International Journal of Innovative Research and Advanced Studies (IJIRAS)
Volume 3 Issue 8, July 2016

ISSN: 2394-4404

Leveraging Social Networks For P2P Content-Based File Sharing In Disconnected MANETs

T. Ramesh
Assistant Professor,
Dept. of Information Technology,
Bharathiar University, Coimbatore

V. Umamaheswari
M.Phil Research Scholar,
Dept. of Information Technology,
Bharathiar University, Coimbatore

Abstract: ***3G technology has stimulated a wide variety of high band width applications on smart phones, such as video streaming and content-rich web browsing. Although having those applications mobile is quite appealing, high data rate transmission also poses huge demand for power. It has been revealed that the tail effect in 3G radio operation results in significant energy drain on smart phones. Recent fast dormancy technique can be utilized to remove tails but, without care, can degrade user experience. Implement Tail Theft in the Network Simulator with a model for calculating energy consumption that is based on parameters measured from mobile phones. When allowing for delays of a few seconds (acceptable for background applications), the energy savings increase to between 62% and 75% for 3G, and 71% for LTE. The increased delays reduce the number of state switches to be the same as in current networks with existing inactivity timers.***

Keywords: MANETs, content-based file sharing, social networks

I. INTRODUCTION

Over a fifth of the 5.5 billion active mobile phones today have “broadband” data service, and this fraction is rapidly growing. Smart phones and tablets with wide-area cellular connectivity have become a significant, and in many cases, dominant, mode of network access. Improvements in the quality of such network connectivity suggest that mobile Internet access will soon overtake desktop access, especially with the continued proliferation of 3G networks and the emergence of LTE and 4G. Wide-area cellular wireless protocols need to balance a number of conflicting goals: high throughput, low latency, low signalling overhead (signalling is caused by mobility and changes in the mobile device’s state), and low battery drain.

The 3GPP and 3GPP2 standards (used in 3G and LTE) provide some mechanisms for the cellular network operator and the mobile device to optimize these metrics but to date; deployed methods to minimize energy consumption have left a lot to be desired. The 3G/LTE radio consumes significant amounts of energy; on the iPhone 4, for example, the stated talk time is “up to 7 hours on 3G” (i.e., when the 3G radio is on and in “typical” use) and “up to 14 hours on 2G”.1 On the Samsung Nexus S, the equivalent numbers are “up to 6 hours 40 minutes on 3G” and “up to 14 hours on 2G”.2 That the 3G/LTE interface is a battery hog is well-known to most users anecdotally and from experience, and much advice on the web and on blogs is available on how to extend the battery life of your mobile device.3 Unfortunately, essentially all such advice says to “disable your 3G data radio” and “change your fetch data settings to reduce network usage”. Such advice largely defeats the purpose of having an “always on” broadband-speed wireless device, but appears to be the best one can do in current deployments.

Show the measured values of 3G energy consumption for multiple Android applications. This bar graph shows the percentage of energy consumed by different 3G radio states. For most of these applications (which are all background applications that can generate traffic without user input, except for Facebook), less than 30% of the energy consumed was during the actual transmission or reception of data. Previous research arrived at a similar conclusion about 60% of the energy consumed by the 3G interface is spent when the radio is not transmitting or receiving data. In principle, one might imagine that simply turning the radio off or switching it to a low-power idle state is all it takes to reduce energy consumption. This approach does not work for three reasons. First, switching between the active and the different idle states takes a few seconds because it involves communication with the base station, so it should be done only if there is good reason to believe that making the transition is useful for a reasonable duration of time in the future.

Second, switching states consumes energy, which means that if done without care, overall energy consumption will increase compared to not doing anything at all. Third, the switching incurs signalling overhead on the wireless network, which means that it should be done only if the benefits are substantial relative to the cost on the network. tackles these challenges and develops a solution to reduce 3G/LTE energy consumption without appreciably degrading application performance or introducing a significant amount of signalling overhead on the network. Unlike currently deployed methods that simply switch between radio states after fixed time intervals an approach known to be rather crude and sub-optimal our approach is to observe network traffic activity on the mobile device and switch between the different radio states by adapting to the workload.

