Role of Network Control Packets in Smartphone Energy Drainage

Chander Govindarajan*, Satadal Sengupta†, Pradipta De‡, Bivas Mitra§, Sandip Chakraborty¶
*†§¶Department of Computer Science and Engineering, IIT Kharagpur, India 721302
‡Department of Computer Science, State University of New York (SUNY),Korea 406-840
Email: *chanderg@cse.iitkgp.ernet.in,†satadal.sengupta.nit@gmail.com, ‡pradipta.de@sunykorea.ac.kr,
§bivas@cse.iitkgp.ernet.in, ¶sandipc@cse.iitkgp.ernet.in

***Abstract*—Energy drainage in smartphones via communication interfaces has been an important area of research in the past few years. While a large portion of the literature focuses on efficient scheduling of data packets to reduce energy wastage, there has been no attempt to study the impact of control packets in smartphone energy drainage. In this poster, we focus on understanding the role of control packets in excess energy consumption. We identify the types of control packets which contribute to this wastage, and point out areas in the design of traditional protocols which may need rethinking keeping in mind the nature of mobile data access.**

I. INTRODUCTION

Energy efficiency for smart-phones has become an important issue in the last few years because smart-phones find diverse usage in day-to-day computation and communication. A significant amount of energy drainage in smart-phones is due to data transfers via communication interfaces; therefore, a large number of recent studies, such as [1] and references therein, have been devoted for developing energy models for third generation (3G) and fourth generation (4G) cellular networks. A popular way of saving energy in smart-phones is via efficient scheduling of data packets that exploit the energy model for 3G/4G networks. According to these energy models, a smart-phone communication interface has 3 power states - IDLE, CELL_FACH (or FACH) and CELL_DCH (or DCH). IDLE is the state when no power is consumed, whereas DCH is the high throughput state with high power consumption. FACH is an intermediate power state between the two. Whenever an interface is in DCH or FACH state, and it does not receive data packets for a time threshold, called the *tail time*, the state is transited to IDLE and the interface is switched off to save power. The tail time is in the order of few seconds, and the exact value varies among network service providers.

Many existing works, such as [2], [3] and the references therein, have developed adaptive data packet scheduling and tail time optimization protocols, such that state transitions can be minimized while an interface can spend more time in the IDLE state by delaying the data packet transmission up to a bound. However, such existing methods only consider the energy state transitions due to data packets, and remain silent about control packets. The motivation behind this work comes from two points. First, data packets come in bursts, while control packets are mostly unregulated and generated by the network management protocols, resulting in a possibility of energy state ramp-up only to serve a control packet. Second, although data packets can tolerate bounded delay based on application requirements, control packets may not be externally scheduled as it will hamper general network management activities. As a consequence, in this poster, we explore the impact of network control packets on smart-phone energy consumption, by observing how many state transitions are affected due to the network control packets. During our experiments, it was observed that indeed a large amount of energy is drained in excess due to network control packets. In this poster, we quantify the state transitions (from a higher state to IDLE state) missed, due to the arrival of a 'rogue' control packet. We also explore the various types of control packets which contribute to such transition-misses, and find that TCP control packets form a majority of the rogue packets. We quantify the energy wastage associated with rogue control packets, and also report some observations on how the existing TCP retransmission policy may not be suitable for use in cellular networks.

II. OBSERVATIONS AND ANALYSIS

For this study, we have conducted a one month long data collection activity, through 2 volunteers and with Motorola Moto X 2^{nd} generation Android mobile phones. The devices have been rooted and the `tcpdump` binary has been installed in each one of them. The Android `Terminal Emulator` app is used to trigger the trace recording, and is left to run in the background while the volunteer uses the smartphone like a regular user. The phones have throughout been connected to the cellular network (mostly 3G), and the volunteers have used a large variety of apps on their devices like a regular smartphone user would. The total trace is around 2 GB in size, and contains user activity logged during different times of the day, under various conditions of user mobility (such as, walking, driving, etc.) and user practices.

A. Transition Miss and Excess Energy Drainage Due to Network Control Packets

We fit the traffic traces in the cellular energy state transition model to figure out the number of FACH → IDLE or DCH → IDLE state transition misses due to control packets. Let t be the time when the last data packet was transmitted or received, and τ be the tail time. If a control packet arrives (or

978-1-4673-9622-6/16/$31.00 ©2016 IEEE

TABLE I

CITM and Corresponding Excess Energy Drainage (CITM is given in the form TP/T, where TP denotes TCP control packets and T stands for total number of control packets which result in a transition miss)

