

# Experimental Evaluation of Energy-Based Denial-of Service Attacks in Wireless Networks

William West, Emmanuel Agu

Computer Science Department , Worcester Polytechnic Institute

## Summary

Battery energy is frequently the most limiting resource on mobile devices. Energy-Based Denial-of-Service (e-Dos) attacks, in which malicious attackers initiate actions that can deplete the battery of a mobile node, have recently emerged. The feasibility of such attacks has been demonstrated at various layers of the network protocol stack and attention has been drawn to them. However, little work has been done to evaluate their effectiveness at different layers of a functional wireless network protocol stack or quantify the energy drain. In this paper, we investigate the energy profile of e-DoS packet flood attacks at the MAC, network, transport, application and physical (by moving the mobile node) layers in a real 802.11 wireless testbed. The AODV, OLSR and DSDV ad-hoc wireless routing protocols are tested. We gauge the effectiveness of our e-DoS attacks by measuring the increased energy consumption of the network card and estimating the resulting decrease in node/router lifetime. We investigate more sophisticated eDoS attacks that also write to the display and system hard disk that consume more energy than the network card. Finally, we demonstrate a combined eDoS attack on the network card, hard disk and system display that reduces battery lifetime by 25 percent. Our main goal is to gain a better understanding of this new emergent threat and also to inform new research in this field.

## Key words:

*Wireless LAN, Protocol security, Energy measurement, Ad hoc networks*

## 1. INTRODUCTION

WIRELESS networks are becoming widely deployed and truly ubiquitous. However, wireless networks are currently insecure for several reasons. Wireless signals frequently leak beyond the walls of buildings that they are intended to be used in. Thus, any mobile nodes within a certain range of a mobile client can receive its transmissions; even if encryption is employed, it only prevents passive eavesdroppers from deciphering the contents of the transmissions. Also, since mobile devices are designed to be carried around, they are prone to misplacement or theft, and mobility makes perpetrators of malicious attacks difficult to track. The Wireless Equivalent Privacy (WEP), which is the main standard that is used for securing wireless LANs has well-

documented vulnerabilities [16]. Moreover, research shows that most wireless LAN deployments do not currently use WEP or any other encryption standards. Anderson [7] has also reported that over 90 percent of security breaches result from inadequacies in physical security, including user carelessness and theft, which implies that security vulnerabilities will continue to exist even with current advances in cryptography and tamper-resistant hardware.

Denial-of-Service (DoS) security attacks attempt to deplete the resources of a victim, such as its memory or CPU cycles. DoS attacks affect mobile devices more since they already have limited resources such as memory, battery and CPU power, and wireless networks have limited bandwidths. While CPU and memory capacities have, on average, doubled every 18 months, battery technology has experienced minimal improvement over the past 30 years. As a result, battery life is frequently the most constraining resource on a mobile device. In fact, if malicious attackers can deplete a mobile victim's battery power, this can be considered a type of Denial-of-Service attack [7] since it leads to disruption of service that we refer to as the *Energy-based DoS (e-Dos) attacks*. This is especially true of multi-hop wireless networks, in which nodes every participant may also serve as a router.

Several authors have previously demonstrated the feasibility of e-DoS attacks on mobile devices via simulation [5] and measurement [2] [6] under simplistic scenarios. [26] proposes an intrusion detection system against eDoS attacks. Ad hoc networks have dynamic topologies, protocol overheads due to a higher level of participation of user nodes in route discovery and data delivery, and an increased number of nodes with limited battery power. Several questions remain open in this area and to the best of our knowledge no body of work has quantified the effects of e-DoS attacks on real ad hoc networks. Some unanswered questions that we address in this paper include:

- At what layers of the wireless protocol stack are e-DoS flood attacks most effective in ad hoc networks?
- Do certain protocols have intrinsic mechanisms (such as backoff in the MAC layer) that might exacerbate

- the attack or protect the node?
- Is there a relationship between protocol throughput and susceptibility to e-DoS attacks?
- When feasible, by how much can e-DoS attacks increase the energy consumption of the network card?
- By how much can the e-Dos attacks on ad hoc networks drain the battery power?
- Can we derive a model for e-Dos attacks that could predict future e-DoS security threats?
- Can we increase the effectiveness of eDoS attacks by attacking multiple energy-hungry subsystems (such as network card, hard disk, and display)?
- Are combined attacks on these multiple subsystems feasible, and how effective are they?

Our main goal was to systematically investigate how potent e-DoS attacks are, or could become in a real testbed. We orchestrate packet floods on an intermediate mobile router node in a wireless ad hoc network, and investigate the susceptibility of the 802.11 MAC layer, three popular ad hoc routing protocols (AODV, DSDV and OLSR), TCP and HTTP to e-DoS attacks. We also consider low node mobility speeds that are typical in an indoor environment, using a programmable robot since it is feasible that an attacker could forcibly move a mobile node if that was known to increase energy drain. Through our experiments, we are able to:

- Validate prior work in [1] on a real testbed. Specifically, we show that packet flood frequency and size are accurate predictors of node energy consumption.
- Rank protocols by energy overhead under mobility conditions that have been shown to maximize protocol overhead [3].
- Demonstrate that in practice, while some unsophisticated attacks could double the network card's energy consumption, overall battery drain is bounded on 802.11b/g mobile router laptops.
- Show that some naive e-DoS attacks on ad-hoc networks can actually lengthen node lifetime, albeit at the expense of throughput.
- Use measured parameters on laptops to extrapolate our results to encompass e-Dos attacks on smaller PDAs.
- Create a general model for e-Dos attacks on ad hoc

wireless networks and hence predict conditions under which e-Dos attacks are effective.

- Demonstrate dangerous eDoS attacks that attack multiple system energy-hungry components such as network card, hard disk and display.

The rest of this paper is as follows. Section 2 discusses prior work, section 3 describes our energy model, section 4 describes our mobility experiments, section 5 describes our e-DoS flood attacks, section 6 presents a simple model for e-DoS attacks with implications, section 7 describes attacks on the system disk and display and section 8 summarizes lessons learned and section 8 states our conclusions and future work.

## 2. related Work

e-DoS attacks were first described in [7]. [6] also demonstrates DoS attacks on real 802.11 wireless networks but considers only the MAC layer and does not take an energy-centric view. Our work is closely related to the work in [1] and [2]. We apply the per-frame energy model established by Feeney in [1], to estimate the energy consumption of mobile ad-hoc networks. In [2], the authors demonstrate e-Dos attacks on mobile devices that aim to drain the battery.

The conclusions of [2] are quite dramatic, showing live results of crippling effect of power attacks on palmtop computers. While [1] is primarily based upon simulation, it suggests that the potential overhead of ad-hoc routing protocols is not insignificant, and may be a factor in real-life implementation. While a live setup on the scale of the simulations in [1] might strain the limits of a typical research budget, a smaller setup such as [2] can still provide useful results. We set up a modest testbed that would enable us to test the hypotheses of [7] and [1] to determine if the worst-case power effects are of practical concern when current implementations of ad-hoc routing protocols.

## 3. Energy Model

We now expound our model to estimate energy consumption. In an ad-hoc configuration, an 802.11 network interface card may be in any of three energy states: transmit, idle, and receive [1]. For the transmit and receive states, the overhead energy consumption of each frame is modeled as a linear function of frame size  $x$ , with slope  $m$  and a fixed channel acquisition cost  $b$ :

$$E = mx + b, \quad (1)$$

where the energy,  $E$ , are different functions for packet reception,  $E_{Rx}$  and transmission,  $E_{Tx}$ , each with a mobile-device-specific slope and constant that must be obtained from direct energy measurements. The total energy,  $E_T$ , consumed by an interface in a given time interval,  $T$ , is then computed for all transmitted frames  $\frac{Q}{F}$  and received

frames  $\frac{R}{F}$  :

$$E_T = \sum_{q=1}^Q E_{Tx}(F_q) + \sum_{r=1}^R E_{Rx}(F_r) \quad (2)$$

Power  $P_T$  expended over interval  $T$ , is then calculated as:

$$P_T = E_T / T.$$

To experimentally determine the slopes  $m$  and constants  $c$  of  $E_{Tx}$  and  $E_{Rx}$  in equation (1), for our laptops, a Cardbus extender board was used similar to that in [1]. This enabled direct measurements of our network interface across  $V_{cc}$  during operation. The instrumentation configuration used for these direct measurements is represented in Figure 1.