A traffic-aware design to control the state transitions of a 3G/LTE radio taking energy consumption, latency, and signalling overhead into consideration. The design incorporates two algorithms:

- ✓ Make Idle, which uses aggregate traffic activity to predict the end of an active session by building a conditional probability distribution of network activity.
- ✓ Make Active, which delays the start of a new session by a few seconds to allow multiple sessions to all become active at the same time and therefore reduce signalling overhead. This method is appropriate for non-interactive background applications that can tolerate some delay.

II. PROBLEM DEFINITION

Current peer-to-peer (P2P) file sharing methods in mobile ad hoc networks (MANETs) can be classified into three groups: flooding-based, advertisement-based, and social contact-based. The first two groups of methods can easily have high overhead and low scalability. They are mainly developed for connected MANETs, in which end-to-end connectivity among nodes is ensured. The third group of methods adapts to the opportunistic nature of disconnected MANETs but fails to consider the social interests of mobile nodes, which can be exploited to improve the file searching efficiency

- ✓ Direct, where a mobile sink visits each sensor node and collects data via a single hop;
- ✓ Rendezvous, where a mobile sink only visits nodes designated as RPs.

The main goal of protocols in category 1 is to minimize data collection delays, whereas those in category 2 aim to find a subset of RPs that minimize energy consumption while adhering to the delay bound provided by an application.

III. RELATED WORK

MITIGATING 3G RADIO TAIL EFFECT ON SMARTPHONES, FEI YUY, GUANGTAO XUEY, HONGZI ZHU, ET ALL,(2010)

Although having those applications mobile is quite appealing, high data rate transmission also poses huge demand for power. It has been revealed that the tail effect in 3G radio operation results in significant energy drain on smartphones. Recent fast dormancy technique can be utilized to remove tails but, without care, can degrades user experience.Propose a novel scheme SmartCut, which effectively mitigates the tail effect of radio usage in 3G networks with little side-effect on user experience. The core idea of SmartCut is to utilize the temporal correlation of packet arrivals to predict upcoming data, based on which unnecessary high-power-state tails of radio are cut out leveraging the Fast Dormancy mechanism.

Extensive trace-driven simulation results demonstrate the efficacy of SmartCut design. On average, SmartCut can save up to 56.57% energy on average while having little side-effect to user experience. SmartCut, which uses historical 3G traffic data to train ARMA models and further utilizes the predicted arrival time of future data transmission to effectively cut unnecessary tails while having little sideeffect to user experience.

ENERGY CONSUMPTION IN WIRELESS NETWORKS PROGRESS, NICLAS JACOBSSON, MICKAEL LOPES, ET ALL, (2009)

Since the mobile phone relies on a limited power source and provides more demanding services today, minimizing the energy consumption is one of most important factors to consider. In this context, the focus will be set for two areas: modulation formats in the physical layer and the RRC-protocol in the network layer. The modulation formats section will be a literature study only, to show the energy advantages by using higher modulation combined with channel coding. Main focus will be set on the RRC-protocol and its functionality. The simulation chapter will show the importance of precise timer settings to gain energy efficiency as well as a comparison between the original RRC-protocol and a RRC-protocol proposed by us. The result shows decrease of 2.4 % for the proposal RRC and therefore seems to be more effective during web browsing utilization. Furthermore, the effect of the proposed RRC has to be confirmed in other utilization aspects as well as its impact on the transmission delay.

TRAFFIC-AWARE TECHNIQUES TO REDUCE 3G/LTE WIRELESS ENERGY CONSUMPTION, SHUO DENG AND HARI BALAKRISHNAN, ET ALL,(2010)

The 3G/LTE wireless interface is a significant contributor to battery drain on mobile devices. A large portion of the energy is consumed by unnecessarily keeping the mobile device’s radio in its “Active” mode even when there is no traffic. This paper describes the design of methods to reduce this portion of energy consumption by learning the traffic patterns and predicting when a burst of traffic will start or end. We develop a technique to determine when to change the radio's state from Active to Idle, and another to change the radio's state from Idle to Active.

In evaluating the methods on real usage data from 9 users over 28 total days on four different carriers, find that the energy savings range between 51% and 66% across the carriers for 3G, and is 67% on the Verizon LTE network. When allowing for delays of a few seconds (acceptable for background applications), the energy savings increase to between 62% and 75% for 3G, and 71% for LTE. The increased delays reduce the number of state switches to be the same as in current networks with existing inactivity timers. The key idea in this paper is to adapt the state of the radio to network traffic. To put the 66% saving (without any delays) or 75% saving (with delay).