Dataset	**Size (MB)**	**Configuration I**			**Configuration II**		
		CITM-DCH	**CITM-FACH**	**EED (J/KB)**	**CITM-DCH**	**CITM-FACH**	**EED (J/KB)**
1	184.5	51516/54480	1088/1152	0.28968	51624/54615	984/1030	0.28529
2	84.2	37112/37762	593/616	0.20087	37200/37853	506/527	0.20220
3	73.2	30255/30731	120/123	0.25995	30275/30754	100/100	0.26088
4	16.7	8015/8193	199/214	0.32258	8070/8252	145/157	0.34327
5	13.0	6923/7065	447/450	0.33959	6946/7091	424/426	0.32802
6	10.7	6225/6386	220/247	0.36524	6257/6437	191/207	0.36831

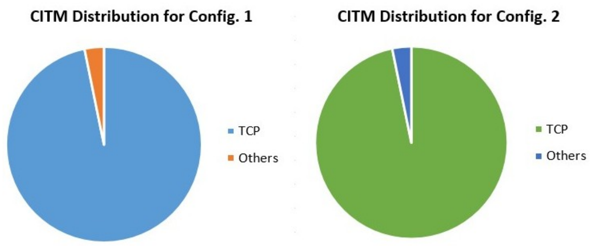

Fig. 1. Contribution of TCP towards CITM

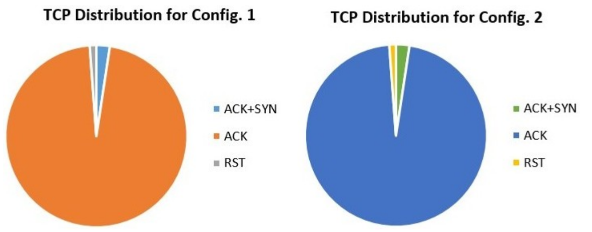

Fig. 2. TCP packet-types responsible for EED

is transmitted) within the time $[t, t+\tau]$, then it results in either a FACH → IDLE or DCH → IDLE state transition miss. We call such a transition miss as *control initiated transition miss* (CITM). Such a CITM indicates that the network interface needs to be in high power state for additional time which results in *excess energy drainage* (EED), which is estimated using the energy model suggested in [4] and references therein.

Table I summarizes our observations from different data sets. We use two different configurations of tail time values for DCH and FACH. In config. I, we set DCH tail time as $3s$, and FACH tail time as $10s$. For config. II, these values are set to $5s$ and $12s$ respectively[1]. We observe that a large number of CITMs are due to control packets which result in EED in the range of $0.2 - 0.4$ Joules/KB, which is significantly high. We further notice that transition misses are considerably higher in number for DCH as compared to FACH.

We get another important observation from Table I. A majority of the control packets that result in energy state transition miss are the TCP control packets. Next we analyze the impact of TCP control packets over excess energy drainage.

B. *Transition Miss due to TCP Control Packets*

Fig. 1 shows how TCP control packets form the majority of rogue control packets. This observation leads us to explore the various types of TCP control packets and their role in CITM. Fig. 2 shows a distribution of different TCP control packets that result in transition misses and excess energy drainage. The figure indicates the TCP acknowledgement (ACK) and reset (RST) packets cause majority of the transition misses. From the packet trace analysis, we observe that these transition misses are mostly in the regions when TCP retransmission occurs due to a timeout or congestion control. It is well studied in the existing literature that TCP triggers many retransmission timeouts and spurious congestion control in mobile wireless environment, due to an intermittent connection breakdown or sudden fluctuations in physical data rates. During TCP retransmissions, TCP sends duplicate ACKs and RST control packets for some duration after the last data packet has already been transmitted. Consequently, such control packets are out-of-sync from the normal data transfers, and therefore causes the system to remain in high energy state for a longer duration.

III. Conclusion

In conclusion, control packets contribute significantly to excess energy drainage in smartphones. While it is possible to delay data packets with the purpose of aggregation and thus save energy, delaying control packets may result in complete breakdown of the network. We observe that out of the various types of rogue control packets, TCP contributes the most to excess energy drainage in day-to-day usage. We point out how the existing TCP retransmission policy may not be suitable for cellular networks. In future, we aim to design and implement packet-scheduling algorithms to save energy, with due consideration to control packets, unlike existing works.

References

[1] A. Nika, Y. Zhu, N. Ding, A. Jindal, Y. C. Hu, X. Zhou, B. Y. Zhao, and H. Zheng, "Energy and performance of smartphone radio bundling in outdoor environments," in *Proc. of the 24th WWW*, 2015, pp. 809–819.

[2] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "Top: Tail optimization protocol for cellular radio resource allocation," in *Proceedings of the 18th IEEE ICNP*, 2010, pp. 285–294.

[3] B. Zhao, W. Hu, Q. Zheng, and G. Cao, "Energy-aware web browsing on smartphones," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 761–774, 2015.

[4] S. Pradhan, S. K. Dandapat, N. Ganguly, B. Mitra, and P. De, "Aggregating inter-app traffic to optimize cellular radio energy consumption on smartphones," *COMSNETS*, 2015.

---

[1] The choice of tail time values is representative of real-world cases [2]. Even though other static values or even dynamic values are possible, we restrict ourselves to 2 configurations due to space constraints.