Design and Evaluation of IP Header Compression for Cellular-Controlled P2P Networks

Tatiana K. Madsen[†], Qi Zhang[‡], Frank H.P. Fitzek[†], and Marcos Katz[♮]
[†]Department of Communications Technology, Aalborg University, Niels Jernes Vej 12, 9220 Aalborg Øst, Denmark,
e-mail: *[tatiana | ff]@kom.aau.dk*
[‡]Department of Communication, Optics & Materials, Technical University of Denmark
Building 343, DK-2800 Kgs. Lyngby, Denmark
e-mail: *qz@com.dtu.dk*
[♮]Technical Research Centre of Finland (VTT),
P.O.Box 1100 FI-90571, Oulu, Finland

***Abstract*— In this paper we advocate to exploit terminal cooperation to stabilize IP communication using header compression. The terminal cooperation is based on direct communication between terminals using short range communication and simultaneously being connected to the cellular service access point. The short range link is than used to provide first aid information to heal the decompressor state of the neighboring node in case of a packet loss on the cellular link. IP header compression schemes are used to increase the spectral and power efficiency loosing robustness of the communication compared to the uncompressed version. By introducing the terminal cooperation supporting header compression the robustness is increased. Within this article we will show that header compression should be applied to reduce the energy consumption of the terminals and moreover the header compression should be supported by cooperation to increase the robustness in terms of a decreased packet loss rate.**

I. INTRODUCTION AND MOTIVATION

Wireless multimedia services are set to play a big role in present and next generations mobile communication systems. Despite of the research efforts, delivery of IP-based services over cellular links is still a challenging task due to the bandwidth limitations. Additionally, the real-time requirements of such applications as Voice over IP (VoIP) should be taken into account. It has been realized that to overcome the resource limitations of the low-bandwidth links, IP header compression techniques can be successfully applied [1]. Header compression is finding its way into cellular networks: e.g. Robust Header Compression (ROHC) [2] has become a part of 3G standard.

However, high bit error rates of the wireless environment are harmful for the IP header compression. Packet losses lead to the failure of the decompression procedure and, upon detection of the missing packets, the context repair mechanism is triggered. The full context update is requested from the compressor entity that is typically located at the concentrator node in the network and not at the access point (AP). Due to propagation and processing delays, responsiveness of the networks on the channel error is low and a single packet error causes the appearance of the error burst. Typically, the application layer can deal with single packet losses, but bursty errors degrade significantly Quality of Service (QoS) perceived by a user. All conventional header compression schemes will suffer from this problem.

In this paper we propose a context update mechanism for header compression that helps to keep the decompressor operational without the need of excessive signalling to the AP. The proposed mechanism avoids bursty errors problem. It helps to keep jitter bounded and results in the improved QoS. The presented novel scheme is based on the concept of *micro cooperation* [3]. The omnipresent architecture in wireless communication consists of autarky terminals that individually receive services from an AP. If the terminals have the capabilities to establish peer-to-peer (P2P) connections with each other as well, e.g. using short-range links, cooperative groups can be formed. We refer to the network architecture formed in this way as cellular-controlled P2P networks. It has been noted that using short-range links for data transmission is less costly compared with the cellular links since higher data rates and lower powers for transmission and reception can be achieved over close distances [3]. Based on this observation, we ague for cooperative behavior of wireless terminals that exploit P2P connections for information exchange in order to keep their decompressor states updated.

The benefit of micro cooperation comes from the multi–path diversity: if a radio path between an AP and a terminal is greatly deteriorated by the instantaneous channel conditions, the whole IP packet is lost. A neighboring user might be experiencing good channel conditions and might be able to deliver information for the context update. The cooperative users are rewarded by keeping their decompressors operational.

The remainder of the paper is organized as follows. Section II describes the system under investigation, conventional approach for header compression and explains the proposed cooperative method. Section III elaborates on the protocol design for P2P connections. Performance evaluation is presented in Section IV. Section V gives some concluding remarks and outlines future work.