IV. PROPOSED METHODOLOGY

The Radio Network Controller (RNC) is an important component of the UMTS network. Most features of the UMTS Terrestrial Radio Access Network such as radio resource management, packet scheduling, and some mobility management functions are implemented in the RNC. Radio resources shared among UEs are potential bottlenecks in the network. To use the limited resource efficiently, the RRC protocol described in maintains a state machine for both the UE and RNC.

Promotion and demotion are the two types of state transitions. The UE consumes more (less) radio resources and power after promotion (demotion). Promotion, which includes IDLE→DCH, FACH→DCH, and IDLE→FACH2, is triggered by the Buffer Occupancy (BO) level (downlink and uplink buffers are separated) in the Radio Link Control (RLC) layer. When the state machine is in the IDLE state and the BO level is greater than 0, IDLE→DCH promotion occurs. When in the FACH state and when the BO level of either direction exceeds the configured threshold, FACH→DCH promotion occurs. In contrast to promotion, demotion comprises DCH→FACH, FACH→IDLE, and DCH→IDLE. Demotion is triggered by network throughput and inactivity timers α and β, which are managed by the RNC. In the DCH state, the α timer is reset whenever considerable traffic is generated (because low traffic volume does not trigger timer being reset). When the throughput is 0 or less than the configured threshold for the RNC configured time ($T1$), the α timer stops, and the state is demoted to the FACH state. Similar to the α timer, in the FACH state, the RNC resets the β timer whenever it observes traffic. However, when the throughput is 0 for the configured time ($T2$), the β timer stops, and the state is demoted to the IDLE state.

A. PRELIMINARY FORMULATION

a. *DIFFERENTIATING APPLICATIONS*

Common network applications in UEs include instant messaging, e-mail, news, social networking (e.g., Facebook, twitter), browsing, media (e.g., videos, radio streaming), maps, RSS feeds and system (e.g., software updates). According to data accessing characteristics, these applications can be classified under three categories: (1) applications that can tolerate delays, (2) applications that can benefit from prefetching, and (3) real-time applications, the data of which are neither delaytolerant nor prefetchable. E-mail, RSS feeds and system are applications that can tolerate a small user- or application specified delay. For example, a user may be willing to wait for a short time before e-mail is sent or received if it can save substantial energy. News, social networking, browsing, media and maps are applications that can benefit from aggressive prefetching.

The data accessing requests of the three aforementioned categories of applications can be formulated as undivided requests, where each request i has two attributes: an arrival time ai and a deadline di. The corresponding three categories of requests are as follows: (1) real-time requests, which require instantaneous scheduling after arrival and satisfy ai = di, (2) delay-tolerant requests, which can be scheduled at a delayed period with di − ai after arrival at ai and satisfy ai < di, (3) prefetchable requests, which can benefit from prefetching and satisfy ai = di. Prefetchable requests can be handled in accordance with real-time requests, or add one or more previous attempts before the corresponding prefetchable request arrives.

Let PRi = {pr1, ...} denote these previous attempts, where the size of PRi is greater than 0. Suppose prj ∈ PRi, the arrival time of prj is aj and satisfy dj = ai (Note that previous attempts would not be attempted). If the data obtained by previous attempt prj is exactly the desired data indicated by the prefetchable request i, this request no longer requires scheduling. If the attempts fail, request i should be scheduled. Delay-tolerant requests and previous attempts can be delayed, but the difference between them is that the latter would be discarded as the deadline approaches because of the risk of prefetching failure.

b. *TRADE-OFFS BETWEEN RESOURCE AND USER LATENCY*

An ideal situation is one wherein requests can always be responded to in real-time. However, limitations in UE battery capacity, radio resources, and the RNC processing capacity make this ideal state difficult to achieve in practice. The request scheduling problem introduces trade-offs between resource and user latency. Given a request queue R and a schedule S, we calculate two categories of metrics to characterize these trade-offs. Some previous studies focus on only one factor. Although recent studies have provided three types of metrics, these Metrics are not the trade-offs that we consider in the current study and are not sufficiently precise. The two categories of metrics that we define are described as follows.

