Application-Driven Network Capacity Adaptation for Energy Efficient Ad-Hoc Networks

Yongho Seok, Nakjung Choi, Yanghee Choi
Seoul National University
Seoul, Korea
{yhseok, fomula, yhchoi}@mmlab.snu.ac.kr

Jihong Kim
Seoul National University
Seoul, Korea
jihong@davinci.snu.ac.kr

duce the energy consumption of mobile stations in order to increase their battery lifetime. Many related works, assuming the presence of a fixed, high traffic load, concentrate on reducing the energy consumed during ongoing packet transmission. However, even when the traffic load dynamically changes, it is important to improve the performance such as throughput and energy consumption. In this paper, we propose an energy saving mechanism which works well independently of the traffic load, even when the traffic load is low. We present an algorithm called the *network capacity adap-*

according to the traffic load. The proposed NCA algorithm can be used together with other algorithms in order to save energy. Experimental results show that the proposed algorithm reduces the energy consumption by 52%, 55%, and 37% when used in conjunction with the IEEE 802.11, IEEE 802.11 PSM, and SPAN, respectively.

I. Introduction

A mobile ad hoc network is a set of wireless mobile nodes forming a dynamic autonomous network through a fully mobile infrastructure. Nodes communicate with each other without the intervention of centralized access points or base stations. Since mobile stations usually operate under a limited energy source, such as battery, minimizing energy consumption is an important design requirement in mobile ad-hoc networks. Considerable progress has been made on developing a low-power communication protocol for mobile devices, whose NIC (Network Interface Card) still consumes energy continuously even when no communication is taking place.

IEEE 802.11 Power Saving Mode (PSM) [1] allows wireless hosts to periodically power off their wireless NIC, resulting in energy saving. In PSM, a centralized host sends a beacon message to neighboring mobile stations, requesting them to either turn on or off their NIC. Moreover, many protocols (e.g., PAMAS [3], PCMA [4] and PARO [5]) have been proposed in the past, which operate in the MAC and Network layers. However, these protocols are only effective when the traffic load of the network is very high. Because most studies in this area focused on the energy consumed per packet transmitted, the energy saving gain of the proposed power-aware communication protocols is inefficient when a majority of the mobile nodes have nothing to transmit over a long period of time. The [2] presents a series of

i t hi h bt i d d t il d t f th energy consumption of an IEEE 802.11 wireless network interface operating in an ad hoc networking environment. In an ad hoc mode, the idle power consumption is significant, as hosts must maintain their network interfaces in idle mode so as to cooperate in maintaining the ad hoc routing fabric. This means that a new power management approach is needed to complement the previously proposed protocols when most of mobile stations' lifetime is spent in idle mode.

Therefore, in this paper, we propose a novel technique for reducing the energy consumption in idle mode. Our proposed technique called the *network capacity adaptation* derives an effective network capacity required by a given application and *logically* slows down the network capacity when the traffic load is low. By slowing down the *logical network capacity*, the neighboring mobile stations can stay longer in the power-off mode, significantly saving the energy consumed in idle mode.

The remainder of this paper is organized as follows. Section II provides a brief overview of IEEE 802.11 DCF and the basic ideas behind our proposed algorithm is introduced in section III. Then, we present the overall concepts about network capacity adaptation, called NCA in section IV. Section V presents performance evaluation, such as power consumption, throughput and latency through the simulation by using an $ns2$ simulator. Finally, we summarize and conclude this paper in Section VI.

II. Overview of IEEE 802.11 DCF

We briefly review the IEEE 802.11 Distributed Coordinated Function (DCF) [1]. As described in [1], a transmitting mobile station must first sense an idle channel for a time period corresponding to the Distributed InterFrame Spacing (DIFS), after which it generates a random backoff timer chosen uniformly from the range $[0, w-1]$, where w is referred to as the contention window. At the first transmission attempt, w is set to the minimum contention window CW_{min}. After the backoff timer reaches 0, the mobile station transmits a short request-to-send (RTS) message. If successfully received, the receiving mobile station responds with a short clear-to-send (CTS) message. Any other mobile stations which hear either the RTS or CTS message

This work was supported in part by the Brain Korea 21 project of Ministry of Education and in part by the National Research Laboratory project of Ministry of Science and Technology, 2003, Korea.