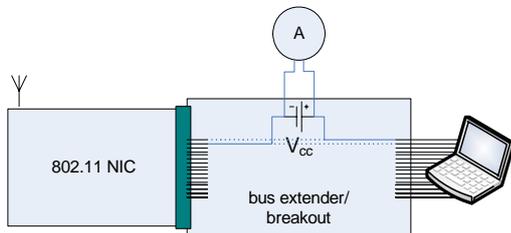


Figure 1 - Direct measurement of supply current using Cardbus extender.

While the author of [1] used a high frequency multi-meter to measure current, we found that accurate measurements were possible with an off-the-shelf ammeter by taking average current measurements over fixed intervals and frame rates. For each frame size from 20 bytes to 1500 bytes (as throttled via the interface's MTU setting), a current reading was taken. All readings for the transmission case were adjusted for the number of frames sent and linearly regressed using the least squares method in order to find the coefficients of equation (1); Error was calculated to be less than 1%. Measurements taken at different frame rates (1000 frames/s, 2500 frames/s, 5000 frames/s) offered consistent results. Reception, on the other hand, was found to vary in proportion to frame rate only; meaning that for our hardware, every received frame had the same fixed energy cost, regardless of size. Only

channel acquisition consumed energy in receive mode, and therefore only an increased frame reception rate could increase the rate of energy consumption.

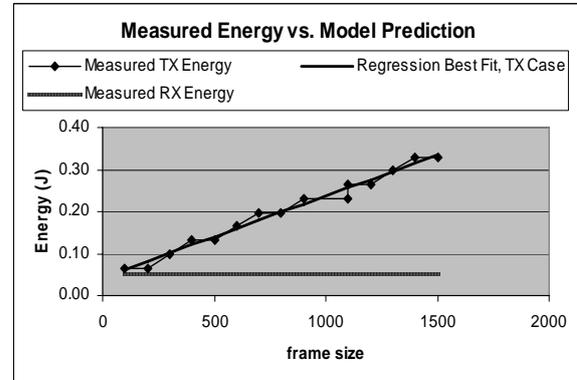


Figure 2 - Energy per frame measured Vs predicted for send and receive states

The transmission and reception coefficients obtained for equation (1) were then validated by comparing the calculated energy consumption to the battery drain observed on the host computer's power management interface (ACPI [17]), during similar test runs. Since ACPI provides relatively coarse data, longer sample times were used. Indeed, the energy consumed as reported by ACPI correlated well with those calculated using our energy model and the cardbus extender. Figure 2 shows our measured and predicted energy usage.

#### 4. experimental Testbed Setup

The nodes in our ad-hoc network were Dell Latitude D600 model laptop computers equipped with Netgear WG511 Cardbus 802.11b wireless network interface cards. All machines were equipped with Knoppix V3.3 [18] and a stripped, custom kernel built from the official Linux 2.4.28 sources. Except where noted, all machines used a recent CVS build of the *ndiswrapper* module along with the latest Windows drivers for the card. Various *libpcap*-based tools are used at each node to record traffic.

We tested implementations of the following popular ad-hoc routing protocols:

- *AODV*, or Ad-hoc On-demand Distance-Vector (Uppsala University AODV v0.8 [19])
- *DSDV*, or Destination-Sequence Distance-Vector (a module in MIT's *Click* modular router [20])
- *OLSR*, or Optimized Link State Routing Protocol (Uppsala University OLSR v0.4.8)

AODV and DSDV proved more stable than OLSR on our platform. While the OLSR implementation provided some useful results, it was apparently not correctly configured on our system and would periodically exhibit lack of service in our simple four-node arrangements. AODV and DSDV, were chosen because they differ in one fundamental characteristic – AODV is a reactive, on-demand protocol, while DSDV is a proactive, table-driven protocol. We felt that their responses to e-DoS attacks would be representative of their genre of protocols.

### Mobility experiments

#### 4.1 Mobility Testing Environment

Our first set of experiments was to test the stability of ad hoc protocols while the nodes were in motion. Mobility has previously been shown to maximize protocol overhead in ad hoc networks [3]. Moreover, a malicious attacker could physically move mobile devices if indeed mobility caused a form of e-DoS attacks. In order to methodically control node mobility, we employed a programmable robotic platform shown in figure 3.

Mobility experiments in [3] suggest that protocol overhead generated by handoffs, speed and node orientation matter less than range and frame size. With this in mind, we endeavored to maximize the variance in range and the number of handoffs.

The nodes were arranged in a column as shown in figure 4, with a mobile node oscillating up and down the 65 meter track at 0.5 m/s, varying the range and rapidly switching between one, two, and three hop topologies in an attempt to trigger handoffs and increase protocol overhead. In addition, periodic HTTP transfers were done between the mobile nodes  $N_4$  and  $N_0$ , as indicated in the dashed line in figure.



Figure 3 – Programmable robotic platform used in mobility tests

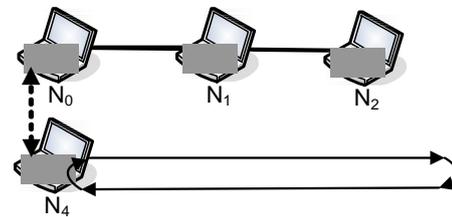


Figure 4 - Mobility testing topology

#### 4.2 Mobility results

Mobility was tested incrementally from the worst case (high robot mobility) to the best case (no robot mobility), with a static three-hop experiment as a base case. In the mobile case, the mobile node traversed the column three times, yielding a time interval of roughly 900 seconds for the shortest runs. After testing the static and high-handoff scenarios, we decreased routing overhead by introducing pauses in the robot’s movements. The robot’s pauses ranged from 0 seconds (no pause) and then in 5-second increments until the energy overhead approached that of the static case. Each run was conducted three times.

Energy expenditure for each run was calculated by feeding frame traces into equation (2). Drain time  $T_d$  was linearly extrapolated from idle power  $P_i$ , overhead power  $P_T$  (both in Watts) as calculated from equation (3), and total battery energy  $E_b$  (in Joules) as:

$$T_d = \frac{E_b}{(P_i + P_T)} \tag{4}$$

Tables 1 and 2 tabulate the average drain time of all nodes in the network by protocol. We found that at such a

low speed relative to the simulated nodes in [1] and [3], protocol overhead varied little between the mobile and static cases. If the rightmost columns are compared, it becomes evident that the drain times between the static case and the mobile worst-case are insignificant – at a per-second clock resolution, there is no discernible difference in predicted drain times. Pilot attempts to vary range, mobility pattern, and speed also did not seem to elicit any dramatic behavior, though it is worth noting that in both static and mobile configurations, AODV uses the least energy of all the ad hoc routing protocols considered, expending about 25 percent (4 Joules) less energy than DSDV and OLSR. However, these results are not normalized to goodput, and therefore do not give an indication of power efficiency relative to protocol performance.