The resource factor can be quantified by four metrics.

- ✓ The energy consumption of the UE, denoted by *E(S)*, is the total energy consumed during the schedule. *E(S)* comprises the energy consumed during state transitions, transmissions, and the tail time. *E(S)* not only reflects the resource consumption of the UE, but also affects radio resource utilization and the RNC overhead.

- ✓ Duration in the DCH state, denoted by *D(S)*, quantifies the dedicated radio resources consumed by the UE on dedicated channels.
- ✓ Duration in the FACH state is denoted by *F(S)*. *F(S)* quantifies the radio resources consumed by the UE on shared channels, which should be considered.
- ✓ The number of promotions of IDLE→DCH and FACH→DCH, denoted by *P(S)*, quantifies the overhead incurred by state transitions that increase the resources used in the RNC. The overhead of state demotions is disregarded, because it is significantly smaller than that of state promotions.

c. *APPLIED ENERGY CONSUMPTION MODEL*

In this section, we introduce an energy model applied by TailTheft for calculating energy consumption E(S). Although some energy models have already been presented, we only consider energy models that are used by existing tail optimization approaches because the energy consumption model is not the primary focus of this paper. The model given by TailEnder. Does not determine the energy consumption affected by transmission rate (which refers to the amount of data transferred per second) in the FACH state and does not measure FACH→DCH promotion energy, which cannot be disregarded. TOP provides the power consumed during state transitions but does not determine the energy consumption affected by transmission rate. To construct an accurate energy model, we conduct a series of measurements on a Nokia N81 device using Energy Profiler to obtain a set of energy consumption data. Based on the data set, we analyze the energy consumption of different states and state transitions. Illustrates the power curve of a measured transmission process, where a transmission process refers to the change in power state from low to high and then back to low. Let Ei denote the energy consumed by a transmission process, then E(S) = _Ei. We divide Ei into four parts, which are shown below.

(1) The energy consumed by IDLE→DCH promotion denoted by E1p is affected by the average power and duration of IDLE→DCH. Let Wi2d and Pt1 denote the average power and duration of IDLE→DCH, respectively, then E1p = Wi2d × Pt1.

(2) The energy consumed in the DCH state is denoted by E1s Power in the DCH state is influenced by both the transmission rate and duration of the DCH state. Let function Dw(vt, t) denote the power in the DCH state, where vt is the transmission rate at time t. Then,

E1s= _t2Dt1 D Dw(vt, t)dt,

where t1 D is the start time, and t2 D is the end time of the DCH state.

(3) The energy consumed by FACH→DCH promotion denoted by E2p is affected by the average power and duration of FACH→DCH. Let Wf 2d and Pt2 denote the average power and duration of FACH→DCH, respectively, then E2p = Wf 2d × Pt2.

(4) The energy consumed in the FACH state is denoted by E2s Power in the FACH state is influenced by both the transmission rate and duration of the FACH state. Let function Fw(vt, t) denote the power in the FACH state, where vt is the transmission rate at time t. Then

E2s= _t2 F t1 F Fw(vt, t)dt,

where t1

F is the start time, and t2

F is the end time of the FACH state.

For a certain UE in a specified UMTS network, E1p and E2p are constants, denoted by C1 and C2, respectively. Suppose Np FACH→DCH promotions occur (Np ≥ 0, not all transmission processes have only one FACH→DCH promotion), then the energy consumed by a transmission process is computed as:

$$E_i = \int_{t_D^1}^{t_D^2} D_w(v_t, t)\mathrm{d}t + \int_{t_F^1}^{t_F^2} F_w(v_t, t)\mathrm{d}t + C_1 + N_p \times C_2. \quad (1)$$

B. TAILTHEFT DESIGN

a. *VIRTUAL TAIL TIME*

Achieving batching and prefetching in the tail time requires identification of the time during which these tasks can be performed. The types of demotion, namely, DCH→FACH→IDLE or DCH→IDLE, can be accurately determined. The former demotion type has two tail times (T1 and T2), and the latter has only one tail time (T1). The mechanism of the two tail times can be directly applied to one tail time. Thus, we consider only the former. i separate the two tail times primarily because the scheduling rates in these two periods are distinct.