II. SYSTEM UNDER INVESTIGATION

The system under investigation follows the architecture presented in Figure 1. A number of wireless terminals connected to access points (APs) are interested in data services: downloading emails, listening to MP3 radio etc. As an example, we have chosen to consider Voice over IP (VoIP) application that is becoming more and more popular due to its reduction of telephone calls costs. In general case, the last wireless hop can be presented by the same or different wireless technologies, e.g. WLAN IEEE 802.11 or GPRS. The considerations given below will hold as long as they share the same concentrator.

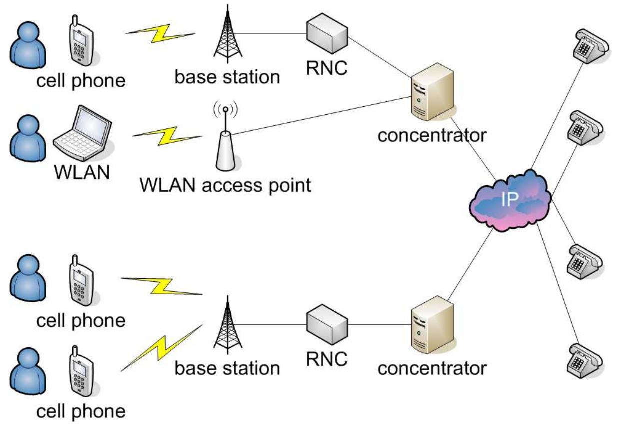

Fig. 1. System under investigation.

Speaking about data transmission over IP-based networks, a transport overhead due to the protocol headers should be taken into account. VoIP uses the Real-time Transmission protocol as transport layer protocol. RTP/UDP/IP suit causes 40 bytes overhead for IPv4 and 60 bytes for IPv6. Considering that a VoIP packet is typically 20 bytes (when G.729 is used), the total overhead is more than 70%. Larger address space and mobility support of IPv6 means even higher header costs. Transporting IP data over wireless networks, the IP headers consume a significant portion of precious bandwidth and limit the system capacity. One way to overcome this inefficiency is by using IP header compression.

A. Autonomous Operation (Non Cooperative Header Compression)

Traditionally, header compression schemes are designed to work on an individual link. Header compression is typically performed on the headers of the network layer and above. The packet headers are compressed at the sending node. The decompressor at the receiver side reconstitutes the original header before delivering the packet to the higher protocol layers.

A number of header compression schemes have been developed over last years. One should mention Compressed Real Time Protocol (CRTP) presented in RFC 2508 [4] for compression of RTP/UDP/IP streams or RObust Header Compression (ROHC) [2] that is especially developed for cellular networks. ROHC is designed to withstand the impairments of wireless channel: large bit error rates and long round-trip times.

Despite the fact that there exist over a dozen different schemes for different types of packet streams, they all rely on the same principle: redundancies between contiguous packets of a given flow allow significant reduction in size by using *differential encoding*. Random fields of packet headers are transmitted unchanged, whereas delta fields are compressed by reference to the previous packets, so called *CONTEXT*. The context is known and maintained at the receiver side as well and used for decompression of the incoming packets. M packets with full headers are required in order to establish the context at the decompressor; afterwards packets with compressed headers are sent. The context is updated with every new packet. Packet losses lead to inconsistencies in the context state at the decompressor and failure of the decompression procedure. Upon an update request from the receiver, packets with full headers will be transmitted in order to repair the context. However, in the omnipresent system architectures, compressor is located not at the AP but somewhere in the network (see Figure 1). For example, in GPRS system compression is performed in Serving GPRS Support Node (SGSN). The system can not be highly responsive to the update request and it can take d packets before the context will be re-established due to propagation and processing delays. We refer to the parameter d as *responsiveness* of the network on the channel errors.

If a number of wireless terminals establish data connections with an AP and all entities posses header compression capabilities, compression of packet streams will be performed individually for each active connection - we refer to this situation as *autonomous operation mode*. All conventional compression schemes are non-cooperative in their nature.