0-7803-7822-9/03/$17.00 © 2003 IEEE.

uses the *duration*[1] field of this control message to update their Network Allocation Vector (NAV) containing information about the time period during which the channel will remain busy. Thus, all mobile stations, including hidden nodes, can defer transmission so as to avoid collisions. Finally, a binary exponential backoff scheme is used such that the value of w is $2^{n+5}-1$ (*retry counter* n=0,...,5), beginning with an initial value of 31, up to a maximum value 1023. Fig. 1 presents the sequence of packets transmitted with DCF in IEEE 802.11 ad hoc mode.

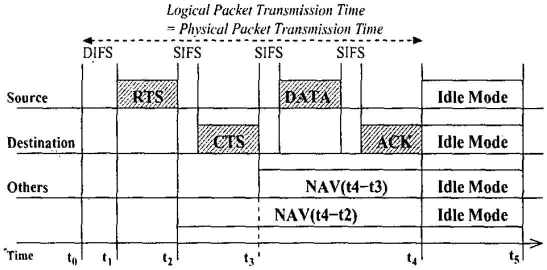


Fig. 1. Transmission Timeline in the original IEEE 802.11

III. Basic Idea

In this paper, the energy saving mechanism of mobile stations consists of two main parts. First, when active nodes (*Source* and *Destination* in Fig. 1) that transmit or receive the data packet start communication, those neighboring nodes (*Others* in Fig. 1), among the active nodes which overhear the RTS or CTS message, update their NAV. Here, while the NAV is busy, we can power off the network interfaces of *Others* nodes, so that these nodes do not start any other communication in accordance with the hidden terminal problem in IEEE 802.11 [1]. Secondly, after the data packet delivery is finished, the network interfaces of the other nodes are started in idle mode to send or receive other packets whenever this becomes necessary. If we decrease the idle time of the mobile station by increasing the *Logical Packet Transmission Time*, we have the more energy saving gain, since *Others* nodes will have a longer power-off time.

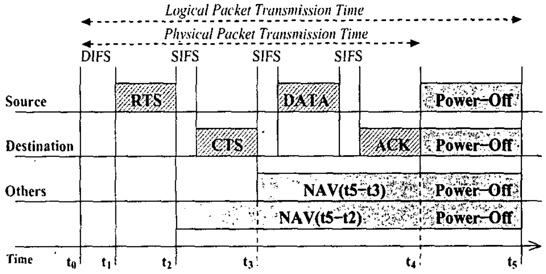
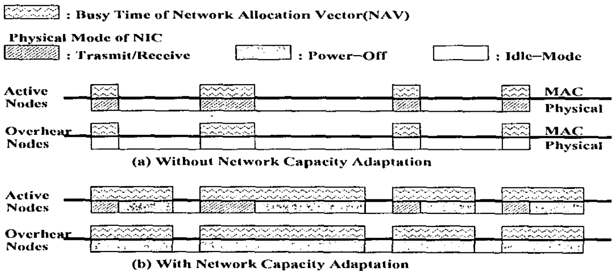


Fig. 2. Basic idea for energy saving

Fig. 2 shows the sequence of packets transmitted when the energy saving mechanism is applied. *Others* nodes that overhear the RTS and CTS power off their network interfaces when the NAV is busy instead of entering into idle mode in Fig. 1. Also, the time $(t_5 - t_4)$ spent in idle mode in Fig. 1 is changed into the time spent in power-off mode, so that *Logical Packet Transmission Time* of Fig. 2 is increased more than that of Fig. 1.

The NAV in the MAC layer is increased by updating the *duration* field of the MAC frame with a value corresponding to the *Logical Packet Transmission Time* in the MAC layer, but the practical *Physical Packet Transmission Time* in the physical layer is never changed by the energy saving mechanism. As you can see in Fig. 1 and Fig. 2, the time taken by a *Source* node to receive the ACK packet is the same in both cases. In practice, we do not change the data rate in the physical layer. This is because the transmission power of a network interface is set to a fixed value irrespective of the data rate, so a longer transmission time in the physical layer requires more power consumption.



(a) Without Network Capacity Adaptation

(b) With Network Capacity Adaptation

Fig. 3. Timeline showing the NIC states and NAV during multiple packet transmissions.

