DETOUR: Delay- and Energy-Aware Multi-Path Routing in Wireless Ad Hoc Networks

Nadine Shillingford, David C. Salyers, Christian Poellabauer and Aaron Striegel
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556
{*nshillin, dsalyers, cpoellab, striegel*}*@nd.edu*

***Abstract*—Streaming real-time applications require the timely distribution of information in mobile ad-hoc and sensor networks. At the same time, such networks must operate energy-efficiently to maximize the lifetime of mobile devices and applications. In multi-hop networks, multiple communication paths between a single sender and receiver can be established, with varying real-time and energy characteristics of each path. This paper introduces the DETOUR (Delay- and Energy- aware mulTi-cOUrse Routing) protocol that applies feedback-driven path diversification, where traffic load is balanced across two or more paths to ensure both timeliness and energy-efficiency. We apply the (m,k) model for firm real-time communication to wireless networks, i.e., the protocol aims to meet at least m end-to-end deadlines out of k packet transmissions, thereby sacrificing additional improvement in latency in order to maximize the lifetime of the network by minimizing energy consumption. The experimental results of this paper show the protocols ability to reduce energy consumptions (up to 35%) while meeting the data streams firm real-time constraints.**

I. INTRODUCTION

Wireless mobile ad hoc networks (MANETs) are important in today's communication systems. However, MANETs have high mobility, network size and a complex topology making individual nodes dependent on auxiliary power supplies such as batteries and generators. DETOUR aims to use path diversification to reduce energy consumption in MANETs while ensuring timely communication with limited overhead.

Bouckaert, Bergs and Naudts [3] describe a situation where a MANET is constructed as a search and rescue response to a building fire. The reconnaissance team of firemen, equipped with wireless mobile devices, enter the burning building while the crisis team is located away from the scene to direct the rescue effort. The firemen use their mobile devices to relay images of the fire and to communicate to the crisis team. The crisis team can assess the data being sent from each fireman and give instructions to each one based on a collective consensus of the situation. The wireless devices used by the reconnaissance team and the crisis center form a multi-path MANET.

In this scenario, energy management and timely communication between the reconnaissance team and the crisis team are important. It is difficult for the crisis team to make a decision when the quality of transmission is below a required threshold. Meanwhile, a depleted battery from one of the devices means that the crisis team is unable to assess the fire in that area of the building.

DETOUR aims to exploit path diversification to maximize multiple competing constraints (such as energy management and timely communication required in the previous scenario). This paper focuses on two constraints: real-time communication and energy management. The first refers to the *timely* communication between source and sink nodes and the latter refers to choosing the path(s) with the lowest energy costs or the path(s) with the nodes with the largest energy reserves.

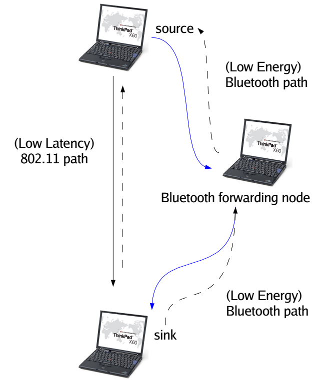

Fig. 1. Low-energy (LE) and low-latency (LL) paths in a MANET.

Our protocol is based on a feedback system: the destination node (sink) informs the source of the network conditions of different paths between the source and the sink. The source relies on this feedback information to decide on a particular path that satisfies the given communication constraints (low-energy and low-latency). This idea is represented in Figure 1.

More specifically, we apply the (m, k) model for firm real-time systems, where packet delays have to be limited. However, occasional late information is acceptable. The (m, k) model [8] is frequently used to express these constraints. Out of any window of k packets, we aim to receive m within specified deadlines and $k - m$ can be late. For applications whose accuracy depends on the number of packets received, late reception of data is *preferred* over simply dropping $k - m$ packets, which is the simplest way to preserve energy. In such

This work is supported in part by NSF under grant number CNS-0545899

environments, the successful arrival of all required data is vital to application performance and effectiveness. For instance, in a remote surveillance application, when packets are late, retransmission of these packets can significantly improve the quality of the surveillance (or may even be essential for the archival to support retrospective surveillance), *even if the packets arrive after their end-to-end deadlines*. Another scenario where late arrivals are profitable is a live streaming environment with multicast connections in which the receivers also act as media recorders [7]. In this case, a late frame arrival can improve the recorded media quality though it may have violated the timeliness requirements for the live streaming process.

DETOUR is a novel approach which uses path diversification to balance energy conservation and timely communication. In order to achieve this balance, DETOUR looks at all possible paths from source to sink and focuses on a combination of two main values: (i) the power consumption of each node and (ii) an estimate of the number of future late/on-time transmissions based on previous packet transmissions and feedback from the sink. DETOUR uses these two values to determine the correct distribution of packets by ensuring that a minimum number of packets is sent on any high energy path but by being constantly aware that the on-time transmissions are above a particular threshold.