Static Case – HTTP traffic, 3 hops		
Protocol	Energy (J)	Drain Time (s)
AODV	12.3	6972
DSDV	16.0	6969
OLSR	16.2	6969
idle	0.00	6981

Table 1 - Mobility experiments, static control case

Mobile Case – HTTP traffic, 1-3 hops		
Protocol	Energy (J)	Drain Time (s)
AODV	12.5	6972
DSDV	17.2	6968
OLSR	16.2	6969
idle	0.00	6981

Table 2 - Mobility experiments, mobile worst-case

There are two factors of note here: in ad-hoc mode, the protocol overhead is insignificant compared to the idle power overhead, which is constant. Additionally, the Netgear card represents, at a maximum, roughly 10% of system-wide energy consumption, which represents an upper limit on how much the network card can shorten a laptop's lifetime. More importantly, in ad-hoc mode, the constant overhead of listening for and sending packets in order to maintain cell associativity in a mesh dominates the overall energy cost, minimizing any impact of variation in traffic load, including protocol overhead.

### Energy-Based DoS e-Dos Attacks

We then attempted to orchestrate e-DoS attacks by flooding an intermediate node in a multi-hop network (see figure 5) from an attacker node with the sole intent of

draining its battery power and thus creating an e-DoS. The attack packets were sent at the application, transport, network and MAC layers of the protocol stack in order to compare the energy consumption at different protocol layers.

### 4.3 e-Dos attack Environment

For energy attacks, we switched to a static, two-hop topology as shown in figure 5 and reserved one node ( $N_4$ ) for the role of attacker. Again, a nominal traffic consisting of periodic 50KB HTTP transfers load was sent between two non-neighbor nodes  $N_0$  and  $N_2$ , while attacker  $N_4$  attempted to drain  $N_1$ 's battery with the intention of prematurely rendering a needed route useless.

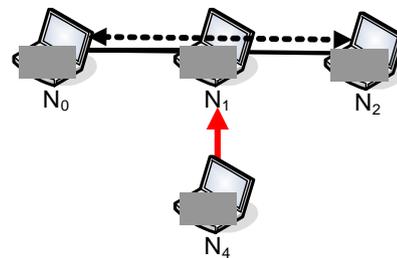


Figure 5. Power attack topology

In figure 5, node  $N_0$  initiated HTTP transfers and node  $N_2$  was a server that responded to the requests. It was hypothesized that flooding packets node  $N_1$  at different protocol layers and different rates might trigger reactions such as expensive routing updates, or 802.11 backoffs and retransmissions that could decrease its battery power and create a denial of service.

In these attacks,  $N_4$  is equipped with a Netgear WG511 interface with *ndiswrapper* in all cases except during link-layer specific attacks, where dual Netgear MA401 cards with *hostap/prism2* drivers are used; one for recording frames and another for launching attacks.

### 4.4 application layer attacks

For the application layer, we recreated the benign service attacks in [2]. This involved repeated legitimate requests of available network services, which might cause the targeted router to spend more energy listening, retransmitting, and re-establishing routes.

For 500 seconds, HTTP traffic is sent over our two-hop route while a fourth machine attacks. Each attack is performed three times at 7 different frequencies and for each of the 3 ad hoc routing protocols, for a total of 63

runs. Total packet energy expenditure is calculated for the router and plotted as an average in Figure 6. Battery drain predictions are calculated from these averages and plotted in figure 7.

At low frequencies, the attacks generally succeeded in increasing both card energy consumption and battery drain. At 10 attacks per second, the attacks double DSDV's card power consumption. However, beyond 10 attacks per second, there is an apparent drop-off in the effectiveness of the attack's ability to drain more power. As the victim node is overwhelmed with requests from the attacker, it stops actually servicing the replies to both the attacker and the non-attacking client. The result is that while queues build up, delays increase and throughput drops, power usage drops off markedly for these brute-force flooding attacks. The energy effect of increasing routing protocol chatter by sending small packets at a high frequency are minor compared to large HTTP downloads at a low frequency. The DSDV routing protocol sees the worst-case attack, which shortens the lifetime of the node by more than 20 seconds. It is worth noting at this point that while the router's survival time was the lowest in this case, DSDV yielded the best throughput. The OLSR case provides further evidence of an inverse relationship between routing throughput and node lifetime, since it fails early on in the attack and preserves the node's battery at the cost of not servicing traffic during the attack. In summary, higher-frequency attacks become reception floods, which are not as energy-intensive as the legitimate traffic. Only attacks that elicit more transmission from the target router are bound to be effective in lowering the life expectancy of the battery.

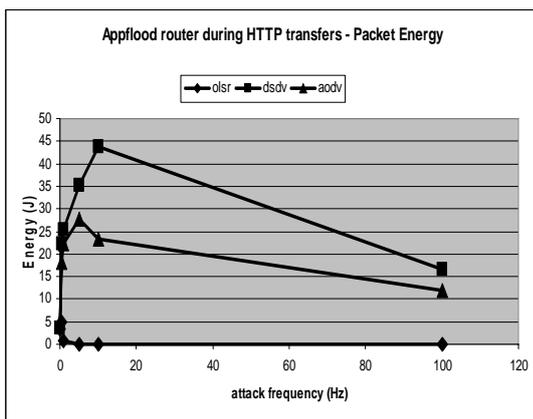


Figure 6 - Packet energy consumed plotted by increasing attack frequency in Hz (attack HTTP requests per second)

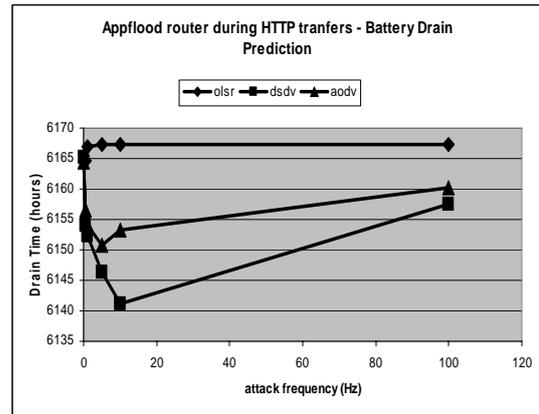


Figure 7 - Predicted battery drain caused by repeated HTTP requests plotted by attack frequency (HTTP requests per second)

### 4.5 Transport Layer Attacks

The traditional way for an attacker to consume resources and cause Denial of Service attacks at the TCP layer is by flooding the server with SYN packets. We initiate SYN floods to see if the resulting SYN/ACK responses combined with the increased packet reception and any added routing protocol chatter are costly enough to significantly affect battery life. We send the SYN packets to an available TCP service on the target router – in this case, the SSH server. Packets are sent with increasing frequency, until they are sent as rapidly as possible by the attacker.

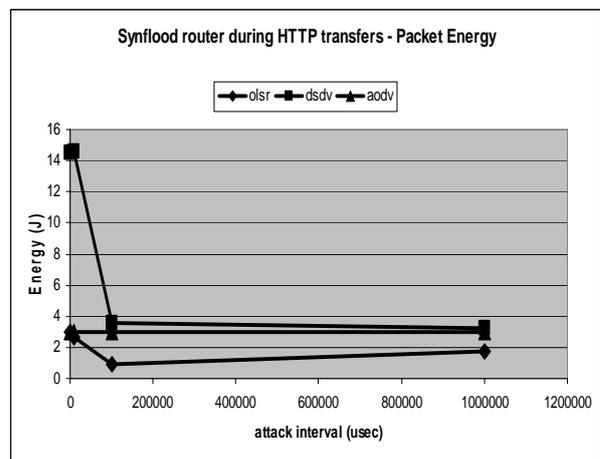


Figure 8 - Packet energy consumed plotted by increasing attack interval in microseconds (interval between attack SYN packets)

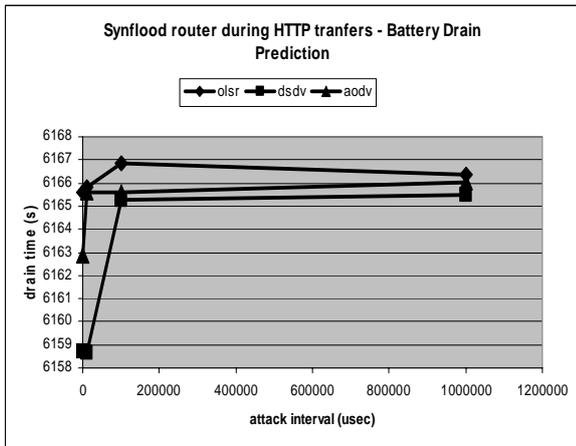


Figure 9 - Predicted battery drain caused by repeated SYN packets plotted by attack interval (interval between attack SYN packets)

Figure 8 shows that card energy consumption actually drops for the SYN floods and the battery drain time increases slightly. Web traffic is only affected to a limited degree, so no marked decrease is observed as in the application layer attacks. Energy consumed by the attack is only about half as much as in the application attack case, and the resulting lifetime reduction is also halved to about 10 seconds, as can be seen in Figure 9.