Fig. 3 shows the mode (e.g. Transmit/Receive, Idle and Power-off) of the NIC in the Physical layer and the NAV in the MAC layer when multiple packet transmissions are in progress. The major difference between Fig. 3 (a) and (b), which refer to situations in which the energy saving mechanism is or is not applied, respectively, is that both *active nodes* and *overhearing nodes* in Fig. 3 (b) have more power-off time than that in Fig. 3 (a). As we mentioned above, this energy saving gain is caused by increasing the NAV through the *logical capacity adaptation* in the MAC layer. However, the issue of how much capacity[2] is required in MAC layer is an important one. Meanwhile, the data rate in the MAC layer must be set to an optimal value, which allows for the power saving gain to be maximized, such that the idle time of the mobile stations may be decreased to a value near to zero. This data rate must be carefully selected since the throughput will be degraded due to congestion, if the data rate is less than the required capacity of the mobile stations. In the next section, we present the capacity estimation algorithm which is designed to resolve this problem.

[1] In our proposed scheme, the *duration* field plays a key role in reducing the energy consumption of mobile stations.

[2] In this paper, if not mentioned specifically, capacity means *logical capacity* defined as the capacity in the MAC layer.

IV. Network Capacity Adaptation

A. Architecture of Network Capacity Adaptation (NCA) Algorithm

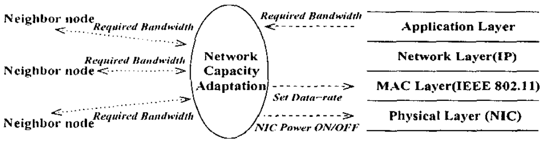


Fig. 4. Architecture of the network capacity adaptation.

The architecture of a mobile station supporting the proposed network capacity adaptation (NCA) algorithm is shown in Fig. 4. The NCA algorithm consists of both the energy saving mechanism presented in Section III and the protocol used to estimate the required capacity of each mobile station presented in Section IV-B, which is utilized for the Section III. An application layer provides the required bandwidth for the given session through the signaling process to the NCA component of the mobile stations followed the path from source to destination node. Each mobile station shares information related to its traffic load with neighboring nodes by means of a signaling protocol. It is important to carefully compute the required capacity since it effects the network performance. If the required capacity is under-estimated (*load* > *capacity*), the throughput of the network is decreased due to network congestion. If the required capacity is over-estimated (*load* ≪ *capacity*), the optimal energy saving gain cannot be achieved. With the information about the load, each mobile station determines the logical data rate to be used in order to calculate the period of communication. Then, the mobile station updates the *duration* field in subsequent control messages for communication in the MAC layer, such that irrelevant neighbor nodes can turn their NICs off to save power consumption.

B. Estimation of Required Capacity in Mobile Stations

TABLE I
Notations used in Section IV-B

Notation	Description
R	A set of all sessions in a MANET
N_i	A set of all neighbor nodes of $node_i$
$x_{ij}(r)$	Traffic load from $node_i$ to $node_j$ on a session r
L_{ij}	All traffic load flowed between $node_i$ and $node_j$
L_i	Traffic load of all nodes in the coverage of $node_i$
C_{ij}	Logical capacity between $node_i$ and $node_j$
C_i	Logical capacity of $node_i$
$MaxCapacity$	Maximum C_i for all nodes
IR_i	Ratio of idle time to total alive time
$MaxIdleRatio$	Maximum IR_i for all nodes
U_{ij}	Resource utility between $node_i$ and $node_j$
U_i	Resource utility of $node_i$

When the proposed NCA algorithm changes the network capacity according to traffic load, the network throughput should not be degraded. To satisfy this constraint, each mobile station must know its own traffic load so that it can estimate the minimum network capacity required to ensure that congestion is never caused. Estimating traffic load as accurately as possible is a key function of our algorithm.

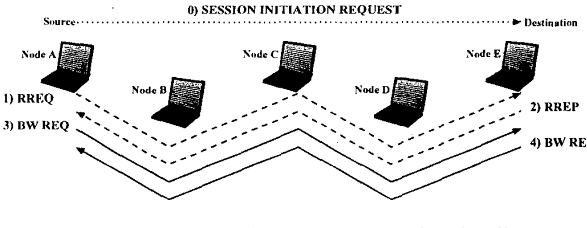


Fig. 5. Required bandwidth announcement using signaling protocol between a source and a destination.