II. RELATED WORK

The issues of energy management and timely communication has been studied by researchers such as Durresi, Paruchuri and Barolli [5], leading to solutions such as the DSR routing protocol [6]. Subbarao [9] developed a protocol called minimum power routing (MPR). The aim of MPR is to select the path between a given source and destination that will require the least amount of total power expended, while still maintaining an acceptable signal-to-noise ratio (SNR) at each receiver. One major difference between DETOUR and MPR is that MPR data must constantly be circulated around the network. This results in increased cost and contributes to network congestion. In DETOUR, the number of packets to be transmitted per path is calculated on demand. The feedback packets from source to sink are the only form of control packets transmitted. This method reduces the overhead of control packets.

Agarwal, Katz and Joseph [1] propose a routing protocol that performs path selection based on the cost of communication, rather than the latency of communication. Our results from DETOUR indicate that it is important to consider both the energy cost of communication as well as the latency.

Djukic and Valaee [4] conducted studies which focused on path diversification to provide low probability of packet loss in wireless networks. In their protocol, the source uses forward error correction (FEC) codes to encode each packet into multiple fragments. The result is a guarantee on reliability and control of the energy used to transmit packets. DETOUR differs from this protocol in that it reduces the overhead applied to each packet by removing the need to include the parity K required for each packet of M fragments. In addition, the Djukic model sends each packet on multiple paths. Although this may result in higher reliability, it increases the congestion of the network as well as the energy costs.

III. DETOUR ROUTING

Several configurations can exist that require delay- and energy-aware multiple path communication. In this section we explore two such configurations. First, we consider an architecture of sensor networks, i.e., a network consisting of low-power, low-cost motes[1] and more powerful gateway devices (e.g., Crossbow's Stargates[2]). Table I compares the characteristics of both the 868/916MHz radio transceivers used by motes and the IEEE 802.11 devices used by the gateways and sinks.

TABLE I
A COMPARISON OF 868/916MHZ RADIO RECEIVERS AND 802.11 WIRELESS INTERFACES.

Parameter	868/916MHz	IEEE 802.11 b/g
Max. Outdoor Range	500 feet	< 1200 feet
$I_{transmit}$	$27mA$	$< 545mA$
$I_{receive}$	$10mA$	$< 424mA$
I_{sleep}	$1\mu A$	$< 18mA$

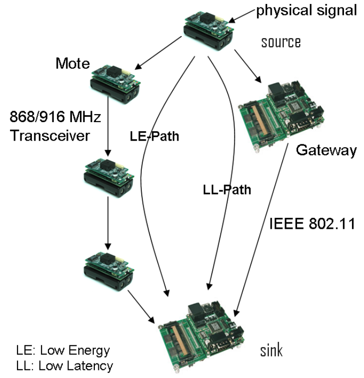

Fig. 2. Low-energy (LE) and low-latency (LL) paths in a WSN.

The 802.11 devices can transmit up to 1200 feet, while the motes can only transmit up to 500 feet (in open areas). In addition, in transmit mode the 868/916Mhz transceivers draw a current of only 27mA compared to up to 545mA of the 802.11 devices. These devices can therefore be configured into the scenario shown in Figure 2. In this example, the two paths shown are as follows: (a) the sensor data can travel across multiple low-power motes until it reaches the sink, or (b) the sensor data can travel across a 802.11 gateway device to the sink. Considering the numbers from Table I, the left-hand path is more energy-efficient, while the right-hand path will be faster.

---

[1] A mote is a micro electromechanical sensor (MEMS) that is able to communicate over a wireless network using a transceiver.

[2] http://www.xbow.com

The second scenario is taken from the area of MANETs. Mobile devices rely on different communication technologies, e.g., WiFi, Bluetooth, Zigbee, or Infrared. For example in our driving scenario introduced in Section I, the devices held by the reconnaissance team can establish paths using low energy 802.15.1 (Bluetooth) interfaces and low latency 802.11 interfaces (Figure 1). IBM Bluetooth 2.0 interfaces have a maximum speed of 3Mbps while current 802.11 interfaces have a speed of up to 54Mbps[3]. In addition, the range of the 802.11 interfaces can reach a maximum of about 1200 feet while the Bluetooth interfaces can only reach a maximum of 100 feet. In terms of energy consumption, the average current consumption of the Bluetooth interface is about 60mA while 802.11 interfaces use up to 545mA. Therefore, if latency is an issue the communication between two 802.11 interfaces is going to be faster. If energy consumption is an issue, the Bluetooth interfaces are preferable.

The DETOUR system model exploits the presence of multiple paths and their differences in both energy and delay characteristics. The quality of service (QoS) parameter (m, k), which is specified by the application or the user, expresses that in each window of k packets, m should be on time. In this paper, we assume that our packets are uniform in size and type. We intend to explore non-uniform packets in a future study. In the next two subsections, we explore (i) two-path DETOUR consisting of a low-energy path and a low-latency path as laid out in Figure 1 and (ii) multi-path DETOUR consisting of n paths where $n > 2$.

