Energy-efficient internetworking with DTN

Dimitris Vardalis[1] and Vassilis Tsaoussidis[2]

[1]Space Internetworks Ltd, 12 Rethymnou, 10682 Athens, Greece
[2]Space Internetworking Center, Dept Of Electrical and Computer Engineering,
Demokritos University of Thrace, Xanthi 67100, Greece

Abstract—**We claim that Delay-Tolerant Networking has the potential to form an internetworking overlay that shapes traffic in a manner that exploits the capacity of last hop wireless channels and allows for energy-efficient internetworking. We demonstrate DTN potential for energy-efficient internetworking through an overlay-architecture and a tool we have developed that captures the required state transitions of the mobile device WNIC. We propose a novel rendezvous mechanism and show experimentally that the DTN overlay can re-shape traffic in a manner that allows the receiver to exploit the energy-throughput tradeoff better: it may condense sporadic packets into a burst and, in return, prolong sleep duration without risk to miss incoming packets and without degradation in throughput.**

Index Terms—**DTN, rendezvous mechanism, energy efficiency, internetworking.**

I. INTRODUCTION

Mobile devices became a useful every-day tool for communication, storage and entertainment. They are equipped with increasingly more powerful CPUs, larger memory capacities, faster wireless network adapters, and a set of applications that require internetworking capabilities. However, advances in battery technology have not been able to match the increased energy demand.

So far, the main focus of attention was the energy-efficient administration of device components, the selection of routing with energy capacity to route packets, or the capacity of battery itself. The (inter-) network *per se* has never attested its vital role in energy efficiency. The Internet, due to its probabilistic nature, constitutes a major reason for the lack of end-device energy control. Its scheduling uncertainty forces a receiver on continuously waiting for random contacts and, inevitably, enforces an energy-wasting policy; as a result, the device cannot incorporate energy-efficient modes of operation, such as *sleep*, into its transition state diagram. Therefore, a main reason for the inability to optimize end-device operations is the inherent inability of the internetwork to deliver data tactically, due to its store-and-forward and statistical multiplexing characteristics. Practically, what imposes the limitation to schedule transmission with accurate precision is the variation of delay, along with the potential lack of sufficient resources to store packets in transit, albeit having such a capacity could have, instead, permitted the network to shape traffic and corroborate a scheduling policy among the devices *and* the network.

In this work, we extend our previous work done in [1] and attempt to further demonstrate the potential of Delay Tolerant Networking (DTN) ([2], [3]) to shape internetwork traffic in a manner that allows mobile devices to balance their energy expenditure with minimal cost on throughput. In this context, we design and deploy an internetworking overlay that exploits two major DTN properties: (i) to store packets for as long as it is necessary, regardless of disruptions in communications; and (ii) to enhance the edge nodes with functionality to wait for sufficient amount of packets to arrive, prior to transmitting to the end node. We then develop scenarios and weigh our expectations using a broad set of experiments that reflect potential variations in traffic characteristics, error probabilities and disruptions.

In order to evaluate the proposed scheme we use the ns-2 network simulator. Since there is no established and reliable DTN implementation for ns-2 we opted for emulating the behavior of the bundle protocol by introducing a proxy application at the Base Station (BS). The energy expenditure model of ns-2 was also not sufficient for this work, since the whole state transition diagram for the Wireless Network Interface Cart (WNIC) had to be recorded. Hence, energy expenditure calculations involved hacking into the physical layer of the wireless node and adding tracing for the state transitions. Additionally, we propose a novel *rendezvous* mechanism, deployed between the BS and the mobile host, which takes advantage of the traffic shaping capabilities of DTN. Based on the incoming data rate the BS sets a rendezvous with the mobile host at a future time. In the meantime, the mobile host can switch its WNIC to sleep mode achieving energy conservation. Our results show that (i) energy expenditure can be reduced drastically and (ii) clearly, DTN appears as an appropriate tool to construct such overlay: it allows for permanent storage and, in addition, it does not require full, but instead, partial deployment on top of IP; in fact, even a minimal deployment of DTN nodes can satisfy the architectural requirements of energy-saving overlay as soon as the edge nodes deploy such service.

The rest of the paper is organized as follows: In section 2 we discuss related work, focusing on more recent approaches to save energy on mobile devices. In section 3 we discuss our proposal and in section 4 we detail our experimental tools and methodology, including parameters for evaluation, metrics and scenarios developed. In section 5 we discuss the results and, finally, in section 6 we summarize our conclusions.

II. RELATED WORK

The research literature on energy efficiency for mobile devices includes two broad categories that correspond to the device optimization and communication optimization. In [4], the authors identify as the main sources of energy expenditure the various subsystems of a handheld device, such as: the processor, the memory, the display, the audio system and the wireless networking. For network intensive applications, the power consumption of the WNIC can reach up to 60% of the total power necessary for the operation of the mobile device [5]. In [6] Jones et al. provide a comprehensive survey on the design principles of efficient network protocols. The authors focus on the specific mechanisms that can be employed in each of the layers in the network protocol stack (i.e. Physical, Data Link, Network, Transport, OS/Middleware and Application).

In [7], Adams et al. propose a buffering technique that exploits the inherent power-saving mode of 802.11. Data destined to a certain mobile node is hidden and thus, not made available to it for multiple beacon intervals. Mamatas and Tsaoussidis in [8] monitor the state transition of the WNIC and introduce two metrics for energy potential and unexploited available resource, to capture the energy performance potential of various protocols. The authors in [9] present a theoretical and empirical study of the impact of packet transmission interval and burst length on observed network performance characteristics, with or without use of a proxy service. They propose an adaptive, energy-saving streaming mechanism that adjusts the burst length. By the same token, however now focusing generally on TCP traffic, Anastasi et al. [10] introduce at the BS a proxy service that employs regular TCP on the wired network side and the novel Power-Saving Transmission Protocol on the wireless side. Authors in [11] expand on the proxy idea by introducing a scheduler service at the BS and a proxy at the mobile terminal. The scheduler incorporates a novel algorithm, called priority-based bulk scheduling (PBS). In [12] the proxy idea is again implemented at the BS introducing the novel RT_PS protocol for the communication between the BS and the mobile node. The RT_PS protocol includes *idle* intervals information so that the client can put the WNIC into *sleep* mode when applicable. By the same token, Batsiolas and Nikolaidis [13] propose a regulation of ACKs for TCP in a manner that allows the sender to predict with some accuracy when ACKs are expected; and meanwhile the sender enters a sleep mode.