Two mechanisms are employed for online determination of whether now is the tail time. Power-based state inference mechanism is used to infer the current RRC state based on power consumption. Determining the current RRC state is the foundation of distinguishing between the two kinds of tail time. However, no API or known work on directly accessing RRC state information in smartphone systems is available. Radio resources are the major power consumer in UEs, thereby making power consumption a convenient factor for inferring the RRC state. Moreover, a power-based state inference mechanism has been proven effective with high accuracy. Given that it can exhibit high accuracy (more than 95%), the error estimate of this mechanism is disregarded. Virtual tail time that is used to Parameters Set in TailTheft Implementation determine whether now is the tail time, which corresponds to the original inactivity timers maintained by the RNC. After transmitting data in the tail time, the inactivity timers are reset, such that the physical tail time is broken. We refer to the used tail time as the virtual tail time. A timer is required to determine whether now is the virtual tail time in the current RRC state. We refer to this timer as the virtual tail timer, which performs operations that are similar to those performed by the inactivity timer maintained by the RNC. Two timers correspond to the virtual tail times of the DCH and FACH states, which are denoted as γ and δ, respectively.

Similar to the inactivity timer α, the virtual tail timer γ is activated when the throughput is 0 or under the configured threshold (Table 2). (1) If timer γ is activated when the throughput is 0, TailTheft can start transmitting data after the timer γ is activated and stop when the timer γ expires or is reset. (2) If timer γ is activated when the throughput is under the configured threshold but greater than 0, TailTheft cannot transmit data after the timer γ is activated. If TailTheft transmits data under this condition, the transmission of real-time data may be ongoing when the timer γ expires, and demotion at this time would trigger additional state promotions. Thus, having no transmission at the second condition would not reset the inactivity timer α, and the state is demoted to the FACH state after the expiry of timer α. When in the FACH state, the virtual tail timer δ would be activated only when the throughput is 0. TailTheft can start transmitting data after timer δ is activated and stop when the timer δ expires or is reset.

b. DUAL QUEUE SCHEDULING ALGORITHM

Scheduling requests feasibly, Applications submit network requests by calling the API Defined as According to the parameter r_delay, requests can be divided into two categories: requests that must be scheduled instantaneously (r_delay is zero, including real-time and unsuccessfully prefetched requests) and requests that can be delayed (r_delay is positive or negative). Requests that can be delayed, referred to as TailTheft requests, include delay-tolerant requests and previous attempts. TailTheft employs a dual queue scheduling algorithm for scheduling these two categories of requests.

TailTheft schedules requests by maintaining two queues:

(1) the real-time queue for requests that must be scheduled instantaneously, and (2) the TailTheft queue for TailTheft requests. TailTheft schedules requests in the real-time queue if requests are present in this queue and schedules those in the TailTheft queue if the real-time queue is empty or if the deadline of the first request in the TailTheft queue approaches.

ALGORITHM FOR DUAL QUEUE SCHEDULING IN TAIL THEFT

```
TailTheft API
_ dl: the latest deadline of request in Qt
procedure SUBMITREQUEST(r_delay)
i ← the submitted request
ai ← current time tnow
if r_delay = 0 then
Qr.enqueue(i)
else
dl ← Qt.enqueue(i)
θ.restart(dl − ai)
end if
end procedure
```

However, if the first request in the TailTheft queue has not been transmitted when the timer θ stops, the request should be directly dequeued. TailTheft processes this request according to the value of r_delay. If r_delay is positive, the request is delay-tolerant and should be scheduled immediately, regardless of the current RRC state. If r_delay is negative, the request is a previous attempt and should be discarded. Similarly, the timer θ should be reactivated according to the deadline of the next request in the queue after dequeuing.

V. PERFORMANCE EVALUATION

A. ENERGY CONSUMPTION

Energy consumption is the consumption of energy or power. The energy model represents the energy level of nodes in the network. The energy model defined in a node has an initial value that is the level of energy the node has at the beginning of the simulation. This energy is termed as initial energy. In simulation, the variable “energy” represents the energy level in a node at any specified time. A node looses a particular amount of energy for every packet transmitted and every packet received. The energy consumption level of a node at any time of the simulation can be determined by finding the difference between the current energy value and initial energy value. If an energy level of a node reaches zero, it cannot receive or transmit anymore packets. The energy level of a network can be determined by summing the entire node’s energy level in the network.