A. Two-Path DETOUR

We can express the probability that a packet sent across a low-latency path will be on time as p_{ll} and the probability that a packet sent across the low-energy path will be on time as p_{le}. Our general assumption is that the transmission is non-lossy, that is, that any lost packets are retransmitted. A feedback packet is sent from the sink to the source (over path i) after path i receives a specified number of packets referred to as a *fixed path specific window*, k_i. We use a fixed or 'jumping window' for the purpose of this paper. However, an analysis of the conversion from a fixed window to a sliding window is given in a later section. The feedback contains the probability for successful transmissions, p_i for each path. The probability is calculated using the following equation:

$$p_i = \frac{r_i - l_i}{k_i} \tag{1}$$

where r_i is the total number of packets received, l_i is the number of late packets and k_i is the path specific window.

Figure 3 is the result of a mathematical model designed to illustrate the probability of successful transmissions of m packets out of a window of k packets as a fraction of the number of packets transmitted across the low-latency path. It compares the probabilities of having at least m out of k successful transmissions for a 2-path connection with path probabilities $p_{ll} = 0.8$ and $p_{le} = 0.5$.

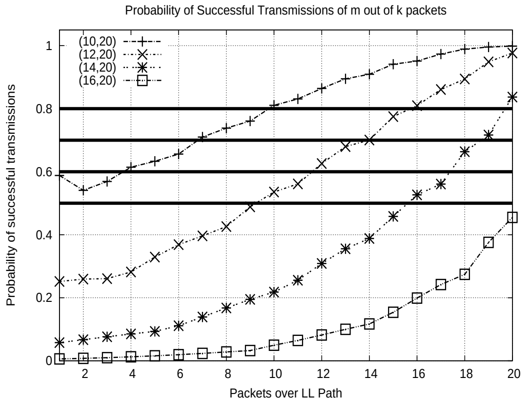

Fig. 3. Probability of successful transmission of m packets out of a window of k packets over the number of packets transmitted across the low-latency path (x-axis), with varying values for (m, k).

The straight lines indicate the corresponding 'target' probabilities, i.e., the probability should be at or above this line. The goal is to select the value for k for the low-latency path as small as possible, i.e., select a value where the probability is at or above the target probability and to the furthest left possible (leaving the largest number of packets for the low-energy path). For the remainder of this paper we refer to the value of k for the low-latency path as k_{ll} and the value of k for the low-energy path as k_{le}.

For example, for $(m, k) = (10, 20)$, all probabilities are above the target line at 0.5, i.e., we can select $k_{ll} = 0$ and $k_{le} = 20$. In other words, all packets will travel across the low-energy path and none will travel across the low-latency path. For $(12, 20)$, the probability graph cuts the target line at 11, i.e., $k_{ll} = 11$ and $k_{le} = 9$. In situations such as shown for $(16, 20)$, the probability curve is always below the target graph, indicating that the (m, k) target will not be reached. Here, the goal is to select the maximum probability possible, e.g., for $(16, 20)$, we select $k_{ll} = 20$ and $k_{le} = 0$, thereby giving us the best chance of reaching the $(16, 20)$ constraint.

Based on the analysis of the mathematical model, we derive formulas for DETOUR. Using the two per-path probabilities (p_{ll} and p_{le}), we compute the discrete probability distribution, P of on-time transmission of at least m packets in a window of k packets for each path:

$$P_{le} = \sum_{i=m_{le}}^{k_{le}} \frac{k_{le}!}{(k_{le} - i)! * i!} * {p_{le}}^{i} * (1 - p_{le})^{k_{le} - i} \tag{2}$$

$$P_{ll} = \sum_{j=m_{ll}}^{k_{ll}} \frac{k_{ll}!}{(k_{ll} - j)! * j!} * {p_{ll}}^{j} * (1 - p_{ll})^{k_{ll} - j} \tag{3}$$

where

$$m_{le} + m_{ll} = m \tag{4}$$

---

[3] Wireless-N (draft 802.11n) is proposed to transmit at a maximum of 540Mbps

and

$$k_{le} + k_{ll} = k \quad (5)$$

for a given (m, k) constraint.

The goal is to distribute traffic among the available paths to achieve the (m, k) constraint while minimizing the energy overheads associated with communication. Using the previous formulation, we aim at maximizing the traffic across the low-energy path and minimizing the traffic across the low-latency path:

$$P_{(m,k)} = P_{p_{ll}} * P_{p_{le}} \quad (6)$$

and the goal is to ensure

$$min(k_{ll}) \quad (7)$$

The source determines the values for k_{le} and k_{ll} such that $P_{(m,k)}$ is greater or equal to m/k. This is shown in Algorithm 1.