III. ENERGY-EFFICIENT INTERNETWORKING OVERLAY

We depart from the mobile device *modus operandi*: at any given moment, a WNIC can be in one of five operation states, each with different power requirements: *transmit*, *receive*, *idle*, *sleep* (or doze) and *power-off*. The transmit, receive and idle states of the WNIC constitute the active states, whereas the sleep and power-off constitute the inactive states. The power requirements of active states are comparable with each other [4]. In the transmit state maximum power is consumed, while in the receive state slightly more power is required than in the idle state. In the sleep state the energy expenditure is an order of magnitude less than that of the idle state, and finally, in the power-off state no power is required. The IEEE 802.11 [14] protocol provides an inherent mechanism for allowing client devices to manage the power requirements of their WNICs. The mechanism relies on the central role played by the Access Point when the network is in the PCF (Point Coordination Function) operation mode (in this paper we do not consider the ad-hoc or Distributed Coordination Function of 802.11).

It becomes apparent that the challenge in energy-saving strategies now translates into how to maximize the duration that the WNIC spends in either the sleep or the power-off state. It is obvious, however, that the BS has limited buffer capacity and therefore, the scheduling precision and flow control of the BS determine the efficiency of each strategy. Otherwise, the risk of entering a sleep mode may lead to costly retransmissions and throughput degradation due to buffer overflow or receiver inactivity. Data buffering essentially shapes traffic in bursts so that periods of inactivity are prolonged, allowing the WNIC to stay in a low-power state longer, while at the same time minimizing the performance trade-off with throughput. Hence, it makes sense to design an overlay that buffers at least a Delay X Bandwidth product (DxB) between the edge node and the receiver (prior to transmitting) and distributes data storage among nodes in a manner that packet dropping is avoided. This sort of sophistication has not been developed or analyzed in our present work. We construct a rather demonstrative overlay where buffers have enough space; the disruptions never go beyond the storage capacity, so the possibility to cause buffer overflow is cancelled, in practice. In this context, we evaluate only the ability of the overlay to shape traffic in a manner that optimizes the transitions of the mobile device; and we evaluate the characteristics of the tradeoff with throughput.

In our first piece of work on exploring the use of DTN towards achieving energy conservation [1] we reported on the potential of traffic shaping had an appropriate mechanism to exploit this potential been present. Expanding on this idea, we propose a *rendezvous* mechanism that allows for taking advantage of this potential and that can be applied in a real-world situation. The main idea of the rendezvous mechanism is that when the BS flushes the buffer to the wireless receiver it also communicates the interval until the next transmission (i.e. next buffer flush), thus allowing the receiver to switch its WNIC to sleep mode in the meantime. The time for the next rendezvous is calculated at the BS based on the amount of data (sent by the wired sender) that has been accumulated during the last buffering period. A detailed layout of the rendezvous mechanism is presented in the next section.

The pseudo-code of the algorithm at an arbitrary DTN node of the overlay is as follows:

```
ReceiveIncomingData
If (HasConnectivityToNextHop) ForwardData
Else
{
   If (BufferAvailable) StoreData
   Else RejectDataCustody
}
Go Back to the Beginning
```

The pseudo-code of the algorithm at the DTN node running on the BS is as follows:

```
ReceiveIncomingData
If (BufferAvailable) StoreData
Else RejectDataCustody
RendezvousTimeElapsed
{
   If (HasConnectivityToMobileEndNode)
      FlushData
      CalculateNextRendezvousTime
      CommunicateNextRendezvousTimeToMobileNode
      RescheduleRendezvous
}
Go Back to the Beginning
```

What we also consider as novel in our proposal is the ability of the edge node (i.e. the last DTN node) to make a decision locally, with better precision than the sender of a typical end-to-end application. The architecture has the potential to exploit further the custody property of the DTN protocol, which leads to an automated balancing of storage as a result of the operation to shift data towards the destination only when the accepting nodes do have storage availability.

IV. EXPERIMENTAL METHODOLOGY

The bundle protocol behavior is emulated by introducing a proxy application at the BS. Measuring the energy expenditure was facilitated by modifying the physical layer in ns-2 so that it logs the state changes of the WNIC. The proxy application is connected to one input TCP agent, receiving data from the sender on the wired part of the network, and one output TCP agent, transmitting data to the receiver over the last hop wireless link. The proxy application buffers data received on the incoming end and flushes it out to the outgoing end as soon as the next rendezvous time with the receiver has been reached. This setup emulates the functionality of a DTN overlay on an IP network, when TCP is used as the convergence layer.

The proxy application was implemented as a subclass of the Application class in ns-2, sitting on top of the transport agents on the network stack. Additional class members, necessary to emulate the DTN node, included an agent that receives data from the source on the wired network and an option for the target buffer occupancy. The target buffer occupancy corresponds to the desired amount of accumulated data that will be flushed at the next rendezvous. Ideally, the amount of data flushed at every rendezvous would be equal to the target buffer occupancy; however, since the incoming data rate is unpredictable, fluctuations do occur.

Energy conservation is achieved by allowing the receiver to switch its WNIC to sleep mode until the next chunk of data is transmitted by the BS. At every rendezvous the BS calculates the time to the next rendezvous and communicates this information along with the flushed data to the receiver. The receiver then decides whether there is enough time for it to switch to sleep mode in the time between the end of the reception and the next rendezvous. If that is the case, it switches the WNIC to sleep mode and awakes it in time for the next chunk of data to be received, otherwise it remains in idle mode waiting for the next set of data to arrive. The time for the next rendezvous is calculated based on the smoothed target buffer occupancy portion received at the BS according to the following formulas:

```
bufferPortion = receivedData/targetBufferOccupancy

smoothedBufferPortion =
bufferPortion - (bufferPortion - 1)/2

nextRendezvous =
previousRendezvous/smoothedBufferPortion
```

In case the `smoothedBufferPortion` is 0 (no data was collected over the previous idle period) the next rendezvous is set as:

```
  nextRendezvous = previousRendezvous * 2
```

The general idea is that the next rendezvous interval is calculated based on the ratio of the amount of data received during the previous rendezvous interval over the target buffer occupancy. If the received data exceeds the target buffer occupancy the next rendezvous interval will be shorter than the previous rendezvous interval and vice versa. The introduction of a smoothing factor limits the fluctuations so that convergence is sooner achieved. The buffer portion is smoothed to a value halfway through the buffer portion itself and the unit (i.e. 1.3 becomes 1.15, 0.7 becomes 0.85). In future work the algorithm could be improved so that it takes into account historical buffer portion data, and thus, achieve convergence more promptly. A two-round example using a 30KB target buffer occupancy follows:

```
  1st rendezvous (0.5 sec, 45KB of received data):
bufferPortion = 45/30 = 1.5, smoothedBufferPortion
= 1.5 - 0.5/2 = 1.25, nextRendezvous = 0.5 / 1.25
= 0.4 sec

  2nd rendezvous (0.4 sec, 25KB of received data):
bufferPortion      =      25/30      =      0.83,
smoothedBufferPortion = 0.83 + 0.17/2 = 0.915,
nextRendezvous = 0.4 / 0.915 = 0.44 sec
```

In order to facilitate the energy expenditure calculations we modified the physical layer of the wireless node. Code that traces the duration the WNIC spends in each of the three states (i.e. send, receive and idle) was added. By having an accurate picture of state transitions we are able to carry out post simulation calculations and find the potential energy savings by employing the rendezvous mechanism that would switch the WNIC into the sleep state during idle intervals. In this paper we aim at emphasizing the energy saving potential that a DTN-based protocol can have for mobile devices. Addressing various optimization aspects, including the inherent load balancing capability of DTN custody transfer, is part of our future work plans.