As shown in Fig. 5, the session initiation process from Node A to Node E consists of looking up the routing path and reserving the resource required by the application. Fig. 5 illustrates the situation being considered. Once route setup is completed through RREQ-RREP, the next step is to announce the bandwidth to be used for data transmission. Using the above established route, Node A sends a BW-REQ to Node E for bandwidth confirmation which is acknowledged by a BW-RESV from Node E. During this negotiation process the intermediate nodes reserve the necessary bandwidth for these two peers' communication. We utilize INSIGNIA [6] which is a resource reservation protocol for a mobile ad hoc network extended from RSVP used in wired networks. By means of this signaling protocol, each mobile station obtains information about the amount of incoming and outgoing traffic for all sessions passing through itself. Based on this information, we define the traffic load (L_{ij}) between adjacent nodes ($node_i$ and $node_j$) and traffic load (L_i) of $node_i$ as follows:

$$L_{ij} = \sum_r \{x_{ij}(r) + x_{ji}(r)\}, \quad r \in R \qquad (1)$$

$$L_i = \sum_k L_{ik}, \qquad k \in N_i \qquad (2)$$

Here, the traffic load (L_i) of $node_i$ is represented as the sum of the incoming traffic from and the outgoing traffic to all neighboring nodes in the coverage area of $node_i$. This is because $node_i$ is affected by all nodes within a two-hop distance of itself.

The idle ratio of $node_i$ and $MaxIdleRatio$ can also be calculated using the above equations.

$$IR_i = 1 - \sum_k \frac{L_{ik}}{C_{ik}} \leq MaxIdleRatio, \quad k \in N_i, \forall i \quad (3)$$

IR_i represents the ratio of idle time to total alive time in $node_i$. $Eq.(3)$ is derived as follows.

$$IR_i = \frac{IdleTime_i}{NonIdleTime_i + IdleTime_i} = \frac{TotalTime_i - TotalTime_i \times \sum_k \frac{L_{ik}}{C_{ik}}}{TotalTime_i}, \quad where\ k \in N_i \qquad (4)$$

Given the above definitions, the objective of our proposed algorithm is

$$\text{Minimize} \quad MaxIdleRatio \qquad (5)$$

This is a min-max problem. Minimizing $MaxIdleRatio$ means to minimize maximum idle time for all nodes, thus resulting in more energy saving fairly. Of course, we can solve this problem by minimizing the sum of all nodes' idle time ($Minimize \sum IdleTime_i$). Although such a solution can provide minimum total idle time from the viewpoint of the whole network, it may make specific nodes consume more energy than others, resulting in such problems as network partitioning. Therefore, this problem should be solved as a min-max problem to keep the lifetime of all nodes as fair as possible. When we minimize $MaxIdleRatio$, we should consider the constraint that is needed to prevent each mobile station from being congested due to its load exceeding its capacity. The constraint is that the resource utility (U_{ij}) between $node_i$ and $node_j$ should be less than 1.

Subject to

$$U_{ij} = \sum_k \frac{L_{ik}}{C_{ik}} + \sum_{k'} \frac{L_{jk'}}{C_{jk'}} \leq 1$$
$$. \ where\ k \in N_i.\ k \ \in N_j.\ j \in N_i.\ \forall i \qquad (6)$$

When we consider the traffic load between two communicating nodes, we need to consider not only the individual traffic load between these two nodes, but also the traffic load between these two nodes and their neighboring nodes.

And since IR_i is defined as in $Eq.$(3), our objective, $Eq.$(5), can be translated as follows.

$$\text{Minimize } MaxCapacity \qquad (7)$$

$MaxCapacity$ is defined by means of an expression similar to $Eq.$(3).

$$C_i \leq MaxCapacity, \quad \forall i \qquad (8)$$

C_i is a variable that represents the capacity of $node_i$, which we want to determine.

In the real world, we need to know the states of all neighbors in order to evaluate $Eq.$(6). However, as the number of neighbors that a node has increases, the number of the states that it needs to keep track of also increases. This may cause memory wastage and excessive energy consumption. Therefore, $Eq.$(6) can be modified as follows, in order to define the resource utility of each mobile station (U_i), in an effort to reduce the information that a node needs to keep track of.

$$U_i = \frac{L_i}{C_i} + \frac{L_j}{C_j} \leq 1, \quad j \in N_i,\ \forall i \qquad (9)$$

Then, we can find the optimal solution with $Eq.$(8) and (9).