Let us assume that we can model each link between an AP and a WT using uncorrelated packet errors. We denote the probability to loose a packet as p. Let the random variable γ denote the number of packets within the cycle, i.e. the number of compressed headers that can be sent before the decompression procedure suffers from erroneous transmission and the context update takes place. The expectation of γ can be calculated as

$$N = E[\gamma] = \frac{1}{p} + d \tag{1}$$

In order words, we expect that in case of available feedback channel the context update will be repeated every N packets on average.

B. Cooperative Operation (Cooperative Header Compression)

Bits transmitted over air uses scarce wireless bandwidth and depletes batteries of mobile terminals. Considering the limited bandwidth resources and the current status of battery technology, increase in bandwidth savings and decrease in energy consumption are the utmost goals of compression schemes. These goals can potentially be achieved by improving robustness of the algorithms towards channel errors. In this paper we propose a context update mechanism that is less costly in terms of bandwidth than the conventional full context update and that can support QoS in terms of packet delivery ratio.

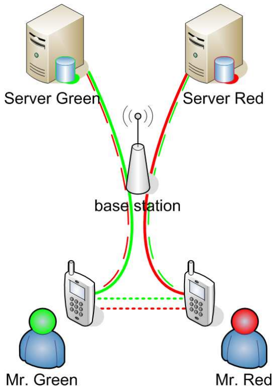

Fig. 2. Example of a cooperative group consisting of two terminals.

Let assume that J terminals form a cooperative group (Fig. 2). We assume that each WT has the capability of communicating with the AP and simultaneously with other terminals (by using either the same or different air interfaces). Each terminal receives a data stream with compressed packet headers together with some additional information intended to the neighboring terminals. This information, refereed to as *additional information container (AIC)*, can be used to update the context state of the decompressor - note that we are speaking about the decompressors of other terminals within the cooperative group. Each terminal will receive $J-1$ AICs. AICs exchange is performed on demand using P2P connections. In order to facilitate the context maintenance, an AIC should contain differentially encoded fields of a compressed header. For example, if CRTP scheme is used for compression, the size of an AIC is 2 bytes (total size of a compressed header of 4 bytes minus 2 bytes reserved for random fields, such as UDP checksum).

Since the context update from the AP is required only in case when all WTs fail to receive packets correctly at the same time instant, the average cycle length can be found as

$$N = E[\gamma]_{coop} = \frac{1}{p^J} + d \tag{2}$$

From formulas (1) and (2) one can see that the average cycle length in cooperative scenario is longer compared with the conventional schemes. It means that more packets with compressed headers can be sent resulting in better compression gain and better bandwidth efficiency.

The presented above scheme is an extension of Cooperative Header Compression (COHC) [5]. COHC has been proven to be an efficient scheme for multichannel communication systems.

III. DESIGN OF AICS EXCHANGE OVER SHORT RANGE LINK

In order to limit the costs of AICs exchange, it should be performed on demand, i.e., by a request issued by a terminal. Additionally, we assume that the packet streams for different terminals have the same rate. In this case the AICs on different channels are received in tact by terminals. Examples of such synchronous streams are VoIP and MP3-radio. In case of asynchronous streams, the AICs sending rate should be adjusted to the rate of the packet flows.

Furthermore, two possible approaches in the design of AICs exchange can be distinguished: centralized and distributed. In the first approach, a leader of the cooperating group should be elected and it will assume a controlling role. In the second approach all terminals have the same role.

In this paper we consider Bluetooth technology to provide short-range connectivity. One terminal is assigned a role of a master and all communication flows go through this terminal. Therefore, it is natural to grant a role of the leader of AICs exchange procedure to the master. Let us assume that a cooperating group constitutes a piconet with a master and $J-1$ slaves ($J \leq 8$ - it is unrealistic to have more than 8 cooperative terminals in a group due to increase in overhead with the increase in the number of terminals). In order to minimize power consumption, the usage of low-power modes is advisable. We propose to use park mode:

- Park mode has the lowest duty cycle comparing with hold and sniff modes. If no AICs exchange is needed, the only nodes' activity is periodical listening to the beacon signal from the master for synchronization purpose.
- Being parked, a slave can send a request to the master. In all other modes, a slave can not initiate communication without first being polled. This feature of park mode can be successfully utilized when occasional information exchange between the nodes is foreseen.