Algorithm 1 Probability Computation

```
1:  INPUT: The $(m,k)$ constraint and per-path probabilities
    for successful transmissions feedback $p$;
2:  find_k_ll($p_{ll}, p_{le}, m, k$):
3:      $k_{le} = k$; $k_{ll} = k - k_{le}$;
4:      while($k_{le} \geq 0$):
5:        $m_{le} = m$; $m_{ll} = m - m_1$;
6:        $P_{le} = 0.0$; $P_{ll} = 0.0$;
7:        $k_{max} = 0$;
8:        $P_{max} = 0$;
9:        while($m_{ll} \geq 0$):
10:           for($j = m_{ll}$; $j \leq k_{ll}$; $j = j + 1$):
11:               Execute Equation 2;
12:           for($i = m_{le}$; $i \leq k_{le}$; $i = i + 1$):
13:               Execute Equation 3;
14:           Execute Equation 6;
15:           if($P_{total} > P_{max}$):
16:               $k_{max} = k_{ll}$;
17:               $P_{max} = P_{total}$;
18:           if($P_{total} \geq \frac{m}{k}$):
19:               return $(k_{ll})$;
20:           $m_{ll} = m_{ll} - 1; m_{le} = m - m_{ll}$;
21:           $P_{le} = 0; P_{ll} = 0$;
22:       $k_{le} = k_{le} - 1; k_{ll} = k_{ll} + 1$;
23:   return $k_{ll} = (k_{max})$;
```

The function $find\_k\_ll$ takes four arguments: the path probabilities p_{ll} and p_{le} (monitored by the sink), and the m and k values for the (m, k) constraint. This algorithm performs a simple linear search for the cut-line shown in Figure 3 (a more efficient approach using advanced search algorithms can be used instead).

The search begins with $k_{le} = k$ and $k_{ll} = k - k_{le}$ (line 3), i.e., with all packets traveling across the low-energy path. As long as the computed probability is below the target probability, we continue to execute the while loop. In lines 5 and 6 we initialize the m values for both paths and the probabilities for both paths.

In the while loop between lines 9 and 21, we compute the probabilities for both paths for successful transmissions of *at least* m_{le} packets on the low-energy path and *at least* m_{ll} packets on the low-latency path. The probability computation follow Equations 2 and 3 and the total probability (both paths combined) shown in line 14 follows Equation 6.

Lines 18 and 19 return the value of k_{ll} if the total probability P_{total} is above the target probability. Otherwise, further searching is done with new values for m_{le} and m_{ll}. Finally, if no solution has been found, the algorithm returns the largest k_{ll} value that resulted in the largest total probability.

In the event that no packets are sent on a path, that is k_{ll}=0 or k_{le}=0, DETOUR periodically (e.g., every 3rd window) transmits packets across this path to trigger the recomputations of the probability of this path, to allow them to be considered again in the future.

On the source side, the computed path-specific (m_i, k_i) constraints are now used to schedule packets among the i paths. That is, the values for k_i indicate the number of packets to be transmitted across path i and therefore can be interpreted as *weights*.

A naive strategy could be to simply send k_{le} packets over the low-energy path and k_{ll} packets over the low-latency path. However, that will lead to increased congestion on forwarding nodes on these paths. It is therefore preferable to evenly distribute the load among all paths, i.e., the packet transmissions on all paths should be spaced out as far as possible. To achieve this, a simple *Earliest Deadline First (EDF)*-style approach can be chosen where the $1/k_i$ values are interpreted as periodic deadlines and packets are scheduled on the paths with the currently smallest deadline.

Algorithm 2 Two-Path (m,k) Scheduling

```
1:  INPUT: The overall $(m,k)$ constraint and the computed
    path $(m_i, k_i)$ constraints (from Algorithm 1);
2:  init():
3:          $d_{ll} = \frac{1}{k_{ll}}$;
4:          $d_{le} = \frac{1}{k_{le}}$;
5:  select_path():
6:      $j = min(d_{ll}, d_{le})$;
7:      schedule $path_j$;
8:      $d_j = d_j + \frac{1}{k_j}$;
9:  if($d_{ll} > \Delta$ && $d_{le} > \Delta$):
10:     $d_{ll} = d_{ll} - \Delta$;
11:     $d_{le} = d_{le} - \Delta$;
```

Algorithm 2 shows this approach; lines 2-4 initialize the pseudo-deadline for each path to its value of $1/k_i$ determined by Algorithm 1. In lines 5-8, the smallest value of d_i is then determined, i.e., the min function performs a search and returns the index for the path with the smallest deadline. Then, the packet is transmitted across this path in line 7 and in line 8, the new deadline is computed by adding the packet's pseudo period k_j. If the deadlines for both paths go above some constant Δ, we reset both deadlines by subtracting Δ. We do this to prevent overrunning numbers. In our experiments, we use Δ=10000 for our evaluation.