A. Energy Expenditure Calculations and Comparison

During the post-simulation analysis a multi-purpose script identifies the energy-related log entries in the ns-2 trace file and calculates the energy expenditure based on certain input parameters. The script takes the following input parameters: transmit power (`txPower`), receive power (`rxPower`), idle power (`idlePower`), sleep power (`sleepPower`), transition power (`transPower`) and transition time (`transTime`).

The energy expenditure for each of the transmit and receive intervals is calculated as the duration of the interval multiplied by the transmit and receive power respectively. Idle intervals can either remain idle intervals (no switch to sleep mode occurs) or be converted to sleep intervals in order to achieve energy conservation. For idle intervals that are not converted to sleep intervals the energy expenditure is calculated as the duration of the interval multiplied by the idle power. For sleep intervals the energy expenditure is calculated as:

```
  2 * transPower * transTime + (idleDuration - 2 *
transTime) * sleepPower
```

Idle intervals candidate for conversion are those that occur in the meantime between the completion of a buffer flushing and the commencement of the next buffer flushing, and are identified by marking the end of each buffer transfer in the log files. The end of each buffer transfer is marked by having the TCP agent report upon the reception of the last acknowledgment on the BS side. During a candidate idle interval, a transition to the sleep state is considered feasible only if the time it takes to fall into the sleep state and then back to the idle state (i.e. `transTime`) is less than the duration of the idle interval itself. The transition is considered meaningful if the energy required for switching back and forth to the sleep state, plus the energy spent during the sleep state is less than the energy expenditure if the WNIC had remained in the idle state for the whole interval duration. The decision is based on the following formula:

```
 2 * transPower * transTime + (idleDuration - 2 *
transTime) * sleepPower < idleDuration * idlePower
```

It becomes obvious that the longer the duration of the idle intervals the greatest the opportunity for conserving energy.

The power figures for the various WNIC states are set as follows [15]: `txPower` = 1.400 Watts, `rxPower` = 0.950 Watts, `idlePower` = 0.805 Watts and `sleepPower` = 0.060 Watts. Based on measurements in [17] we can safely consider the energy expenditure in the transitive state to be the same as that of the idle state (i.e. `transPower` = `idlePower` = 0.805 Watts) so all feasible switches to the sleep state turn out to be meaningful switches as well. As reported in [16], transition time from sleep to idle and vice versa is in the order of tens of milliseconds. However, when switching back to the active state, the controller needs to wait for the next beacon in order to start receiving the buffered data. If we assume that it takes 50ms to switch states and another 100ms for the next beacon signal to be broadcast (a rather pessimistic estimation since 100ms is the default beacon interval for 802.11) the total for the transition amounts to 200ms. Therefore, we set the transition delay at 100 ms for both state transitions.

B. Model Description

Error! Reference source not found. depicts the network topology that was implemented in order to evaluate the energy conservation that could be achieved by employing a DTN-like protocol in *wired-cum-wireless* scenarios. Network nodes are named as N1 – N6. N4 is the BS node that is connected to both the wired and the wireless networks and node N6 is the wireless receiver. Links L13, L23, L34 and L45 are wired, while WL is the wireless link between the BS and the receiver.

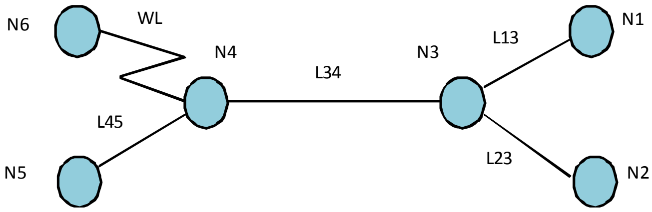

Fig. 1. Network Topology.

The data transfer studied in this experiment follows the N1→N3→N4→N6 route, while the competing flow follows the N2→N3→N4→N5 route. The bandwidth of the backbone link (L34) is changed so that the desired congestion is introduced into the network. The rest of the links are kept at the same delay and bandwidth values for all the experiments.

The bandwidth and delay values for all the links are as follows: L13 - 2Mb, 100ms, L23 - 3Mb, 100ms, L34 – 300ms delay and varying bandwidth depending on the desired network characteristic, L45 - 3Mb, 100ms, WL - 802.11 with a data rate of 11Mb and a basic rate of 1Mb. Stochastic Fair Queuing is applied on all queues, the file transfer size is 10MB and, when present, the competing flow sends a 1000 byte packet every 3 ms. TCP packet size is 1460 and maximum window size 100 packets..

In the *End-to-End* (E2E) scenarios there is an end-to-end connection between N1 and N6, whereas in the *Split* scenarios there is a splitting application that buffers data at the BS. The splitting application acts as a transport layer mediator and, as previously mentioned, plays the role of the DTN Bundle protocol. In our groundwork [1] we had carried out experiments that, for the most part, compared the energy saving potential (i.e. the energy expenditure had all the energy saving potential been exploited) of the E2E vs the Split cases by applying the same potential calculation algorithm to both strategies. In the current set of experiments, no energy saving mechanism is employed in the E2E case, while in the Split case the rendezvous mechanism, as described in previous sections is used. In certain experiment sets we also report on energy saving potential values taken from our previous work and label them as P-E2E and P-Split. The resulting energy expenditure values for all cases are reported in Joule.

In order to measure the delay of the wireless connection between the BS and the mobile receiver we employed a ping agent. The delay was found to be around 2ms. The DxB is thus: 0.002 x 11 000 000 / 8 = 2.750 KB. Except otherwise noted, the splitting application uses a target buffer occupancy of 30,000 B (around 10 x the DxB) for the calculation of the rendezvous times. In case the wireless connection is the bottleneck and data piles up at the BS we make the assumption that there is no limit in the available memory used for buffering data and, therefore, no data is dropped.

Finally, experiments are carried out both for an FTP connection transferring 10 MB of data as well as CBR traffic that flows for a specified amount of time. In certain experiments we are introducing a packet error rate on the wireless LAN. The error follows a uniform distribution and is applied on the outgoing interface of both the BS interface as well as the WNIC of the mobile node.