#### 4.6 Network Layer Attacks

Next, we flooded OLSR, DSDV and AODV with packets to see if they had any routing behaviors or flaws that would trigger any expensive energy usage. Since it was evident from our energy model and the subsequent experiments that transmission was expensive, an attacker that wants to drain a target battery must endeavor to cause it to transmit more and larger frames. Figures 10 and 11 show our results for ad hoc routing floods.

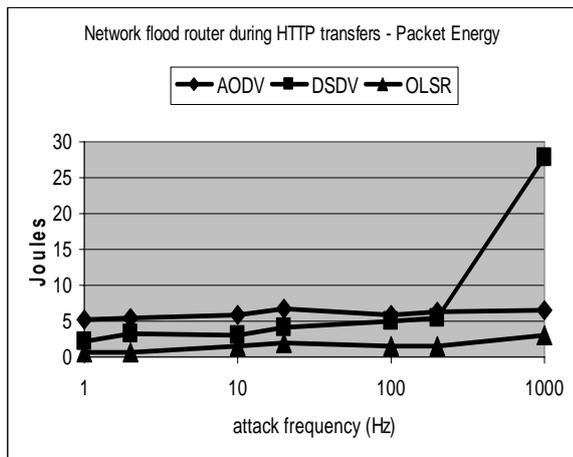


Figure 10 - Packet energy consumed plotted by increasing attack frequency in Hz (protocol attack packets per second)

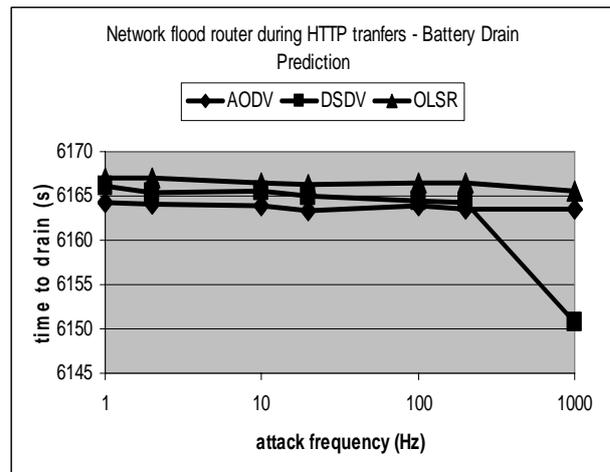


Figure 11 - Packet energy consumed plotted by increasing attack frequency in Hz (protocol attack packets per second)

#### AODV RREQ Attack

AODV floods route requests throughout the network. By repeatedly making a valid AODV route request (RREQ) for a non-existent node, we can force the target node to re-transmit our request to its peers. Since AODV maintains sequence numbers to prevent duplicate servicing and flooding loops, our attack must properly observe sequence numbering of the RREQ packets. This attack also falls under the benign service attack category [2]. The only difference between a non-malicious node and our attacker is that our program makes the request at an increasingly high frequency. A simple C program was used to generate and broadcast our malicious AODV RREQ packets and to throttle flood frequency. However, the effect proved to be insignificant in that it did not measurably affect either card energy consumption or node longevity.

#### DSDV Table Update Attack

DSDV nodes periodically broadcast table updates. The *Click* implementation provides a parameter for throttling these updates. The attack as we employed it could be much improved if it were possible to quickly change the local routing table before each table update packet – this might force the target to flood the update to its neighbors due to the change. As it stands, we did not implement the attack in this way, so any cost is due mostly to additional receive overhead. This attack proved to be the most successful, approaching the application layer attacks at high frequencies in terms of effect on energy consumption, as shown in Figures 10 and 11.

### OLSR HELLO Flood

The OLSR attack is similar to the DSDV attack in that while it takes advantage of the target's willingness to receive valid OLSR packets (in this case, HELLO packets), it is not smart enough to modify routing information and force retransmission. Though OLSR has an optimized flooding mechanism, for localized attacks, the effect should in theory be just as pronounced as with other traditional flooding protocols. The results were similar to the AODV attack – both card energy consumption and node longevity are not measurably affected.

### 4.7 Link Layer Attacks

Traditionally, denial of service attacks at the MAC layer involve denying other stations access to the medium by exploiting the carrier sense multiple access (CSMA) medium control scheme, which demands that stations honor any declared request-to-send (RTS) by backing off for a period of time before transmitting. Additional RTS packets indicate a threat of collision, and back off times increase. Flooding with RTS silences other stations. While this is effective at consuming the medium as a resource, it should not do the same for the battery. Since all transmission would be silenced by this attack, the only cost would be the reception of extra RTS frames and increasingly infrequent CTS responses. Indeed, that was the case. Even at very high frequency, RTS flooding does not quite approach the resource utilization of benign service attacks at the application level, though it does shorten the battery life by about 10-15 seconds. The backoffs triggered by the RTS floods actually forced the receiving node into a receive mode whose energy consumption is lower than transmission and in effect saves its battery power. Figures 12 and 13 show our results for the MAC layer floods.

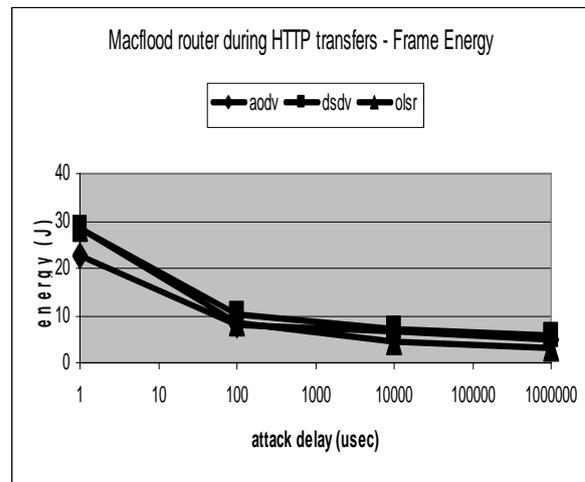


Figure 12 - Frame energy consumed plotted by increasing attack interval in microseconds (interval between attack RTS frames)

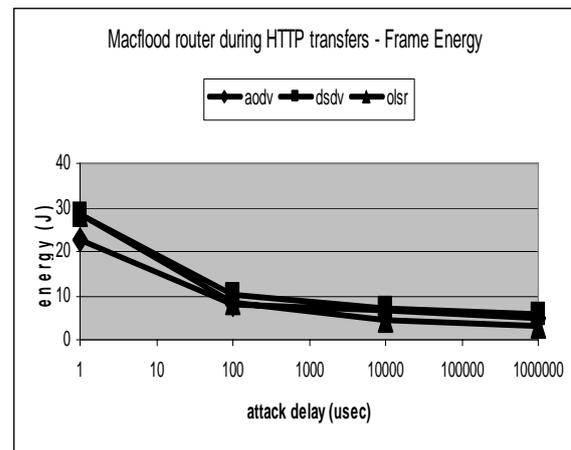


Figure 13 - Packet energy consumed plotted by increasing attack frequency in Hz (protocol attack packets per second)

### 4.8 modified attacks

Since it became apparent that forcing the target to transmit was causing a considerably larger power drain, we eliminated the delay, and made the attacker to transfer a large file in order to generate a continuous stream of transmissions. We also surmised that a machine with a smaller battery and more energy conserving peripherals would see a greater impact. We thus tried to extrapolate our results to a PDA with smaller AAA batteries using constant-power data from the energizer website [15]. We applied Peukert's battery discharge model [14] to the tested laptop and a contrived PDA case based on power levels reported for an IPAQ in [2], plus packet energy.