B. Multi-Path DETOUR

DETOUR can be generalized to n different paths with varying levels of energy and latency. For example, we can extend the scenario in Figure 1 so that there are multiple nodes dynamically configured to represent multiple paths.

The feedback mechanism of DETOUR is used to determine the number of on-time packets that are received at the sink. However, we require a method to calculate a suitable transmit power level for each hop in a path. This transmit power level can be used as a cost function in the DETOUR path selection mechanism. A possible method for calculating transmit power level is indicated in [2]. DETOUR can then use these transmit power values as cost functions for each path. Since DETOUR is mainly a route selection algorithm we need to pair DETOUR with a preexisting on-demand routing protocol. For the purpose of this discussion, we use the dynamic source routing protocol (DSR) [6]. The implementation of DETOUR with DSR requires a minimal degree of changes on DSR.

Each node keeps a record of the total amount of power needed to reach a destination for each path. We modify the *route discovery* process of DSR so that as the Route Request (RREQ) packet heads downstream to the destination we keep a total of the transmit power levels, $\tilde{P}_{TX}$, of each node. Instead of storing the paths (also called source routes in DSR terminology) from a source to a destination, in increasing order of number of hops, a combination of DETOUR and DSR stores the paths in ascending order of total transmit power levels.

When a source needs to send packets to a sink it applies Algorithm 1 to the feedback from the sink for the lowest energy path and a combined probability of all the other paths in the routing table. This results in a k value for the lowest energy path. The lowest energy path is then removed from consideration. Next, we select the second lowest energy path in the table. This path is compared to all the other paths.

Again, this results in a k value for this second lowest energy path and it is removed from consideration. The process is continued iteratively for each of the remaining paths. Consequently, DETOUR is applied so that as many possible packets are sent over the lowest energy path, then as many as possible over the second lowest energy path and so on. The result is that the highest energy path receives the lowest number of packets[4]. Algorithm 3 shows this approach.

In addition to the changes already mentioned, we make a further modification to DSR by adding a special feedback packet that will be used to send data back to the source. This is because, as it stands currently, DSR does not have an efficient mechanism for performing this task.

The modifications to DSR mentioned in this section increase the routing traffic but, as observed in the results of the 2-path experiments in Section V, the energy savings cancel out this overhead.

Algorithm 3 Generalizing Algorithm 1

```
INPUT: A prioritized list of all the paths, the number of paths (N), the $(m, k)$ constraint and per-path probabilities for successful transmissions $p_i$;
$k'$ = $k$;
for($a = 1$; $a \leq N$ and $k' > 0$; $a = a + 1$):
    $P = 0.0$;
    for($b = a + 1$; $b \leq N$; $b = b + 1$):
        $P* = p_b$;
    $k_a$ = find_k_ll($p_a$, $P$, $m$, $k'$);
    $k'$ = $k$ - $k_a$;
```

C. Reemergence of Broken Links

There is a possibility that a link may get broken during execution. For example, one node leaves the range of the radio of the other device. Once this happens, the algorithm needs to deal with this situation. We periodically check for this occurrence.

When a link on a path i gets broken, packets are sent over the other paths to achieve the correct m and k values for the (m, k) constraint. When path i is reestablished, the source needs to be alerted so that packets can be transmitted over path i (if it is selected by Algorithm 2). However, since path i has been broken, its deadline (as calculated in Algorithm 2) is small in comparison to the other paths. To remedy this situation, we set the deadline of any broken path to infinity. Therefore, once path i is reestablished (DETOUR becomes aware of a reestablished link when the source receives feedback from the sink from this path), we recalculate its deadline. We achieve this using the following calculation:

$$d_i = max(d_{i'}) + \alpha; \tag{8}$$

where d_i is the deadline of the reestablished path, $d_{i'}$ is the deadline of the other paths, and α is a constant. In our experiments, we set α=1. Next, we recalculate the deadlines based on the feedback from the sink and the new deadlines d_i and $d_{i'}$.

D. Jumping vs Sliding Windows

Our algorithm attempts to satisfy the (m, k) constraint for a fixed window. However, this algorithm can be adapted to a sliding window. In [10], the authors discuss a formula for determining a corresponding sliding window constraint. Using their formula and discussion we derived a corresponding formula for DETOUR. Out of k packets m will arrive on time, that is, that k-m can be late. Therefore, if no more than k-m packets can be missed in a fixed window of deadline k then correspondingly no more than 2(k-m) deadlines are missed in a sliding window of k+(k-m) = 2k-m deadlines. Therefore, to derive a sliding window constraint, (m', k'), from a fixed

---

[4] Ideally, no packets should go over the low-latency/high energy path.