V. RESULTS

In this section we present the experimental results, organized in five subsections of experimental setups. Due to limited space we only present a subset of the available results. A full version of the original results reporting on the energy saving potential can be found in [18]. The setups presented in this paper are as follows: FTP and varying wired bandwidth (with and without a competing flow), FTP and varying wireless error (with and without a competing flow), FTP and varying buffering threshold (with and without a competing flow), CBR and varying buffering threshold.

A. Varying Bandwidth FTP transfer

In this set of experiments the backbone link is assigned a constant delay of 300ms, while the bandwidth varies from 0.5Mb to 2.5Mb in steps of 0.5Mb. The total transfer duration (TABLE I) is virtually the same for both the E2E and the Split cases, and is greatly affected by the bandwidth of the backbone link because of network congestion. Since the upstream link (L13) has a 2Mb bandwidth, setting the backbone bandwidth to values greater than 2Mb does not affect simulation results (L13 becomes the bottleneck of the connection and the backbone is uderutilized). Because of this, the transfer duration is equal for bandwidth values of 2Mb and 2.5Mb.

TABLE I
FTP, VARYING BANDWIDTH, NO COMPETING FLOW
TRANSFER DURATION

L34 bandwidth (Mb)	Duration E2E (sec)	Duration Split (sec)
0.5	216.85	216.23
1	145.63	145.19
1.5	111.45	110.98
2	90.09	90.34
2.5	89.95	90.14

In terms of the energy efficiency, it is obvious from the chart of Fig. 2 (lines E2E and Split) that smaller bandwidth values for the bottleneck link lead to greater energy gains for the Split approach. In the extreme setting of the 0.5Mb bandwidth, the energy conservation achieved in the Split case is approximately 46% of the energy expenditure in the E2E case.

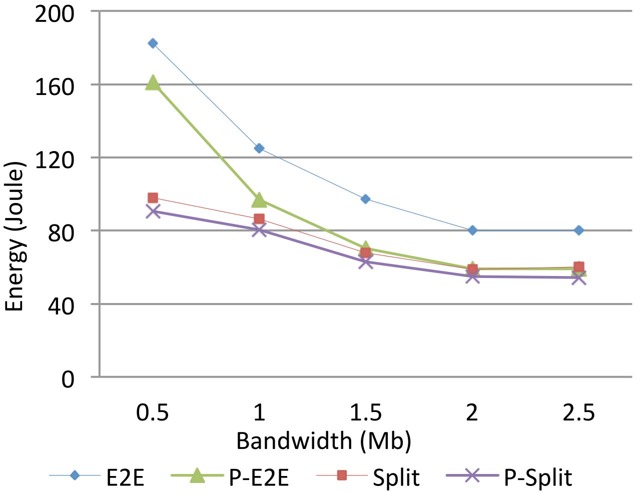

Fig. 2. FTP, Varying Bandwidth, No Competing Flow Energy Expenditure (E2E, Split) and Potential Energy Expenditure (P-E2E, P-Split).

As the network congestion eases, due to the increase of the bottleneck link capacity, the energy conservation becomes less dramatic, remaining however at a significant level. In the 2.0Mb and the 2.5Mb bandwidth cases it reaches the 20 Joule value, representing around 25% of the energy expenditure when no energy saving mechanism is employed. When severe network congestion is present the overall transmission duration in both cases is extremely high. In the E2E case, the mobile node at the receiving end is involved in this extra effort and has to remain in idle mode for a considerable amount of the transmission duration, waiting for incoming packets to arrive. Conversely, in the Split case the BS absorbs the effects of the network congestion (lost packets that lead to out-of-order data reception) and only transmits to the mobile host when consequent, buffer-sized chunks of data are available. In the intervals during which the BS accumulates incoming data the mobile node has the opportunity to switch its WNIC to sleep mode through the rendezvous mechanism, achieving great energy savings.

In DTN terms, the BS will be receiving the incoming bundle without engaging the receiving node in the process. Upon successful bundle reception, forwarding of the data to the mobile node would start. In the meantime, the WNIC of the mobile host can switch to sleep mode and save significant amounts of energy. The higher the congestion level the greater the energy gains achieved by the Split approach.

Fig. 2 contains two additional lines representing the potential energy expenditure of the E2E and the Split cases (labeled as P-E2E and P-Split), based on experimental data from our original work on the subject [1]. For a heavily congested network (a bottleneck bandwidth of 0.5 Mb) the E2E and P-E2E cases have comparable values. Reasonably, the E2E case exhibits the highest energy expenditure since in this case, no energy conserving mechanism is employed whatsoever. The P-E2E value for the same setting is 21 Joule smaller. This amount of energy could be potentially saved during long intervals of inactivity for the mobile host, when packets are dropped due to congestion and timeouts occur. When no congestion is present on the network the P-E2E case produces almost identical results as the Split case (>= 1.5 Mb), while the E2E case requires additionally 33% energy for the data transfer than the rest of the settings. For all bottleneck bandwidth values the Split case has slightly higher energy expenditure than the P-Split case, showing that the rendezvous mechanism is capable of taking advantage of almost all the opportunities there are for energy saving.

In the previous scenario we added cross traffic and rerun the experiments. In this set of experiments the maximum bandwidth value for the backbone link was increased to 3.5, since beyond that capacity the measurements remain unchanged.

TABLE II
FTP, VARYING BANDWIDTH, COMPETING FLOW
TRANSFER DURATION

L34 bandwidth (Mb)	Duration E2E (sec)	Duration Split (sec)
0.5	364.61379	358.45089
1	225.86923	194.862247
1.5	159.26399	158.83949
2	125.38937	124.685583
2.5	111.675838	111.188834
3	89.789484	89.442547
3.5	90.736727	90.324862

As expected, the transfer duration, shown in TABLE II, is significantly longer for both the E2E and the Split approaches when a competing flow is present. Nevertheless, unlike the previous scenario, small bandwidth values (i.e. 0.5 and 1 Mb) result to a slightly shorter transfer duration for the Split case.

Regarding the energy consumption, it can be observed in Fig. 3 (lines E2E and Split) that despite the significantly longer time it takes for the transfer to complete for the 0.5Mb setting (over 65% more in both cases), in the E2E approach the additional energy requirement is 65%, while in the Split approach the additional energy requirement is only around 13%. As the bandwidth increases and the network is relieved from congestion, the energy conserved by the Split approach is limited to approximately 17 Joule (a value that was observed in almost all congestion-free experiments). Regarding the potential energy expenditure the conclusions that can be drawn are roughly equivalent to that of the scenario when no competing flow is present. The only difference is that, because of the additional network congestion, at low bandwidth values it takes longer for the P-E2E energy expenditure values to reach the values of the Split and P-Split energy expenditure. Again, the main conclusion from this chart is that the rendezvous mechanism takes advantage of almost all energy conserving potential.