The Peukert model [14] (drain time = delivered charge/current<sup>n</sup>, where n is a constant for the battery) is

not perfect in the case of laptops because there is a power regulator maintaining constant wattage, meaning that as the battery drains the current increases inversely with the battery's voltage drop and over short intervals of wattage, it is almost indistinguishable from the linear extrapolation, in shape and amplitude.

Figures 14 and 15 show plots of the drain times for the attack and idle cases as a function of power consumption. Drain time decreases with each attack, though the shallow slope in Figure 14 indicates a mild impact. The drain time varies by only one second for the laptop.

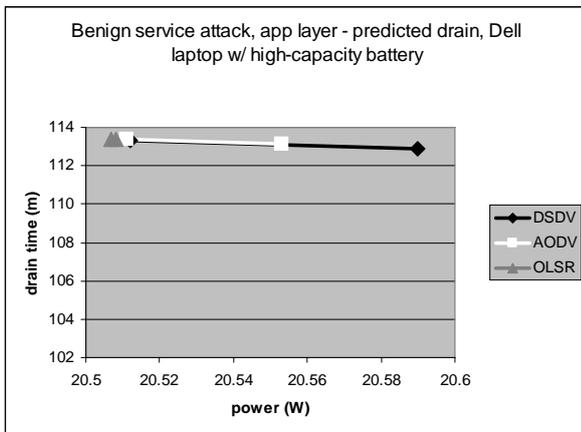


Figure 14 – Laptop router idle and attack consumption plotted for each protocol Vs drain time. Slope indicates severity of attack impact.

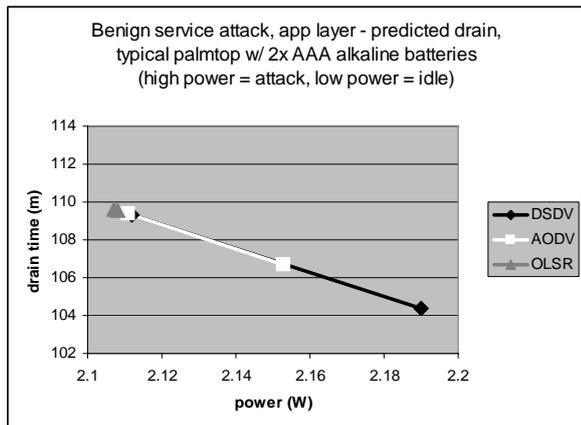


Figure 15 – PDA router to idle and attack consumption plotted for each protocol versus drain time. Slope indicates relative severity of attack impact. Compare with laptop case Figure 14 of same scale.

The IPAQ power utilization was computed from reported wireless card idle power consumption (1.2W) compared to total power consumption with card active (2.0W) adjusted for our card's power idle consumption (0.3W) plus packet power costs. Data from [2] was used as a guideline for idle

power measurements. Comparing figures 14 and 15, we see that the results for the PDA are indeed more pronounced than those seen on the laptop. Overall, since a PDA such as the IPAQ has a smaller screen, solid-state storage, more power-saving features, and smaller batteries, network power consumption plays a larger role in affecting node failure time.

### A model for e-Dos attacks

Based on our experiments, observed behaviors and measurements, we then tried to build a simple model for e-DoS attacks. Such a model would not only succinctly summarize our results, but could be used to also predict variants of new e-DoS strategies that could fatally impact mobile devices. We employ a duty-cycle-based formulation that is typically used for characterizing energy usage on mobile devices [2]. The duty cycle  $D$  expresses the fraction of time a device is on.

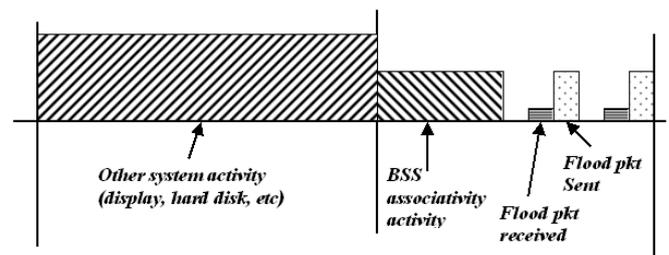


Figure 16 – Summary of power-usage on mobile router node.

Figure 16 visually summarizes our findings in our energy-based DoS attacks. We found that about 85% of system energy was spent on other sub-systems including the display and hard disk. At most 15% of the energy was spent on the network card, of which BSS associativity and management packets consumed a significant amount of energy. Thus, based on the derivations in [21],

$$Average\_card\_power = P_{sleep} (1 - D) + P_{avg\_active} D \tag{5}$$

Where we define  $P_{sleep}$  is the power expended when the mobile device is in sleep mode (typically = 0), and  $P_{avg\_active}$  is the average amount of activity (transmit and receive) that we generate from flood packets. Thus,

$$P_{avg\_active} = P_{idle} (1 - A) + (P_{Active\_TX\_RX} + P_{system}) A \tag{6}$$

where  $P_{idle}$  is the power expended in idle mode,  $A$  is the fraction of time that the mobile device is active including system activity, the time during which the node is transmitting or receiving packets, including flood, regular

HTTP traffic and BSS associativity packets. Thus, we further define  $P_{Active\_avg\_TX\_RX}$

$$P_{Active\_avg\_TX\_RX} = P_{TX} \cdot T_{TX} + P_{RX} \cdot T_{RX} + P_{BSS} \cdot T_{BSS} + P_{Traffic} \cdot T_{Traffic} \quad (7)$$

Where  $P_{TX}$  is the power expended in transmitting flood packets and is proportional to the flood packet lengths,  $P_{RX}$  is the power expended in receiving flood packets and we found this to be a constant.  $P_{BSS}$  is the power expended by transmitting BSS associativity packets and  $P_{traffic}$  is the power expended by the node in transmitting regular (e.g. HTTP) traffic. Likewise,  $T_{TX}$  is the time spent in transmitting flood packets,  $T_{RX}$  is the time spent in receiving flood packets,  $T_{BSS}$  is the time spent in transmitting BSS associativity packets and  $T_{traffic}$  is the time spent by the node in routing regular (e.g. HTTP) traffic. We also note that:

$$T_{TX} = T_{RX} = \frac{\text{bytes\_in\_flood\_pkt}}{\text{transmission\_speed} (11Mbps)} \quad (8)$$

We note that of all the variables in  $P_{Active\_TX\_RX}$  only the flood packet and receive times,  $T_{TX}$  and  $T_{RX}$  (which are inversely related to A) are under our control in the flood attacks. We may then relate the card transmit speed to its receive speed by defining the ration  $TFR$  as

$$TFR = \frac{T_{TX}}{T_{RX}} \Rightarrow T_{TX} = TFR * T_{RX} \quad (9)$$

So for instance, if the card transmit and receive speeds are equal,  $T_{TX} = T_{RX}$ , and  $TFR = 1$ . We can rewrite

$$P_{Active\_avg\_TX\_RX} = T_{RX} (P_{TX} \cdot TFR + P_{RX}) + P_{BSS} \cdot T_{BSS} + P_{Traffic} \cdot T_{Traffic} \quad (10)$$