In order for the optimal solution to the above problem to be found, each mobile station needs to keep track of global information about all mobile stations. Because it is necessary to provide a scalable distributed network, control message communication must be localized into neighboring nodes. Therefore, we propose the following capacity estimation approximation algorithm.

$$C_i = L_i + \alpha \max\{L_k | k \in N_i\}, \quad \alpha \geq 1 \qquad (10)$$

As shown in $Eq.$(10), each mobile station only needs to keep track of the traffic load information of its neighbors, and not of all of the nodes. The disadvantage of *logical capacity adaptation* is that it can cause high end-to-end packet delay, due to the increased *Logical Packet Transmission Time* required to reduce power consumption. In the above equation, α is the *load sensitivity* that trades off the end-to-end packet delay and the power consumption. A large value of α reduces the end-to-end packet delay and a small value of α reduces both the idle time and the power consumption. Moreover, the capacity of each mobile station is subject to the constraint defined in $Eq.$(9), so that network throughput is not degraded. We present the next theorem required to verify the feasibility of this equation,

Theorem 1 Capacity of each mobile station obtained by $Eq.$(11) satisfies the constraint $Eq.$(10). Here, α is 1 for simplicity.

Proof) By the proposed algorithm, C_i *and*
C *can be set to*

$$C_i = L_i + L_j + \delta_i, \quad j \in N_i$$
$$C_j = L_j + L_i + \delta_j, \quad i \in N_j$$

. *where* $\delta_i \geq 0$ *and* $\delta_j \geq 0$. *respectively.*
Thus, we have

$$\frac{L_j}{C_j} + \frac{L_i}{C_i} = \frac{L_j}{L_j+L_i+\delta_j} + \frac{L_i}{L_i+L_j+\delta_i}$$
$$\leq \frac{L_j}{L_j+L_i} + \frac{L_i}{L_i+L_j} = 1 \qquad \square$$

V. Experimental Results

We simulated the proposed NCA approach using the $ns2$ network simulator with the CMU wireless extensions. We compared the energy efficiency of three algorithms, IEEE 802.11, IEEE 802.11 Power Saving Mode (PSM) and SPAN, using two versions each: one with NCA integrated and the other without NCA integrated. In each case, we simulated a 120-node network occupying a square region measuring 1000(m) X 1000(m) for 600 seconds. The nominal radio range was 250 meters and the bandwidth was 2Mbps. In this simulation, we used the power model of a mobile station shown in Table II which gives the average power consumption of a mobile station's NIC according to the state of the NIC [2].

TABLE II
Power consumption parameters used in Simulations (W)

Transmit	Receive	Idle	Sleep	Off
1.4	1.0	0.830	0.130	0.043

In our experimental setup, the total traffic consists of 20 connections, each of which sends a CBR flow to an another node, and each CBR flow consists of three 128-byte packets per second for a total traffic of 60Kbps. The initial positions of the nodes are chosen at random uniformly in the entire simulated region. For the mobility experiments, the motion of the nodes follows the random way point model with a maximum speed 20 (m/s).

Fig. 6 shows the average consumed power (mW) of the three naive algorithms, which are IEEE 802.11, PSM and

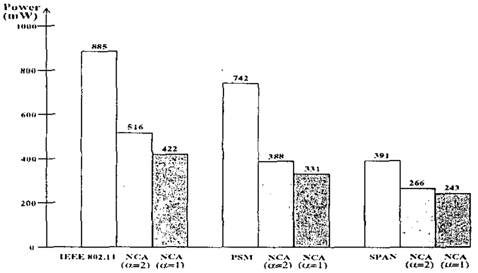


Fig. 6. Comparisons of average power consumptions

SPAN, and the six algorithms employing NCA with various values of α. To evaluate the effect of NCA according to *load sensitivity* (refer *Eq.*(11) in Section IV-B), we consider the two cases of NCA where α is either 1 or 2. Among the three naive algorithms, PSM and SPAN consume less power, 16% and 55% respectively, than the IEEE 802.11. When these algorithms use NCA (in the case of $\alpha = 1$), we obtain a power saving gain of 52%, 55% and 37%, respectively, for IEEE 802.11, PSM and SPAN. The largest power saving is achieved when both SPAN and NCA are utilized together, and the power consumption in this case is 243 (mW), which is much smaller. In the NCA algorithm, as the *load sensitivity* (α) is increased, the consumed power is increased, because the idle time of the mobile stations is increased due to the network capacity of the mobile stations being overestimated.