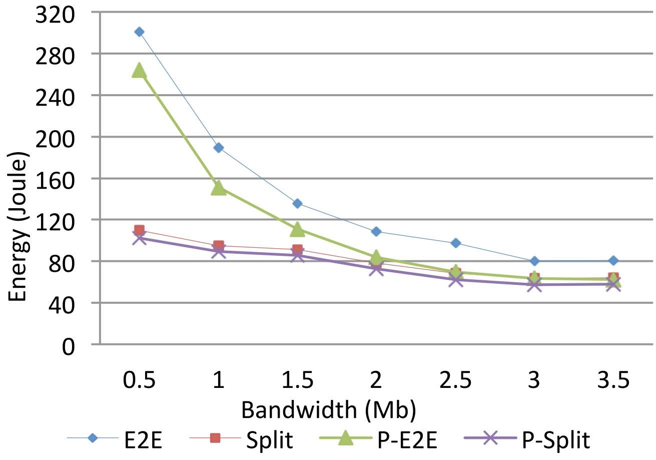

Fig. 3. FTP, Varying Bandwidth, Competing Flow Energy Expenditure (E2E, Split) and Potential Energy Expenditure (P-E2E, P-Split).

Fig. 4 depicts the state transitions for the first 20 indicative seconds of the file transfer in both the E2E and the Split cases when the backbone bandwidth is 2Mb and a competing flow is present. The *active* state includes sending, receiving and idle intervals (not adequately long for switching to sleep state), while the *potential sleep* state includes idle intervals long enough to accommodate transition to the sleep state. This chart emphasizes the increased potential for energy saving that the Split application creates irrespectively of whether a mechanism to take advantage this potential (such as the rendezvous mechanism) is in place.

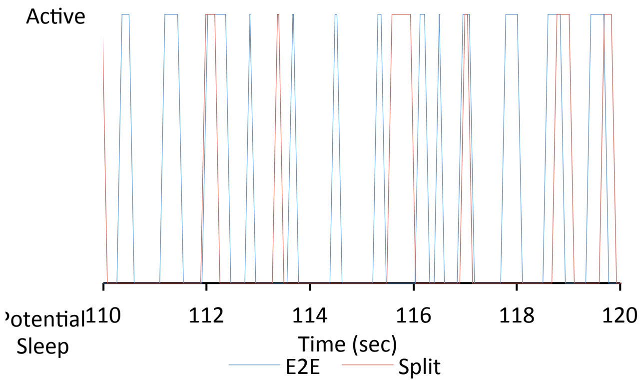

Fig. 4. FTP, 2Mb Bandwidth, Competing Flow State Transitions.

It becomes clear from the chart that in the E2E case, the WNIC needs to switch to active more frequently and that usually it needs to remain active for longer times than in the Split case. For example, between 110 and 114 seconds, in the E2E case the WNIC needs to switch to active 5 times, while in the Split case only 3 switches of significantly shorter duration are required.

B. Varying Wireless Error FTP transfer

The purpose of this scenario is to reveal the effect of packet error on the wireless part of the network both on the transfer duration as well as the energy efficiency of the E2E and the Split approaches. The backbone link was assigned a 300ms delay and a 1Mb bandwidth. The delay was selected so that it is considerably longer than the delays of the rest of the links in the topology and the bandwidth was selected so that the backbone is also the bottleneck link of the transfer route (i.e. network congestion is present).

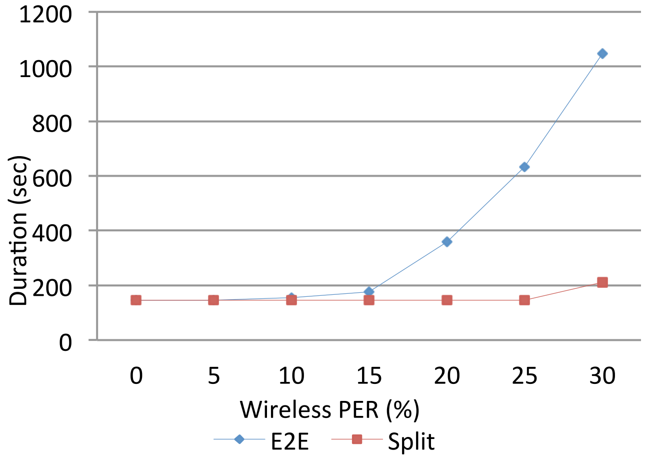

Fig. 5. FTP, Varying Wireless PER, No Competing Flow Duration.

At first glance, it may be striking that as the wireless packet error rate (PER) increases the duration of the E2E transfer increases as well (Fig. 5), reaching in the 30% case more than eight times the duration of the 0% case, whereas the duration of the Split case transfer remains almost the same across all runs. Part of the explanation to this, seemingly bizarre, result can be found in the Indirect-TCP solution proposed in [19]. In the E2E scenario, it takes very long for TCP to recover from packets lost due to the wireless error, because of the long RTT of the connection (around 800ms). On the contrary, the RTT of the wireless link is measured to be around 6-7 ms, and, therefore, a TCP connection between the BS and the mobile host can be very swift in recovering the lost packets. Additionally, since the RTT of the wired part of the route is very long, when compared to that of the wireless part, the time it takes for the BS to buffer the required data is enough for the TCP of the last hop to recover from any packet losses. This observation is not true only for the 30% PER case, in which it is evident that the last hop TCP cannot serve the incoming data, delaying the overall transfer time. In such a case, data could be dropped at the BS if the buffer capacity is depleted. We plan to include monitoring of packets dropped due to limited buffer capacity in future experiments. The enhanced performance achieved by a solution along the lines of Indirect-TCP (i.e. isolation of the wireless error in the short-delay part of the transfer route) comes naturally as an additional benefit when using DTN.

The energy consumption chart in Fig. 6 (lines E2E and Split) shows that up to an error rate of 10% the energy savings achieved by the Split approach are approximately 30%. As the error rate increases so does the conserved energy, reaching as much as 78% for an error rate of 30%. At this error rate the energy required for the completion of the data transfer in the E2E case is 861 Joule, while in the Split case the required energy is 186 Joule.

Fig. 6 contains two additional lines representing the potential energy expenditure of the E2E and the Split cases (labeled as P-E2E and P-Split), based on experimental data from our original work on the subject [1]. As expected, the P-E2E value (i.e. the value for the E2E case less the potential for switching to sleep mode in long idle intervals) is considerably lower than the E2E case, where no energy saving mechanism is employed, and higher than both the Split and P-Split values, where buffering at the BS takes place. Finally, the difference between the Split and the P-Split values falls between 7-10% for all error rates, showing that the rendezvous mechanism exploits 90-93% of the maximum theoretical energy saving amount. The observed difference consists of long idle intervals that cannot be used by the rendezvous mechanism for switching to sleep mode as they fall during data transmissions (i.e. as opposed to falling during the finish of a transfer and the next rendezvous). Nevertheless, the rendezvous mechanism takes advantage of the vast majority of the exploitable idle intervals, rendering it a very promising energy saving approach.