Assume k slaves have missed packets at one time instant. They send unpark request to the master. If the master has received a packet with the compressed header, AICs and data from the AP correctly, it unparks k slaves and sends corresponding AICs to them. If the master misses a packet himself, it unparks an additional slave (the one that has not requested unparking). After unparking procedure, first the master polls the slave to collect from it k AICs and forward them to the k other slaves. If the master received $J-1$ unparking requests and it has not received a packet from AP, then no unparking is performed. If a slave is not unparked, it assumes that no healing of the context can be done by using short-range AICs exchange and it sends a full context update request to the AP.

Let assume that 1-slot DM1 packets are used for transmission (the choice of DM packet over DH is due to its high error-resilience). The payload of DM1 packet contains up to 17 information bytes. This corresponds to the 8 AICs 2 bytes each. Therefore, one packet can contain all necessary AICs. In order to reduce the number of messages sent, instead of sending AICs individually to each slave, the master can broadcast one packet containing all requested AICs. What is more, according to the Bluetooth specification [6], unparking of the slaves can be avoided if general broadcast messages should be carried to the parked slaves. In the beacon train, the master can support broadcast messages to the slaves. However, the unparking will be necessary if the master has lost AICs and has to acquire them from another slave. The master unparks a slave by sending a dedicated LMP command. The first packet sent by the master must be POLL packet [6]. The return packet in response to the POLL packet will contain AICs and it will confirm that the slave has been unparked.

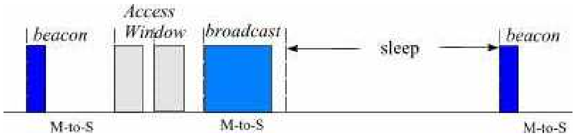

Fig. 3. Communication flow over Bluetooth link: first scenario.

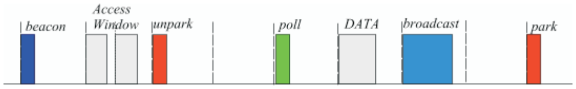

Fig. 4. Communication flow over Bluetooth link: second scenario.

Using these capabilities defined in the specification, only a few slots will be required for AICs exchange. In the first case (when master has AICs for other nodes), the procedure requires 3 communication slots (see Fig. 3); in the second case 7 slots, including overhead for parking/ unparking a slave (Fig. 4). In case of packet losses over short-range link, retransmissions are required and the number of occupied slots can be higher.

Power burden of the communication procedure described above lies on the master of the piconet. To guarantee the fairness among nodes, master-slave switch should be performed periodically where all members of the cooperative group take a role of the master in turn.

One should note that the proposed scheme of the centralized AICs exchange is just one of the many possible algorithms and not the most optimal one.

IV. PERFORMANCE EVALUATION

A. Bandwidth savings

The expected bandwidth savings due to header compression can be calculated as

$$E[bs] = 1 - \frac{M(H_U+x)+(N-M-d-1)(H_C+x)+(d+1)(H_u+x)}{N(H_U+x)}$$

where x is a payload (assumed to be the same for all packets in a stream), H_U and H_C are the sizes of uncompressed and compressed headers, respectively. The results are plotted for the following values of the headers: $H_U = 40$ bytes and

$$H_C = \begin{cases} 4, & non-coop, \\ 4+2(J-1), & coop. \end{cases}$$

The formulas are derived under assumption of a perfect feedback channel. If feedback is error-prone, a burst of packet losses due to re-synchronization of the compressor and decompressor becomes even larger and bandwidth savings are decreased. However, error-prone feedback would have a drastic impact on the performance of the non-cooperative header compression scheme, while its impact on the performance of the cooperative scheme is marginal. In this paper we have chosen to present results only for the case of ideal feedback channel. Note that by doing so we overestimate the efficiency of the conventional scheme.