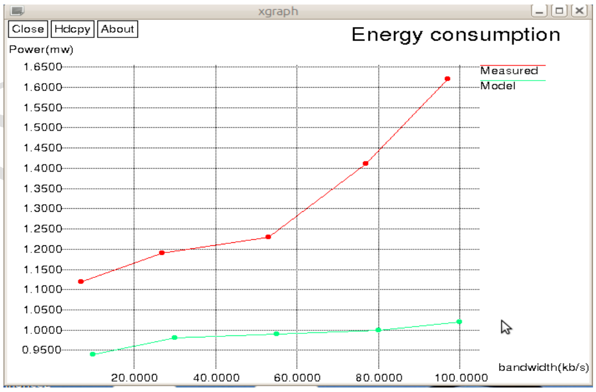

Figure 5.1: Energy consumption

B. FAST ACCESS CHANNEL (FACH)

FACH shows the comparison of Tail theft and Tail ender that is compared on the basis of user. The comparison has been done between the existing Tail time and the proposed TailTheft strategy. It is seen that when the number of nodes increased to 100 the energy seems to deliver the packets in a progressive ratio when comparing to the existing strategy.

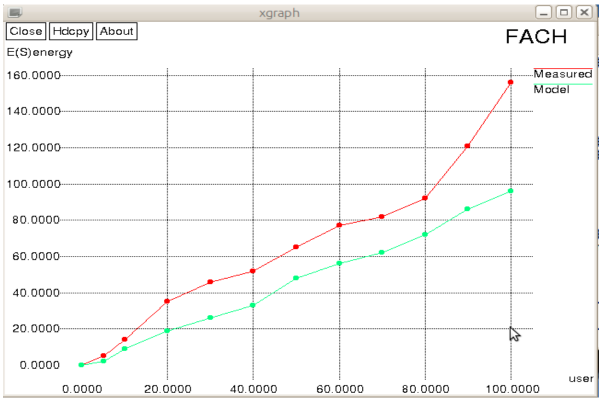

Figure 5.2: Fast Access Channel

C. DEDICATED CHANNEL (DCH)

DCH shows the comparison of tail ender and tail theft that is compared on the basis of Delay line. The comparison has been done between the existing Tail time and the proposed Tail Theft strategy. It is seen that when the number of nodes increased to 100 the energy seems to deliver the packets in a progressive ratio when comparing to the existing strategy.

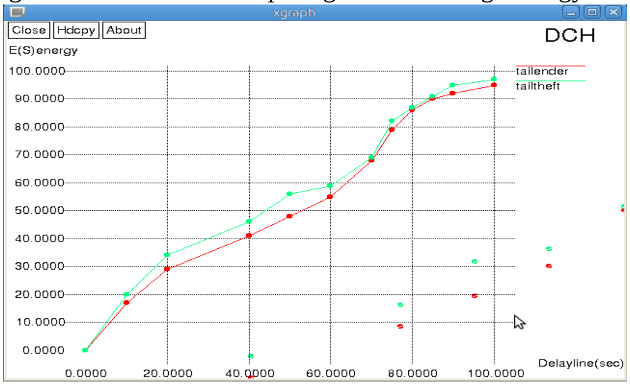

Figure 5.3: Dedicated channel

VI. CONCLUSION

In this paper, we propose a Social Network based P2P file sharing system in disconnected MANETs. Inactivity timers in cellular networks are used to balance the trade-offs between resource efficiency for enhanced user experience and low management overhead. However, considerable radio resources and battery energy are wasted in the tail time. Proposed TailTheft, which leverages the tail time for batching and prefetching. Our work is the first to consider using rather than eliminating the tail time for saving energy.

To utilize the tail time, TailTheft uses a virtual tail time mechanism to determine the amount of tail time that can be used and a dual queue scheduling algorithm to schedule transmissions. Given that numerous transmissions are scheduled in the tail time, energy consumption is significantly decreased. TailTheft can benefit a number of common applications, including delay-tolerant applications (e.g., e-mail, RSS feeds, and software updates), and prefetchable applications (e.g., news, social networking, browsing, media and maps). We have simulated TailTheft in NS-2 and evaluated its performance under various conditions. The experimental results show that TailTheft achieves more significant savings on battery energy and radio resources than Tail Time.