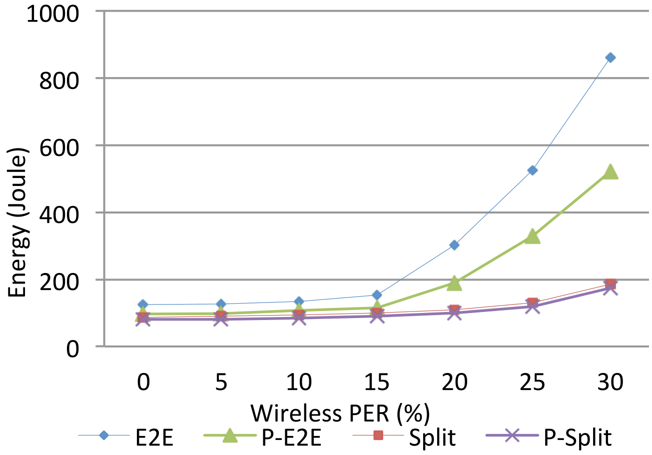

Fig. 6. FTP, Varying Wireless PER, No Competing Flow Energy Consumption.

The experiments in the previous section were repeated with a competing flow present on the network. Because of the huge transfer times that would otherwise be necessary, the bandwidth of the backbone link was set to 2Mb (as opposed to the 1Mb value that was used in the previous section). Even though we cannot directly compare the resulting figures for the cases with or without a competing flow, we can still draw meaningful conclusions for the relative performance of the E2E vs. the Split case. The most significant observation in this scenario as depicted in Fig. 7 (lines E2E and Split) is that when a competing flow is present, the energy expenditure of the E2E approach exhibits higher increase rates for smaller error rates. For higher error rates both the competing and the non-competing flow cases exhibit similar behavior. Comparing the Split and P-Split lines we observe that the difference is even smaller than in the previous set of experiments (when no competing flow is present), and therefore we can conclude that under heavily strained network conditions the rendezvous mechanism exploits virtually all of the available energy saving potential.

C. Varying Target Buffer Occupancy FTP Transfer

In this set of experimental runs the delay is set to 300ms, the bandwidth of the backbone link is set to 1Mb (rendering it as the bottleneck link of the transfer route) and the target buffer occupancy varies from 0 up to 180 KB. For the 0 buffering case we used the E2E approach, while for the rest of the runs we used the Split approach with the corresponding target buffer occupancy. The transfer duration is, reasonably, identical for all buffer sizes, and thus, not reported. The energy expenditure, however, drops significantly as the target buffer occupancy increases (Fig. 8, No-CF line). Since the wireless link of the transfer route is consistently underutilized, buffering more data allows the WNIC of the receiver to switch to a *sleep* state for longer periods, while at the same time not getting penalized for waiting. Increasing the target buffer occupancy beyond a certain value could cause congestion problems on the wireless LAN and/or prolong the transfer duration. Such problems could become more severe in case multiple mobile clients are present and download data using the Split approach. With the increase in the buffering capacity the traffic on the wireless LAN becomes burstier and consequently more prone to congestion. The size of the buffer, in case DTN is employed, can be thought of as the size of the bundle size that the BS would forward to the last hop of the transfer connection.

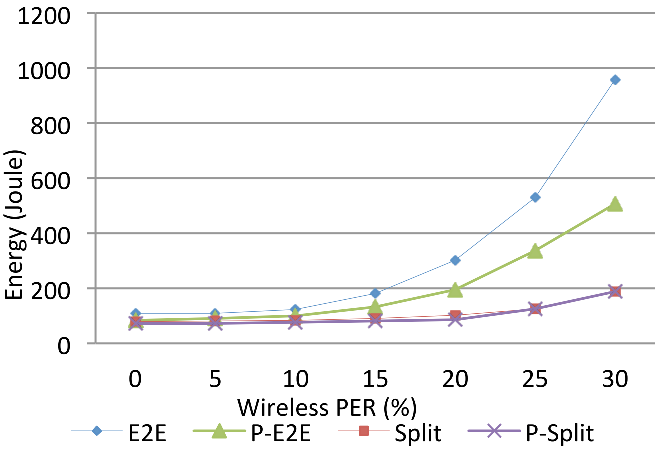

Fig. 7. FTP, Varying Wireless PER, Competing Flow, Energy Consumption.

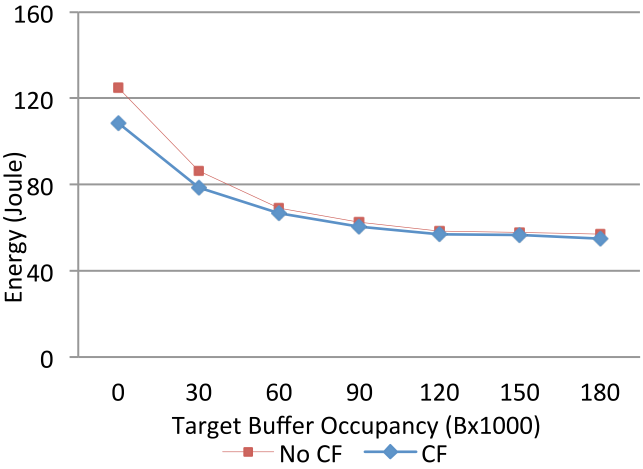

Fig. 8. FTP, Varying Target Buffer, FTP, With and Without a Competing Flow, Energy Consumption.

The experiments of the previous scenario were repeated with the introduction of cross traffic caused by a competing flow. In order to compensate for the higher network traffic the bandwidth of the backbone link was increased to 2Mb. In both sets of experiments the transfer duration remains constant for all runs. With respect to the energy expenditure, shown in Fig. 8 (CF line) it is observed that while for small buffer sizes the amount of energy required by the no competing flow scenario is significantly higher, as the target buffer occupancy increases the two values converge to a common Joule value in the mid 50s. This shows that increasing the buffer size can alleviate differences in the energy expenditure caused by varying congestion conditions in the wired part of the network.

TABLE III reports on the actual mean and max buffer occupancy values for each target occupancy value that was used in the experiments. Reporting on the actual buffer occupancy is of great significance in evaluating our approach because it reveals the effectiveness of the rendezvous mechanism in predicting the incoming data flow, and also allows the administrator to more accurately plan buffer allocation. The mean occupancy is almost always (with the exception of the 30 Bx1000 case) smaller than the target occupancy, ranging from 3% - 14%. As the target occupancy increases, so does the lag of the mean occupancy, showing that the mechanism exhibits slightly conservative behavior. The max occupancy starts at almost 10 times the mean occupancy for the 30 Bx1000 case and drops to less than double for values of 90 Bx1000 and higher. Even though the max occupancy value can be quite high, when multiple mobile nodes are present the collective buffer occupancy at the BS will be highly predictable due to the statistical characteristics of the buffering process.