And putting it all together,

$$P_{Avg} = P_{sleep} (1 - D) + D \left[ \frac{P_{idle} (1 - A) + A (T_{RX} (P_{TX} \cdot TFR + P_{RX}))}{+ P_{BSS} \cdot T_{BSS} + P_{Traffic} \cdot T_{Traffic} + P_{system}} \right] \quad (11)$$

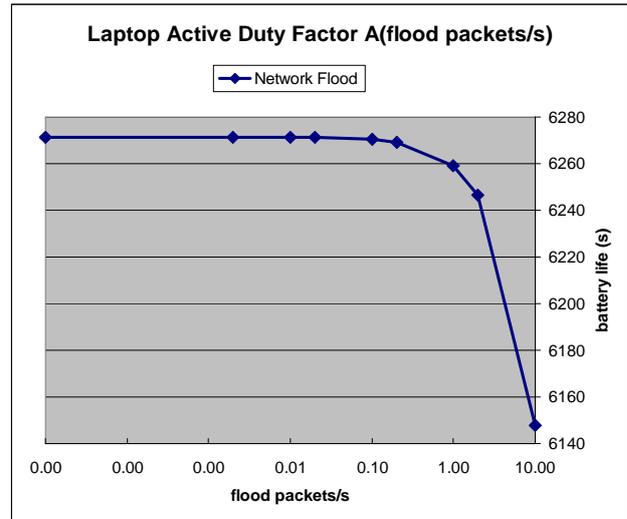


Figure 17 - Battery life trend under network layer flood attacks of increasing frequency using active duty factor model.

Figure 17 shows our model parametrized using data from our test laptop under a network layer flood of attacks. Having created and parametrized this model, in subsequent sections, we can simply use this model to inform our experiments.

#### 4.9 implications of our model

By inspecting equation 11, it can easily be seen that the mobile device is more vulnerable to power drain as its duty factor,  $D$  increases or as it is switched on more often. In our case, the machine is always on, but it is instructive to allow for the more general case that mobile devices are switched off for long periods. Also, since only  $T_{TX}$ ,  $T_{RX}$  and  $A$  are under our control, the attacker can only increase the power expenditure mainly by:

- Reducing idle time (or increasing flood packet length to give longer  $T_{TX}$ ,  $T_{RX}$  and  $A$ ).

However, as we have seen in our experiments, BSS associativity activity on the card limits how much we can increase  $T_{TX}$  and  $T_{RX}$ . We can also comment on factors in the expression that may make the attack more potent. Several of the factors that maximize  $P_{Avg}$  are not in the attacker's control, but merely a consequence of using a particular mobile device. The attacks will drain more card energy if it happens that:

- The mobile device is switched on more often, increasing the duty factor,  $D$
- The network card has high idle power  $P_{idle}$ , transmit power  $P_{TX}$  or receive power  $P_{RX}$  or high  $TFR$  (transmit to receive power expenditure)

The power expenditure of  $P_{Sleep}$ ,  $P_{idle}$ ,  $P_{BSS}$ ,  $P_{traffic}$  and time or duration variables such as  $T_{BSS}$  and  $T_{traffic}$ , are out of our control.

As a final comment, we note that our model enables us to predict some more potent attacks. For instance, if these attacks could also increase  $P_{system}$  by increasing system activity on more energy-hungry sub-systems such as the display or hard disk, overall energy consumption could be increased.

### Attacks on energy-hungry device components

Having attacked the mobile device through the network interface, one realization was that the effectiveness of e-DoS attacks was somewhat dependent on how much power the device consumed. In this section, we attack the mobile computer's hard disk and display to determine the impact of e-DoS attacks on them. In both cases, we cause the attacker to additionally write to the hard disk and display in order to drain the mobile device's power.

First, we measured each device's percentage of total system energy usage on a laptop with 20.5W idle power, and found that the min-max ranges were:

Device	Percentage Energy Usage (min-max)
OLED display	7-35
2.5" hard drive	4-11
Netgear Wi-G NIC (adhoc)	2.5-12
Netgear Wi-G NIC (BSS):	1.5-12

Table 3 – Percentage Energy Usage of each component.

Table 3 shows that the display consumes the most power, followed by the hard drive. We hypothesized that e-DoS attacks on these components shall be more deadly than attacks at any layer of the network protocol stack.

#### 4.10 e-DoS attacks on mobile's hard disk

In this section, we try to drain more system energy by additionally writing to the device's hard disk. This attack is also significant since several computer viruses and Trojans already write to the mobile hard disk in today's world.

Power consumption for disk writes was done by picking a transfer rate (e.g.  $R_{tx} = 100$  KB/s) and using the maximum sustained transfer rate  $M_{tx}$  of the drive as an optimistic metric for calculating the amount of time the disk would spend in write mode when required to transfer, on average  $R_{tx}$  KB/s. The rest of the time, the drive is assumed to be in low-power idle mode (another optimistic boundary assumption).

$T_w$  = Fraction of time spent in write mode is  $1/(M_{tx} / R_{tx})$  sec

$T_i$  = Fraction of time in idle mode is  $(M_{tx} / R_{tx} - 1)/(M_{tx} / R_{tx})$  sec

Idle drive power is already included in active duty factor system power consumption, so only the write power is additional in these cases.

$P_w$  = power overhead in write mode = 2.5W (average) [1]

$P_w * T_w$  = power utilized as a function of  $R_{tx}$

For  $R_{tx} = 100$ KB/s,  $P_w = 0.004$  W, for  $R_{tx} = 200$ KB/s,  $P_w = 0.009$  W. Our experiments were repeated using values of (e.g.  $R_{tx} = 100, 200$  KB/s and 56.3MB/s). The disk write rate of 56.3 MB/s is the maximum speed that the disk can sustain.

Figure 18 shows our results for the disk write eDoS attacks. The worst case disk write decreases battery lifetime by over 500 seconds or 8%

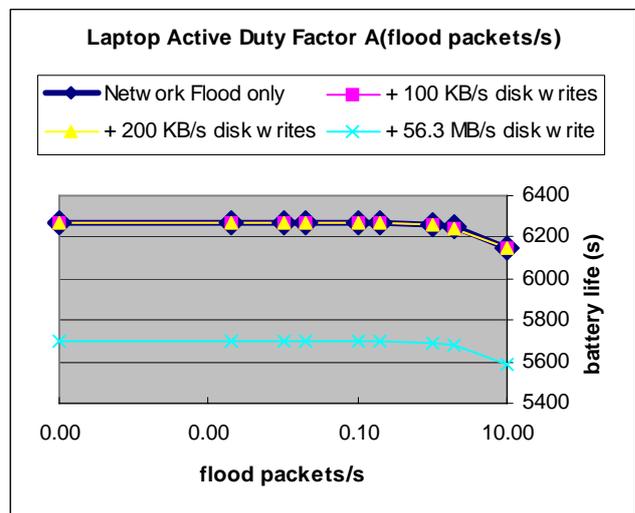


Figure 18 – Battery life trend under network layer flood attacks of increasing frequency – with no disk activity, 100 KB/s disk writes, 200 KB/s disk write, and worst-case 56.3 MB/s disk writes.

### 4.11 e-DoS attacks on mobile’s display

Table 3 already indicated that the mobile device’s display was the most energy hungry component, we also expected attacks on the display to have a significant effect on system energy usage. Thus, we orchestrated attacks that wrote to the mobile device’s monitor in addition to network packet floods. Finally, we combine the results of network flood attacks with maximum disk write rates and writing to the system display.