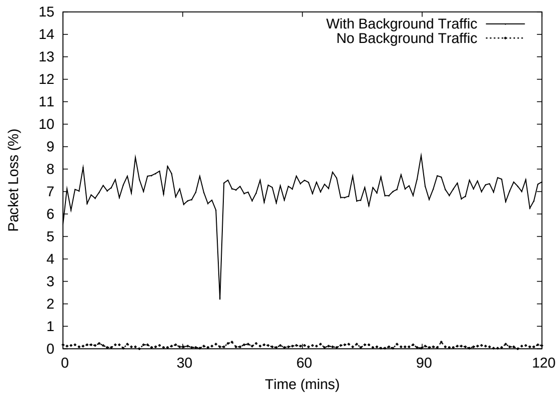

Fig. 4. Average loss rate for 802.11b

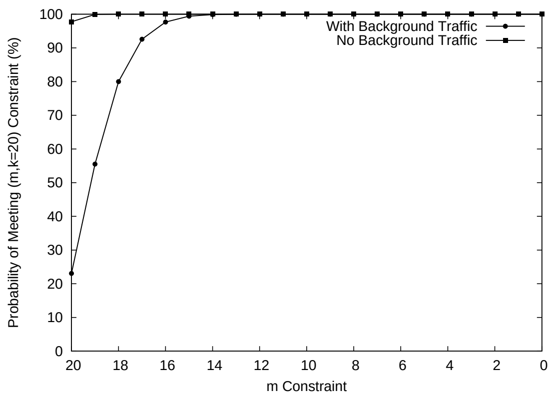

Fig. 5. CDF of probability of meeting (m,k=20) constraint

(m,k) constraint we would need to use the formula:

$$(m',k') = \frac{2\,(k-m)}{2k-m} \tag{9}$$

IV. WIRELESS RELIABILITY

An assumption of DETOUR is that it adapts to changes in wireless networks. Previous thought has been that wireless networks are unpredictable. In this section, we supply a sample of data that we collected from an ongoing extensive study of the wireless medium at the University of Notre Dame. Currently, results of this study suggest that wireless networks are by no means random. Although it is difficult to predict the pattern of behavior for individual packets we can make such a prediction during a window of packet transmissions. We performed a simple experiment with four 802.11b equipped laptops. We took measurements between two laptops communicating with a CBR UDP stream of 615Kb/s (1400 byte packets) over a distance of seven feet. We used broadcasts so that any failed transmission would not be retried and no bandwidth was consumed with MAC layer acknowledgments. The first set of tests measured the performance in terms of loss and delay when no contention on the wireless medium exists. For the second set of tests, the other two laptops were used to provide contention on the medium with long term TCP flows.

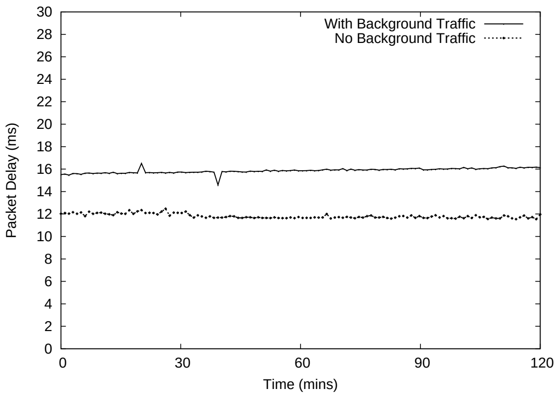

Fig. 6. Average delay for 802.11b

In Figure 4, the average measured loss rate in shown with and without background traffic. As can be seen there is little loss (less than 1%) when background traffic is not present. When background traffic is competing for the wireless medium, the loss rate only varies between 5.5% - 7.5%. These tests indicate an observable level of consistency in wireless performance in terms of loss over a long period of time under steady conditions.

However, overall loss rate only gives part of the overall performance picture. The other aspect of performance in relation to loss is how the losses are grouped. Figure 5 gives this information by showing a cumulative distribution of the probability of meeting various (m,k) constraints where k=30. For example, it can be seen that when there is no background traffic there is a 99% chance of meeting an (m,k) constraint of (20,20). With the introduction of background traffic there is only a 22% chance of meeting a constraint of (20,20), however there is a 99% chance of meeting a constraint of (15,20). With a smaller (m,k) window (such as k=5, not shown), there is a 40% chance of a constraint of (5,5) will be met. In general (for a given value of losses per winodw), as k is decreased there is a higher probability of meeting the specified constraint. However, for a given m, the probability of meeting an (m,k) constraint is increased as k increases.

