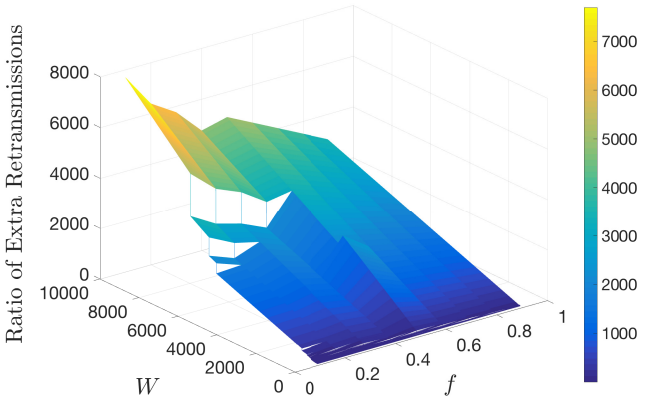

Figure 1: The ratio of extra retransmissions caused by D-WARD over FR-WARD during the transfer of a 15-MB file.

source network during a suspected DDoS attack. Upon detection of a malicious agflow, D-WARD only allows $W * f$ segments to the sender each RTT to mitigate any DDoS attacks. Therefore, when the IoT device follows TCP congestion control, D-WARD drops $W - W * f$ segments every $2RTT$. Furthermore, as W increases or f decreases, D-WARD drops more segments and thus, causes more retransmissions. For FR-WARD, as f decreases, FR-WARD must send more signals to correctly shape the traffic which leads to more retransmissions from the IoT device. This causes the "mountain peak" in Figure 1 when $f \approx \frac{1}{2}$. However, when W surpasses approximately 6500, the amount of traffic D-WARD drops causes overwhelmingly more retransmissions than the number of signals FR-WARD sends even when $f < \frac{1}{2}$.

3 CONCLUSION

FR-WARD is a source-end DDoS defense system that limits DDoS traffic from leaving a policed IoT network, but it also guarantees low negative effect on benign IoT devices. By utilizing the fast retransmit mechanism of TCP, FR-WARD ensures that any DDoS TCP flooding attack that leaves its policed network never surpasses a negligible length of time, but additionally, FR-WARD minimizes resource penalties for benign devices because a benign IoT device always sends traffic at the quickest rate manageable by the receiver and rarely must retransmit dropped packets.

REFERENCES

[1] Samuel Abdelsayed, David Glimsholt, Christopher Leckie, Simon Ryan, and Samer Shami. 2003. An Efficient Filter for Denial-of-Service Bandwidth Attacks. In *Global Telecommunications Conference*, Vol. 3. 1353–1357.

[2] Thomer M. Gil and Massimiliano Poletto. 2001. MULTOPS: A Data-structure for Bandwidth Attack Detection. In *Proceedings of the 10th Conference on USENIX Security Symposium*, Vol. 10. Article 3.

[3] Scott Hilton. 2016. Dyn Analysis Summary Of Friday October 21 Attack. https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack.

[4] Van Jacobson. 1988. Congestion avoidance and control. In *ACM SIGCOMM Computer Communication Review*, Vol. 18. 314–329.

[5] Jelena Mirkovic and Peter Reiher. 2005. D-WARD: A Source-end Defense Against Flooding Denial-of-Service Attacks. In *IEEE transactions on Dependable and Secure Computing*, Vol. 2. 216–232.