Under different tail configurations, Tail Theft obtains much better performance than Tail Ender. To construct an accurate energy model, we conduct a series of testing. Power in the FACH state is influenced by both the transmission rate and duration of the FACH state.

We evaluate the performance of TailTheft using both per-application and real life traces. It shows that TailTheft can achieve an energy saving of 20% to 34% in different cases.

A. FUTURE WORK

In future, we will explore how to determine appropriate thresholds in P2P, how they affects the file sharing efficiency, and how to adapt P2P to larger and more disconnected networks.

In the forth coming works it can also be enhanced in the perspective of employing an energy aware and energy efficient routing, since energy harvesting is one of the vital fact that is to be considered in MANET Routing for attaining a best network performance.

The propose scheme enhances good relationship between ambassadors and Coordinators relationship with allotted time slot for each individual node. This proposed schemed extended to implement in Decentralized approach for intra and inter community phases, mobility enhancement and also providing simple node prevention scheme. This part of work has been planned for implementation of second phase of this project.

REFERENCES

[1] A. Bergman, A. Shavit, N. Leibowitz and R. Ben-Shaul, “Are file swapping networks cacheable? Characterizing P2P traffic,” in Proc. of International Workshop on Web Content Caching and Distribution (WCW’02), Boulder, CO, August 2002.

[2] A. Carroll and G. Heiser, “An analysis of power consumption in a smartphone,” in Proc. USENIX ATC, Berkeley, CA, USA, Jun. 2010, pp. 1–14.

[3] Bram Cohen, “Incentives Build Robustness in Bit Torrent”, May 22, 2003. Available: http://www.bittorrent.org/bittorrentecon.pdf.

[4] Chuan Wu and Baochun Li, Department of Electrical and Computer Engineering University of Toronto, On “Meeting P2P Streaming Bandwidth Demand with Limited Supplies”.

[5] *Emmanuel Stone, Tim Czerniak, Colm Ryan, Rob McAdoo, On* “Peer to Peer Routing”.

[6] Ian Clarke, Oskar Sandberg, Brandon Wiley, Theodore W. Hong, "Free net: A Distributed Anonymous Information Storage and Retrieval System". In Proc. ICSI Workshop on Design Issues in Anonymity and Unobservability, 2000

[7] Hermann de Meer and Ivan Dedinski, “Enabling mobile Peer to Peer networking”, Available: link.springer.com/chapter/10.1007%2F978-3-540-31963-4_16.

[8] H. Holma and A. Toskala, HSDPA/HSUPA for UMTS: “High Speed Radio Access for Mobile Communications”, Chichester, U.K.: Wiley, 2000.

[9] H. Liu, Y. Zhang, and Y. Zhou, “TailTheft: Leverageing the wasted time for saving energy in cellular communications,” in Proc. ACM MobiArch, Bethesda, MD, USA, Jun. 2011, pp. 31–36.

[10] M. Chatterjee and S. K. Das, “Optimal MAC state switching for CDMA2000 networks,” in Proc. IEEE INFOCOM, New York, NY, USA, Nov. 2002, pp. 400–406.

[11] S. Sesia, I. Toufik, and M. Baker, LTE, The UMTS Long Term Evolution: From Theory to Practice, Chichester, U.K.: Wiley, 2009.

[12] Thomas Mennecke. "How Over peer was able to corrupt data on the FastTrack network". 2005.

[13] T. Klingberg R. Manfredi, "Gnutella Protocol Development", June 2002. Available:http://rfcgnutella.sourceforge.net/src/rfc-0_6-draft.html.

[14] Wolfson, O. Dept. of Computer. Sci., Illinois Univ., Chicago, IL, USA. "An economic model for resource exchange in mobile peer to peer networks", Available: ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1311215

[15] Yan Yi, Zheng Yuan Su, Qing Jiang Zhao, Zu Cheng Dai, "Research of P2P Traffic Real-Time Monitoring Technology", The Scientific.Net - Available : http://www.scientific.net/AMM.340.451