The final aspect of the performance picture to be considered is end-to-end packet delay, shown in Figure 6. For both cases, the end-to-end delay shows little variation over time. However, when background traffic is present the average packet delay is about 3.5ms longer. Since retransmissions are not a factor due to the broadcast, this extra delay occurs because each packet spends more time in the queue waiting for the wireless

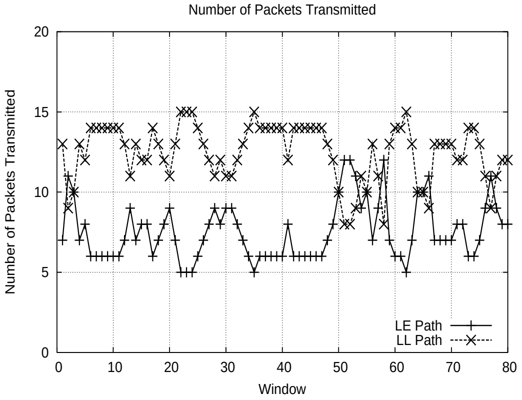

Fig. 7. Distribution of packets over the low energy (LE) and low latency (LL) paths for a deadline of *30ms* and an $(m,\ k)$ constraint of (15,20).

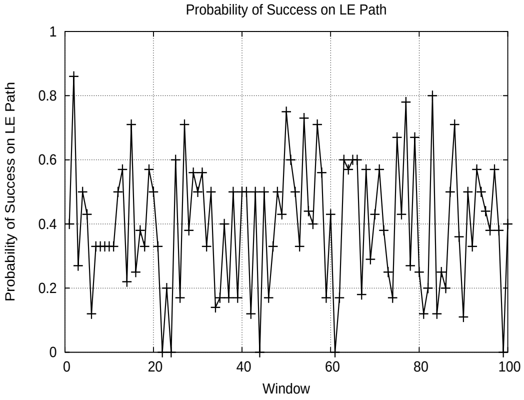

Fig. 8. Probability of on-time arrival on the LE path for an (m,k) constraint of $(15,20)$ and a deadline of 30ms.

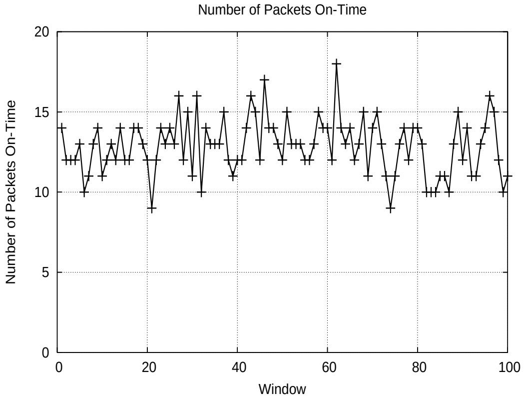

Fig. 9. Total number of on-time packets for an (m,k) constraint of $(10,20)$ and deadlines of 30ms.

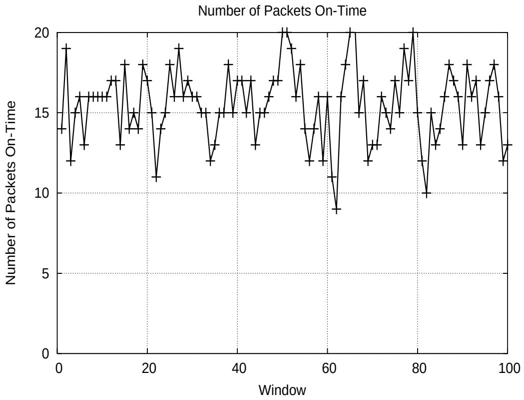

Fig. 10. Total number of on-time packets for an (m,k) constraint of $(15,20)$ and deadlines of 30ms.

medium than when no background traffic is present.

Overall, three areas of wireless performance have been investigated: loss, loss pattern, and delay. This sample of our results indicates that while the wireless network is an unreliable medium, it is possible to achieve consistent performance under stable conditions.

V. EXPERIMENTS AND RESULTS

The experiments described in this section were performed on two Lenovo Thinkpad X41 laptops and one Lenovo Thinkpad X60 laptop. Each of the three laptops is equipped with an internal Intel PRO/Wireless IEEE 802.11b/g network card and an internal Bluetooth card. We performed real time experiments using the configuration in Figure 1. Packets are sent from source to sink based on the distribution calculated in Algorithm 1 and 2. Each packet sent from source to sink contains the deadline which it has to meet.

Figure 7 shows the distribution of packets over the two paths. The upper line shows the number of packets that were transmitted across the low-latency path and the bottom line shows the number of packets transmitted across the low-energy path, both for a deadline of 30ms and an (m,k) constraint of $(15,20)$. The x-axis shows the progress in terms of windows, i.e., during each window 20 packets are transmitted. Here, on average about 7 packets are transmitted across the low-energy path and 13 across the low-latency path, for a window of 20 packets.

We looked at the probabilities that a packet sent over the low energy path would arrive at the sink on-time in Figure 8. On average, the probability that the packets sent over the low energy path would be on time is about 40%. This is because of the low transmission speed of the Bluetooth channel which results in increased retransmissions and consequently, higher delays. However, as we will show in the following paragraph, DETOUR responds to the probabilities and so the overall result is satisfactory.