[25] gives a model for the power consumed by new OLED-based displays as:

$$\text{DisplayPower (Watt)} = P_{\text{CONTROLLER}} + P_{\text{DRIVER}} + \text{PanelPower}$$

$$\begin{aligned} \text{PanelPower} &= \text{PixelArrayPower} \\ &= \sum P_{\text{RED}} \cdot \text{pixel}_R + P_{\text{GREEN}} \cdot \text{pixel}_G + P_{\text{BLUE}} \cdot \text{pixel}_B \end{aligned}$$

where  $P_{\text{CONTROLLER}} = 0.5$  Watt,  $P_{\text{DRIVER}} = 1.0$  Watt

(for OLED display tested in [25]) and  $\text{pixel}_x = [0,1]$  (0 = fully off, 1 = fully on) . The authors of [25] also did direct power measurement to determine the coefficients for their display hardware:

$$P_{\text{RED}} = 4.3, P_{\text{GREEN}} = 2.2, P_{\text{BLUE}} = 4.3 (\mu\text{Watt}).$$

It is clear that attacks on OLED-based displays should therefore aim to maximize the on-states of the sub-pixels under its control; essentially, the opposite of the object of the some of the adaptive strategies described in [25] could be used to increase power without degrading the user experience in ways that may suggest the presence of an attack. For example, we created the following eDoS attacks on OLED displays using this information:

*Bright-scale attack* – brightest pixel turned on, other pixels scaled by same amount

*White-scale attack* – all pixels under attack turned on (probably the most conspicuous)

*RB-scale attack* – R and B set to average value of R, G, and B; G set to zero.

Assuming a 1024x768 pixel display and 600x400 white (RGB=[255,255,255]) pixels in-focus and the remainder of the pixels at RGB=[50,153,153] we can compare the effectiveness of each attack in terms of battery life.

No attack:

$$\begin{aligned} &2.40\text{E}5 \text{ white pixels} \\ &\quad (2.59 \text{ W}) \\ &5.46\text{E}5 \text{ “attack“ pixels} \\ &\quad (3.30 \text{ W}) \end{aligned}$$

5.89 W (ref. power)

Bright-scale attack:

$$\begin{aligned} &2.40\text{E}5 \text{ white pixels} \\ &\quad (2.59 \text{ W}) \\ &5.46\text{E}5 \text{ bright-scaled pixels, RGB}=[84,255,255] \\ &\quad (4.32 \text{ W}) \end{aligned}$$

6.91

W ( $\Delta+1.02$  W)

White-scale attack:

$$\begin{aligned} &2.40\text{E}5 \text{ white pixels} \\ &\quad (2.59 \text{ W}) \\ &5.46\text{E}5 \text{ white-scaled pixels, RGB}=[255,255,255] \\ &\quad (5.90 \text{ W}) \end{aligned}$$

8.49 W ( $\Delta+2.60$  W)

RB-scale attack:

$$\begin{aligned} &2.40\text{E}5 \text{ white pixels} \\ &\quad (2.59 \text{ W}) \\ &5.46\text{E}5 \text{ RB-scaled pixels, RGB}=[119,0,119] \\ &\quad (2.21 \text{ W}) \end{aligned}$$

4.80 W ( $\Delta-1.09$  W)

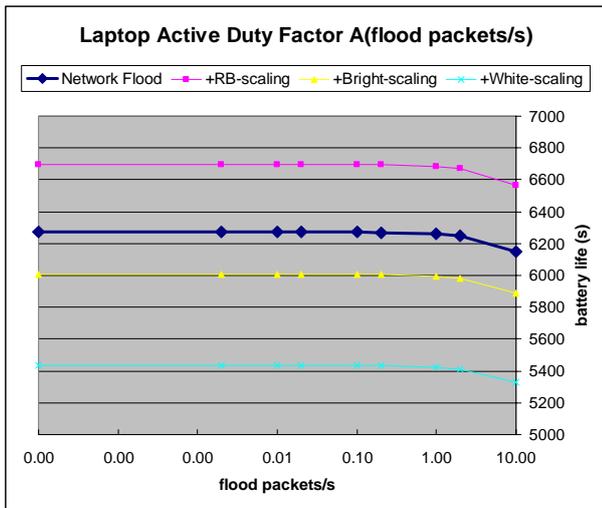


Figure 19 – Battery life trend under network layer flood attacks of increasing frequency – with no display attack, RB-scaling, brightness scaling, and white-scaling.

Figure 19 shows our results for all our described attacks on the display. RB scaling is the least effective, actually saving power. It can be imagined, however, that in a case where a green-heavy color scheme is used, the RB-scale “attack” might negatively impact battery life.

White-scaling is the least-stealthy attack conceived, but it is the most effective because it represents the theoretical maximum of the power model. Battery life is decreased by an additional 13% compared to the network flood-only case.

Display power for the OLED ranges from 1.5 – 8.5 Watts (7-40% of system power). The attacks as-modeled, however only affects pixel array above and beyond a typical average (we assume a typical display utilization of 5.89W, which yields an attack overhead range of 0-2.6W, or 0-13% total system power).

The introduction of adaptive display technologies creates the possibility for attacks, since non-adaptive displays were already inefficient by comparison. Malicious programs which somehow gain the permission to directly or indirectly affect the display (e.g. a malicious web page or trojan) could be crafted to limit or negate the benefits of adaptive color display schemes.

#### 4.12 Combined e-DoS attacks of network floods, hard disk and display writes

Finally, we combine attacks of network packet floods, hard disk and display writes. . If an implementation of a

disk and display attack is run as part of or in concert with a network flood, battery life could be significantly reduced, even on a relatively energy-dense portable system such as a laptop.

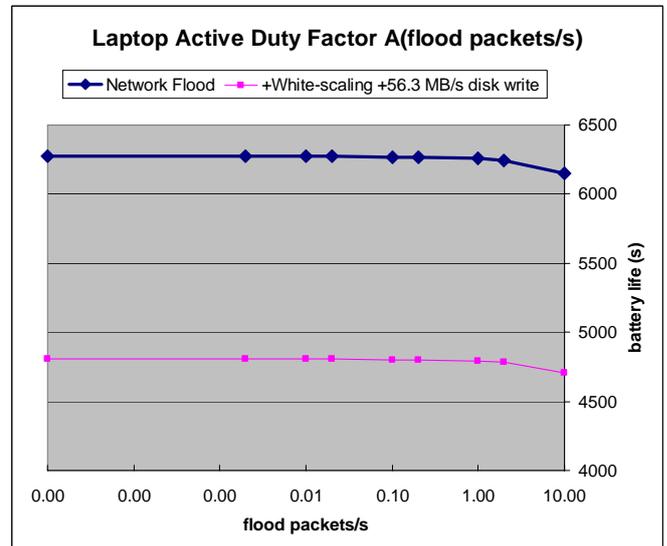


Figure 20 – Battery life trend under network layer flood attacks of increasing frequency – with most effective display attack (white-scaling) and most effective I/O attack (max speed disk write).

Figure 20 shows our results for the combined e-DoS attacks. The attacks reduces total battery lifetime by as much as 25 percent – this translates into 26 minutes of lost battery lifetime for our example configuration. We have also previously noted that the battery density of a laptop is by far the densest of all mobile devices. We feel that these combined attacks would cripple a PDA or cell phone in a much shorter time. On the other hand, these devices have smaller screens and some have flash cards instead of hard disks. Thus, the investigations of other devices require further experimentation.

Since the model does not include things such as increased CPU overhead, fan utilization, and controller overhead, it does not represent a theoretical upper limit to power attacks. Nevertheless, the components of the model do constitute the majority of the power-hungry devices in a mobile computer, and therefore provide a good basis for estimating the potential scope and effects of power attacks, including those not yet conceived.