B. Packet error probability (PEP)

Robustness of a compressed scheme can be judged by Packet Error Probability (PEP). Evaluating PEP, we take into account both losses of packets due to the channel condition and decompression errors. The following formulas are derived for average values of PEP:

$$E\left[PEP_{coop}\right] = \frac{\frac{1}{p^{J-1}}+d}{\frac{1}{p^{J}}+d} \tag{3}$$

By substituting $J = 1$ in formula 3 we get $E\left[PEP_{non-coop}\right]$ for non-cooperative case.

C. Energy consumption

In this paper we consider applications, where data packets are sent periodically, e.g. every 20 ms as for VoIP. The energy consumed by a terminal is in reception over cellular link. In our calculations we omit the energy spent for the transmission of update requests (in this way we underestimate the energy spent in non-cooperative case). Additionally, in cooperative case we have to take into account the energy spent on the communication over short-range link. The total energy spent over cellular link during one cycle can be represented as:

$$E_{cell} = \left(P_{c,rx} \cdot c_{c,rx} + P_{c,i} \cdot c_{c,i}\right) t \tag{4}$$

where

- $P_{c,rx}$ ($P_{c,i}$) is a power consumed by the terminal for the reception over cellular link (when the cellular link is idle);
- $c_{c,rx}$ ($c_{c,i}$) is the proportion of time spent on the reception on the cellular link (when the link is idle);
- t is the duration of the cycle.

The average energy consumed per packet is:

$$E = E_{cell}/N$$

Uncompressed case:

Formula (4) can be used for evaluation of energy consumption when packets are sent with an uncompressed header if we take the cycle length equal to one.

Non-cooperative Header Compression:

The following expression should be substituted in formula (4):

$$c_{c,rx} = \frac{M(H_U+x)+(N-M)(H_C+x)}{v \cdot m \cdot N}$$

$$c_{c,i} = 1 - c_{c,rx} \tag{5}$$

$$t = N \cdot m$$

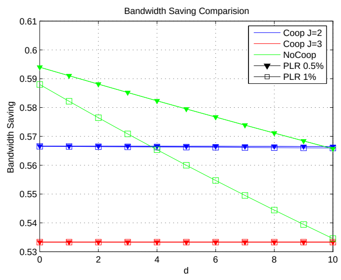

Fig. 5. Bandwidth Saving Comparison. $PLR = 0.5\%,\ 1\%$.

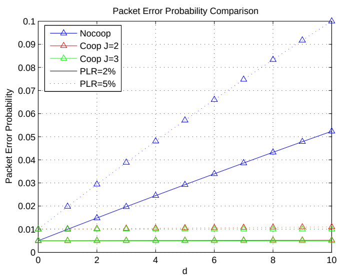

Fig. 6. Packet Error Probability Comparison. $PLR = 0.5\%,\ 1\%$.

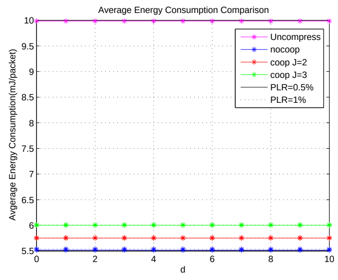

Fig. 7. Average Energy Consumption Comparison. $PLR = 0.5\%,\ 1\%$.

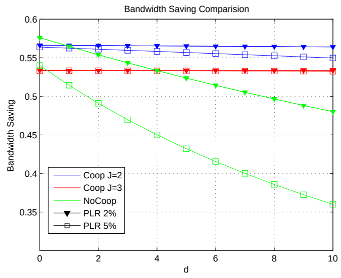

Fig. 8. Bandwidth Saving Comparison. $PLR = 2\%,\ 5\%$.

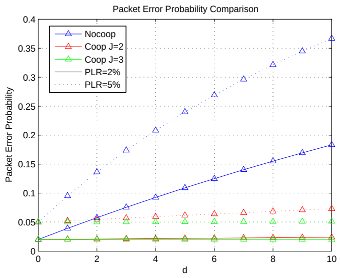

Fig. 9. Packet Error Probability Comparison. $PLR = 2\%,\ 5\%$.