TABLE III
NO COMPETING FLOW,
BUFFER OCCUPANCY VALUES (BX1000)

Target	Mean	Max
30	30.70	125.96
60	57.98	131.40
90	84.30	147.46
120	109.15	186.88
150	132.67	219.00
180	153.99	271.56

In order to better grasp the workings of the rendezvous mechanism we present in Fig. 9 the fluctuations of the buffer occupancy throughout the data transfer for the 60 Bx1000 target buffer occupancy case. The chart can be roughly split into two areas: 0-60 and 60-120 seconds. In the beginning of the transfer TCP enters the slow start phase in order to probe for available bandwidth and the same takes place at around 60 sec, in the latter case due to packet losses and timeouts. It becomes clear from the chart that the greatest buffer occupancy values occur during the slow start phase, in the beginning of each of the two areas. The reason for the intense fluctuations at around 0 and 60 seconds is that during the slow start phase, the connection data flow increases rapidly and the rendezvous mechanism takes a few seconds to adjust. However, for the rest of each of the two areas the actual buffer occupancy lies well with the 40 – 80 Bx1000 area, only 33% around the target buffer occupancy value. Again, this chart gives more evidence that in case of a multitude of mobile devices being serviced by the BS at the same time the buffer occupancy would be highly predictable, allowing for adequate buffer size allocation while avoiding underutilization. It also shows that there is room for making the rendezvous mechanism more intelligent so that it responds better to rapid data flow fluctuations, possibly by taking into account historical data.

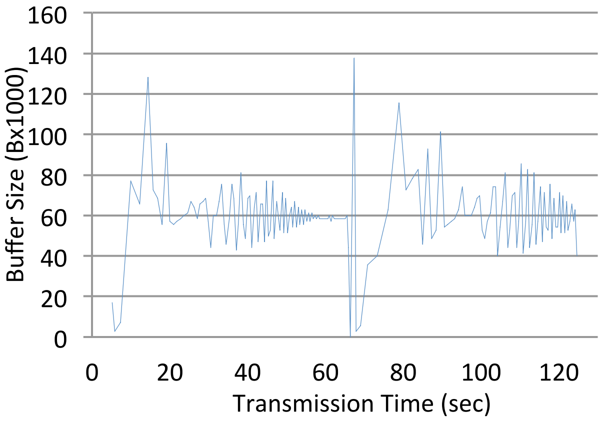

Fig. 9. Actual Buffer Occupancy at 60 Bx1000 Target Buffer Occupancy.

D. Varying Target Buffer Occupancy CBR Transfer

In this last set of experiments a CBR traffic flow of 10MB is considered in a network with no contending traffic. As in most of the previous scenarios the delay of the backbone link was set to 300ms and the bandwidth to 2Mb. The transfer duration does not convey any important information as it merely refers to the time the last packet is received by the mobile node and it depends solely on the end-to-end network delay.

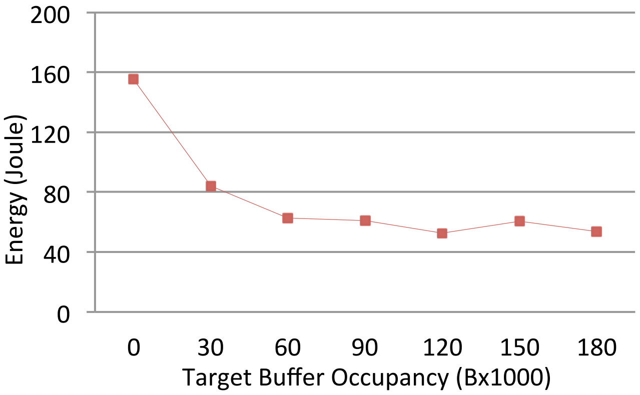

Fig. 10. CBR, Varying Buffering Threshold Energy Consumption.

This scenario can be mostly considered as a placeholder for future work, since the side effects of delaying CBR traffic are not taken into account. When measuring the service quality of a file transfer the time for completion can be an adequate metric. The sooner the file transfer finishes the better it is for the end user. In all the experiments, so far, the transfer duration was not affected by the buffering mechanism, rendering the mechanism acceptable from a user standpoint. However, CBR traffic is, in most cases, related to multimedia streaming. Extra delays introduced by the buffering mechanism can lead to unacceptable service quality, a factor which is not captured by our current experimental setup. Nevertheless, the experiments in this scenario reveal the potential a DTN-based Split approach can have for reducing the energy consumption for CBR traffic.

It becomes apparent from the chart in Fig. 10 that even for a buffer size of 30KB, the energy saving achieved by the rendezvous mechanism is 46% compared to the case where no power saving is used (0 target buffer occupancy). As the buffering size increases the conserved energy increases as well, reaching almost 66% (for a target buffer occupancy of

120 Bx1000) of the energy spent in the no-buffering case. For the current experimental settings it seems that setting the target occupancy to more than 60 Bx1000 does not significantly improve the energy efficiency, even though there is a general such trend along the x axis of the chart.

Fig. 11 depicts the actual buffer occupancy achieved by the rendezvous mechanism in the 60 Bx1000 target buffer occupancy case. As opposed to the corresponding chart for the TCP connection of the previous section, the actual buffer occupancy in this case is almost constant, since in a CBR connection the incoming data flow is constant as well. It should be noted that large buffering in multimedia streams can cause long delays and, thus, unacceptable user experience; however, if used with caution the rendezvous mechanism can deliver significant energy conservation with little or no quality degradation.

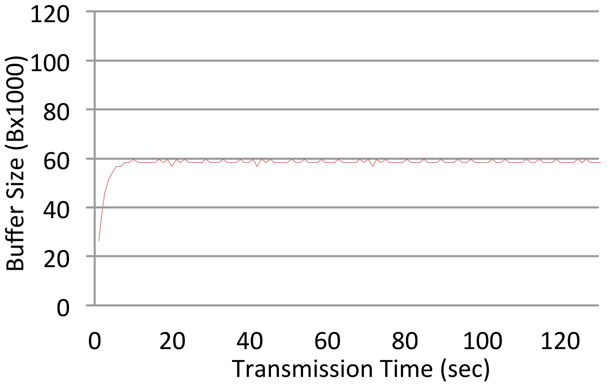

Fig. 11. Actual Buffer Occupancy at 60 Bx1000 Target Buffer Occupancy.