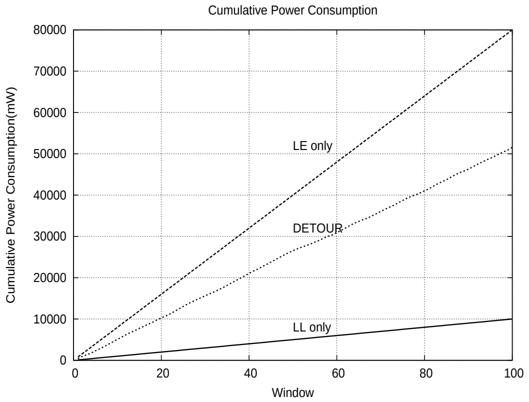

Fig. 11. Energy consumption of DETOUR for a deadline of *30ms* and an (m, k) constraint of $(15, 20)$.

Figures 9 and 10 displays the total number of packets received on time per window for (m, k) constraints of (10,20) and (15,20) respectively. In Figure 9, the total number of packets greater than or equal to the target number (10) is 98%. In Figure 10 the percentage is even higher with a 99% success rate. A look at Figure 7 indicates that DETOUR reacted as expected to this fall below the $(15, 20)$ constraint in window 60 by decreasing the number of packets on the low energy path from 6 to 5 in window 61.

Figure 11 compares the energy overheads for three scenarios - low-latency path only (LL-only), low-energy path only (LE-only) and using the DETOUR protocol for an (m, k) constraint of (15,20). We assume that the energy is directly proportional to the transmit power level used to transmit each packet. This is a course estimation since we ignore factors such as link attenuation, power from packet reception and power fluctuations due to interference. The middle line indicates the cumulative energy consumption for the scenario shown in Figure 1 with a packet deadline of 30ms. It can be seen that there is up to 35% energy savings when path diversification is used instead of shortest-delay only routing.

VI. SUMMARY AND FUTURE WORK

This paper introduced DETOUR, a path diversifying routing protocol that aims to satisfy multiple constraints. The dynamic nature of wireless networks and communications demand an adaptive approach to trade off multiple qualities. Our measurements indicate that, even though network conditions change, they do so by no means in a random way.

The focus in this paper is on firm real-time streams and a cost associated with multi-hop routing, specifically focusing on energy consumption. The (m, k) constraint has been used to steer the path diversification across multiple paths, focusing on real-time communication (meeting m deadlines in k packets) and low-energy communication. The results described in this paper indicate that the proposed approach can reduce energy consumption for streaming in wireless ad hoc and sensor networks while meeting the deadlines for a subset of the transmitted packets.

In our next step we will deploy these algorithms on a heterogeneous sensor network of camera-equipped sensing devices, including PDAs, Stargate devices, and others. These systems have different interfaces (IEEE 802.11, Bluetooth, Infrared, etc.), with different power characteristics and ranges. We will also explore the problem of transmission loss and how it can be incorporated into DETOUR. Finally, while the focus in this paper was on real-time communication and energy management, our future work will address other constraints, such as QoS or reliability.

REFERENCES

[1] S. Agarwal, R. H. Katz, and A. D. Joseph. Reducing the energy consumption of group driven ad-hoc wireless communication. Technical Report UCB/CSD-1-1127, UCB Computer Science Division (EECS), January 2001.

[2] P. Bergamo, D. Maniezzo, A. Travasoni, A. Giovanardi, G. Mazzini, and M. Zorzi. Distributed power control for energy efficient routing in ad hoc networks. *Wireless Networks*, 10(1), 2004.

[3] S. Bouckaert, J. Bergs, and D. Naudts. A mobile crisis management system for emergency services: from concept to field test. In *Proc. of QShine '06*, 2006.

[4] P. Djukic and S. Valaee. Minimum energy reliable multipath ad hoc networks. In *Proc. of the 22nd Biennal Symposium in Communications*, June 2004.

[5] Arjan Durresi, Vamsi Paruchuri, and Leonard Barolli. Delay-energy aware routing protocol for sensor and actor networks. *ICPADS*, 1:292 – 298, 2005.

[6] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[7] C. Papadopoulos and G. M. Parulkar. Retransmission-based error control for continuous media applications. In *Proc. of 6th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 1996.

[8] G. Quan and X. Hu. Enhanced fixed-priority scheduling with (m, k)-firm guarantee. In *Proc. of the 21st IEEE Real-Time Systems Symposium. Orlando, Florida*, November 2000.

[9] M. W. Subbarao. Mobile ad hoc data networks for emergency preparedness telecommunications - dynamic power-conscious routing concepts. Submitted as an interim project report for Contract Number DNCR086200 to the National Communications Systems, 2000.

[10] R. West, Y. Zhang, K. Schwan, and C. Poellabauer. Dynamic window-constrained scheduling of real-time streams in media servers. *IEEE Transactions on Computers*, 53(6):744 – 759, June 2004.