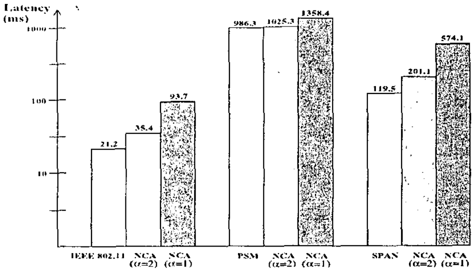


Fig. 7. Comparisons of average end-to-end packet delays

Fig. 7 shows the average end-to-end packet delay (ms) of successfully transmitted packets. In terms of packet latency, both PSM and SPAN increase the average end-to-end packet delay compared with IEEE 802.11. Moreover, when the NCA algorithm is applied to the three naive algorithms, the average packet delay is also increased by 341%, 37% and 380% for the IEEE 802.11, PSM and SPAN algorithm, respectively. The trade-off between the power saving gain and the packet latency when the NCA algorithm is used can clearly be ssen, since NCA decreases the network capacity in order to decrease the idle time of the mobile stations. As we increase α to 2, the average end-to-end packet delay is reduced by 66%, 3%, 68%, respectively, compared with the three naive algorithms. These are smaller increases than in the case where α is 1. Through the simulation results of both Fig. 6 and Fig. 7, we can deduce that when using the NCA algorithms, α can be used to control the tradeoff between the power consumption and the packet delay.

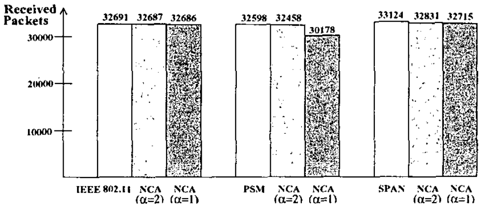


Fig. 8. Comparisons of throughputs

Fig. 8 shows the throughput in terms of the total number of received packets. In all cases of IEEE 802.11, PSM and SPAN, the number of received packets is almost the same, and the cases with NCA show similar results. This means that there is a possibility of power saving being accomplished without degradation of network throughput.

The above experimental results show that our proposed NCA algorithm can improve the energy efficiency of an ad hoc network without degrading the total throughput.

VI. Conclusion

An ad hoc network has some unique characteristics which need to be considered before undertaking its wide deployment. Firstly, there is a quite limited network capacity available for communication. Secondly, we must consider the power-related issues for the battery-operated mobile stations. In this paper, we suggest and simulate a new algorithm, called NCA, to handle energy consumption more efficiently in this limited network capacity environment. The proposed application-driven network capacity adaptation (NCA) algorithm improves the performance of an energy-constrained ad hoc network by means of energy saving via power-off, which prevents power from being wasted when the mobile stations' are in the idle state. An important feature of NCA is that it can cooperate with existed power management algorithms. The simulation results show that when operated with three naive power algorithms (i.e., IEEE 802.11, PSM and SPAN), the NCA algorithm can further decrease the consumed power without throughput being degraded.

References

[1] IEEE Computer Society, "802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," June 1997.

[2] L. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," IEEE INFOCOM'01, April 2001.

[3] S. Singh and C.S. Raghavendra, "PAMAS - Power-aware multi-access protocol with signaling for ad hoc networks," ACM Communication Review, July 1998.

[4] J. Monks, V. Bharghavan and W. Hwu, "A Power Controlled Multiple Access Protocol for Wireless Packet Networks," IEEE INFOCOM'01, April 2001.

[5] J. Gomez, A. Campbell, M. Naghshineh and C. Bisdikian, "PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks," ACM/Kluwer Journal on Wireless Networks (WINET), 2003.

[6] S. Lee, G. Ahn and A. Campbell, "Improving UDP and TCP Performance in Mobile Ad Hoc Networks with INSIGNIA," IEEE Communication Magazine, June 2001