VI. Conclusions and Future Work

We evaluated the potential of DTN to shape Internet traffic in a manner that allows mobile devices to utilize energy-conserving modes of operation, without risk to lose packets or degrade their throughput. In order to take advantage of the traffic shaping capabilities of DTN we deployed a novel rendezvous mechanism between the BS and the mobile host. Our results are based on a post-simulation analysis of the recorded transition state diagram of the mobile device, as this was determined by the communication between the end-device and the edge network node. The most conclusive results of our experiments capture the huge gain in useless transitions and uncover the energy potential of DTN properties. Additionally, the results indicate quite clearly that the rendezvous mechanism, however simple it is, exploits most of the energy saving potential created by DTN.

We briefly discussed the property of DTN to balance network traffic as well. This is not peripheral to energy-saving strategies: Load balancing within the network, unlike typical end-to-end approaches, cancels the risk of packet dropping but also forwards information towards the destination as soon as possible. Application data, instead of waiting at the source, it is pushed at some network nodes closer to the destination. This latter becomes particularly important in environments, such as space for example, where a contact graph is predetermined and the penalty for losing a contact opportunity may delay communication significantly. Hence, load balancing strictly associates with less retransmissions and shortened communication time: two tactics with energy-conserving consequences. We plan to focus on this dimension of energy-saving strategies by extending our present evaluation framework with load balancing algorithms.

Finally, taming uncertainty allows for more sophisticated signaling techniques that deploy a hybrid Time Division and Statistical Multiplexing strategy. This sort of sophistication is under progress.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013_FP7-REGPOT-2010-1, SP4 Capacities, Coordination and Support Actions) under grant agreement n° 264226 (project title: Space Internetworking Center-SPICE). This paper reflects only the authors’ views and the Community is not liable for any use that may be made of the information contained therein.

REFERENCES

[1] D. Vardalis, V. Tsaoussidis, "Energy-efficient internetworking with DTN", Proceedings of 9th International Conference on Wired/Wireless Internet Communications, 2011.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-Tolerant Networking Architecture”, RFC 4838, IETF, 2007.

[3] K. Scott and S. Burleigh, “Bundle Protocol Specification”, RFC 5050, IETF, 2007.

[4] M. Viredaz, L. Brakmo, and W. Hamburgen, “Energy Management on Handheld Devices”, ACM Queue Magazine - Power Management, Volume 1 Issue 7, 2003.

[5] A. Acquaviva, T. Simunic, V. Deolalikar, and S. Roy, “Remote power control of wireless network interfaces”, Proceedings of International Workshop on Power and Timing Modeling, Optimization and Simulation, 2003.

[6] C. Jones, K. Sivalingam, P. Agrawal, J. Chen, “A Survey of Energy Efficient Network Protocols for Wireless Networks”, ACM Journal on Wireless Networks, Volume 7, Issue 4, 2001.

[7] J. Adams and G.M. Muntean, “Adaptive-Buffer Power Save Mechanism for Mobile Multimedia Streaming,” Proceedings of IEEE International Conference on Communications '07, 2007.

[8] L. Mamatas, V. Tsaoussidis, "Exploiting Energy-Saving Potential in Heterogeneous Networks", International Journal of Parallel, Emergent and Distributed Systems (IJPEDS), Taylor & Francis, volume 23, issue 4, 2008.

[9] J. Korhonen and Y. Wang, “Power-efficient streaming for mobile terminals”, Proceedings of ACM International Workshope on Network and Operating Systems Support for Digital Audio and Video, 2005.

[10] G. Anastasi, M. Conti, and W. Lapenna, “A Power-Saving Network Architecture for Accessing the Internet from Mobile Computers: Design, Implementation and Measurements ”, The Computer Journal, Volume 46, Issue 5, 2003.

[11] H. Zhu and G Cao, “A Power-Aware and QoS-Aware Service Model on Wireless Networks ”, Proceedings of IEEE INFOCOM '04, 2004.

[12] G. Anastasi, M. Conti, E. Gregori, A. Passarella, and L. Pelusi, “A Power-Aware Multimedia Streaming Protocol for Mobile Users”, Proceedings of the IEEE International Conference on Pervasive Services '05, 2005.

[13] I. Batsiolas and I. Nikolaidis, “Selective Idling: Experiments in Transport Layer Energy Conservation”, The Journal of Supercomputing, Volume 20, Number 2, 2001.

[14] IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements , Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications , 2007.

[15] Eugene Shih, Paramvir Bahl, and Michael J. Sinclair, “Wake on Wireless: An Event DrivenEnergy Saving Strategy for Battery Operated Devices”, Proceedings of the ACM International Conference on Mobile Computing and Networking, 2002.

[16] T. Simunic and S. Boyd, “Managing power consumption in networks on chips,” ACM DATE, 2002.

[17] Yu Xiao, Petri Savolainen, Arto Karppanen, Matti Siekkinen, and Antti Ylä-Jääski, “Practical power modeling of data transmission over 802.11g for wireless applications”, Proceedings of the ACM International Conference on Energy-Efficient Computing and Networking, 2010.

[18] D. Vardalis and V. Tsaoussidis, “Energy-efficient internetworking with DTN”, Technical Report: TR-DUTH-EE-2011-4, http://www.intersys-lab.org/pages/publications.php

[19] A. Bakre, B. R. Badrinath, “Implementation and Performance Evaluation of Indirect TCP”, IEE Transaction on Computers, Vol. 6, No. 3, 1997

Dimitris Vardalis received his Bachelor degree in Computer Science from the Aristotle University of Thessaloniki, Greece. He holds a MSc degree in Computer Science from the State University of New York at Stony Brook, as well as an MBA from the University of Macedonia in Thessaloniki, Greece. Currently, he pursues a PhD degree at the Electrical Engineering Department of Democritus University of Thrace and works as a software engineer with Intelligen Inc. His research focus is networking and, more specifically, Delay Tolerant Networking and transport protocols for wireless networks.

Vassilis Tsaoussidis received a B.Sc. in Applied Mathematics from Aristotle University, Greece; a Diploma in Statistics and Computer Science from the Hellenic Institute of Statistics; and a Ph.D. in Computer Networks from Humboldt University, Berlin, Germany (1995). Vassilis held faculty positions in Rutgers University, New Brunswick, SUNY Stony Brook and Northeastern University, Boston. In May 2003, he joined the Department of Electrical and Computer Engineering of Democritus University, Greece. His research interests lie in the area of transport/network protocols, i.e., their design aspects and performance evaluation. Vassilis is editor in chief for the Journal of Internet Engineering and editor for the journals IEEE Transactions in Mobile Computing, Computer Networks, Wireless Communications and Mobile Computing, Mobile Multimedia and Parallel Emergent and Distributed Systems. He participated in several Technical Program Committees in his area of expertise, such as INFOCOM, NETWORKING, GLOBECOM, ICCCN, ISCC, EWCN, WLN, and several others.