#### lessons learned, implications and predictions

Our experiments led us to conclude the following:

1. *Battery density matters:* Our results were more dramatic for PDAs with lower battery densities than laptops.
2. *Naïve attacks can actually SAVE battery power:* We found that although most of the attacks were a nuisance by reducing throughput and increasing delay, some unsophisticated attacks forced the mobile node into lower power idle (saturated queues) or receive states (backoffs), in which they actually saved power.
3. *Protocols with good throughput characteristics are most susceptible to e-DoS attacks:* Since card transmission power contributes most to card energy consumption, protocols with good throughput spend more time transmitting and are more susceptible to e-DoS attacks.
4. *Some protocols have intrinsic protection against e-DoS attacks:* Some implementations of ad hoc routing protocols such as OLSR, actually shut off completely when subjected to high rate packet floods, avoiding further battery drainage. Although this reaction was not intended to be part of the initial protocol design, we found this to be a protective side effect.
5. *e-DoS attacks will be most deadly in protocols that incorporate sleep modes:* Although we were successful in moving the router nodes from idle states into a higher transmit states, the difference between sleep modes (zero power) and the idle, receive or transmit states is more dramatic. Hence, we believe that e-DoS attacks will be deadliest as sleep-deprivation attacks that stop the mobile nodes from going to sleep.
6. *Low-speed mobility does not significantly affect battery drain:* These results corroborate the earlier assertions in [1] and [3] that the low speeds typical in indoor environments do not affect battery drain rates. Transmission range and mobility patterns also do not matter at these low speeds. Thus physically movement of the mobile node by the attacker will have a limited e-DoS effect.
7. *There exist upper bounds on network-centric energy-based attacks:* We found that the energy consumed by the NIC was about 10% of total system-wide energy consumption. [11] and [12] have shown that other system components such as the display and hard disk consume large portions of battery energy. We note that more sophisticated attacks (or trojans) that also display images or write to hard disk (e.g., spyware) may significantly increase mobile battery drain.
8. *Network flood attacks that elicit more mobile node transmission are most effective:* We found that prolonged transmission rather than attack frequency was one of the biggest culprits with regard to card energy consumption on the mobile nodes. Hence, any attacks that cause the mobile device to transmit packets drain the most battery power.
9. *Mobile device configuration matter:* Based on our model in section 6, we clearly see that many influencing factors such as the card idle, transmit and receive power and the contribution of network card energy consumption to overall system-wide energy usage, but are out of our control.
10. *eDoS attacks are most effective on devices that consume a lot of power:* we demonstrated that eDoS attacks on the hard disk and display that were more energy hungry than the network card were more effective. In fact, a combined attack on all the network card, display and hard disk could reduce battery lifetime by 25 percent.

### conclusion and future work

We have analyzed the viability of energy-based flood attacks that are emerging in the literature. We found that much of the energy expended on a network card when operating our ad hoc network was in maintaining the connection to ad hoc peers without the ability to go into sleep/power save mode.

The trends displayed here show that while physically moving mobile nodes at low speeds is minimally effectual in terms of energy overhead, flood attacks can increase energy consumption and have a potential to cause some battery drain. The high cost of idle ad hoc power consumption somewhat masks e-DoS effects. Indeed, since CPU-related power overhead is neglected in this study, and trends seem to exist in the frame power consumption, the CPU effects could be larger on a more primitive CPU, and more so in a system such as a PDA where CPU time is typically one of the biggest energy hogs.

Finally, we found that the most effective eDoS attacks were attacks on energy hungry subsystems such as writes to the disk or display. We demonstrated combined eDoS attacks on the network card, hard disk and display that reduced battery lifetime by 25 percent.

## References

- [1] L. M. Feeney, "An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks," in *Mobile Networks and Applications*, 6<sup>th</sup> ed, The Netherlands: Kluwer Academic Publishers, 2001, pp. 239-249.
- [2] Thomas Martin, Michael Hsiao, Dong Ha, Jayan Krishnaswami, "Denial-of-Service Attacks on Battery-powered Mobile Computers," *Proceedings of the 2nd IEEE Pervasive Computing Conference*, Orlando, Florida, March 2004, pp. 309-318.
- [3] G. Ravikiran and S. Singh, Influence of mobility models on the performance of routing protocols in ad-hoc wireless networks, *Proceedings of IEEE Vehicular Technology Conference '04*, May 2004.
- [4] N. R. Potlapally, S. Ravi, A. Raghunath and N. K. Jha, "Analyzing the energy Consumption of Security Protocols", in Proc. Intl' Symposium on Low Power Electronics 2003, pp. 30-35.
- [5] Gupta V, Krishnamurthy S, Faloutsos M, Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks, in *Proc. IEEE MILCOM 2002*
- [6] Bellardo J, Savage S, 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions, *In Proc. USENIX Security Symposium*, August 2003
- [7] Stajano F, Anderson R, The Resurrecting Duckling: Security Issues for Ad Hoc Wireless Networks, in *Proc. International Workshop on Security Protocols*, Cambridge, United Kingdom, 19-21 April 1999.
- [8] Wood A D, Stankovic J, Denial of Service in Sensor Networks, *IEEE Computer*, October 2002, pp. 48-56
- [9] Cardenas A, Radosavac S, Baras J, Detection and Prevention of MAC layer misbehavior in ad hoc networks, in *Proc. 2<sup>nd</sup> ACM Workshop on Security in Ad Hoc and Sensor Networks 2004*
- [10] Kyasanur P, Vaidya N, Detection and Handling of MAC layer Misbehavior in Wireless Networks, in *Proc. International Conference on Dependable Systems and Networks, June 2003*.
- [11] Flinn J, Satyanarayanan M, PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications, in *Proc 2nd IEEE WMCSA*, New Orleans, Louisiana, February 1999.
- [12] Banerjee K, Agu E, PowerSpy: Fine-Grained Software Energy Profiling for Mobile Devices, in *Proc IEEE WirelessCom Symposium on Mobile Computing, Maui, Hawaii, 2005 (to appear)*.
- [13] Avalone S, Botta A, Emma D, Guadagno S, Pescapè, D-ITG V.2.4 Manual, *University of Napoli "Federico II", Naples, Italy December 2004*.
- [14] Peukert W, *Elektrotech, Z.*, 18, p. 289, 1897
- [15] Energizer batteries, [www.energizer.com](http://www.energizer.com)
- [16] Jon Edney, William A. Arbaugh, "Real 802.11 Security: Wi-Fi Protected Access and 802.11i", Addison-Wesley, 2003
- [17] Advanced Configuration and Power Interface, [www.acpi.info](http://www.acpi.info).
- [18] Knoppix Live Linux CD, [www.knoppix.org](http://www.knoppix.org)
- [19] Uppsala Univ. AODV v0.8, <http://core.it.uu.se/AdHoc/AodvUUImp/>
- [20] The MIT Click Router Project, <http://pdos.csail.mit.edu/click/>
- [21] Jayan Krishnaswami, Denial-of-Service Attacks on Battery-Powered Mobile Computers, *Master's Thesis, Virginia Tech*, February, 2004
- [22] Travelstar 5K80 hard disk drives specifications <http://www.hitachigst.com/hdd/support/5k80/5k80.htm>
- [23] The PC Guide . Hard Disk Reference Guide: Internal Sustained Transfer Rates. <http://www.storagereview.com/map/lm.cgi/str>
- [24] HP Flat Panel Monitor Specs. <http://h10010.www1.hp.com/wwpc/ie/en/sm/WF04a/20491-156249-156249-156249-169267.html>
- [25] S. Iyer, L. Lu, R. Mayo, P. Ranganathan. Energy-Adaptive Display System Designs for Future Mobile Environments. Submitted to MobiSys May 2003: San Francisco, CA
- [26] Jacoby, G.A.; Davis, N.J., Battery-based intrusion detection, in Proc. IEEE Globecom 2004