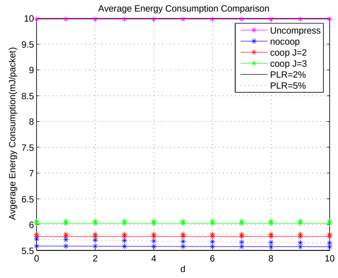

Fig. 10. Average Energy Consumption Comparison. $PLR = 2\%,\ 5\%$.

where v is a data rate (if GPRS is used to provide cellular connectivity and CS1, 8 time slots are used for transmission, the corresponding data rate is 72.4 kbps) and $m = 20$ms.

Cooperative Header Compression:

The average power consumed per packet on the cellular link in cooperative case can be calculated using formulas (3) and (5). Note that in this case the values for N and H_C are different from the non-cooperative scenario. Additionally, we should calculate the energy spent on the short-range link. Table I gives the considered values for power for cellular and short-range connections [7].

The duration of one AIC exchange on average is

$$t_{single} = \frac{1}{J} \cdot 7slots \cdot 625\mu s + \frac{J-1}{J} \cdot 3slots \cdot 625\mu s$$

Therefore, the terminals will have their short-range link active for t_{act} sec during one cycle:

$$t_{act} = t_{single} \cdot N \cdot p \cdot J$$

The rest of the time the terminals will be in sleep mode:

$$t_{sleep} = t - t_{act}$$

The total energy consumed over short-range link can be found as:

$$E_{sh-range} = t_{act} \cdot \frac{P_{TX} + P_{RX}}{2} + t_{sleep} \cdot P_{sleep}$$

and the total energy spent in cooperative case is given by

$$E_{coop} = E_{cell} + E_{sh-range}$$

Power Type	$P_{c,rx}$	$P_{c,i}$	$P_{sr,tx}$	$P_{sr,rx}$	$P_{sr,i}$	$P_{sr,sleep}$
mill Watt	1254	125	100	78	0.486	0.054

TABLE I
TYPICAL POWER VALUES FOR CELLULAR LINK AND SHORT RANGE LINK (BLUETOOTH) COMMUNICATION

D. Energy efficiency

Energy efficiency is defined as

$$\eta = \frac{\text{useful data transmitted}}{\text{energy spent}} = \frac{x \cdot (1 - E[PEP])}{E}$$

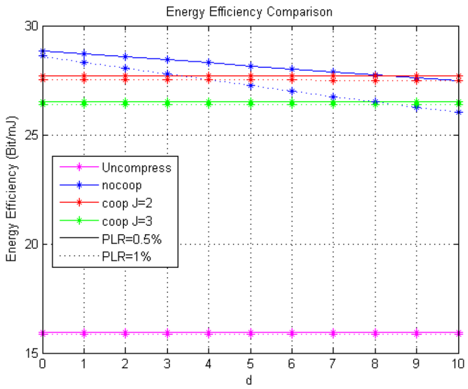

Fig. 11. Energy Efficiency Comparison.

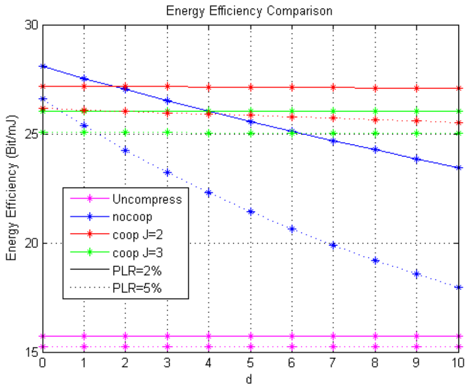

Fig. 12. Energy Efficiency Comparison.

E. *Results*

Figures 5- 12 present comparison of non-cooperative and cooperative approaches for header compression in terms of bandwidth savings, packet error probability, average energy consumption per packet and energy efficiency. All figures are plotted as a function of the network responsiveness, d. The case $d = 0$ corresponds to the ideal situation when the context update is received immediately after a de-synchronization between the decompressor and compressor appears. The results are given for the cooperative groups consisting of two and three terminals ($J = 2$ and $J = 3$) and for four different values of individual packet losses (packet loss rate $PLR = 0.5\%$, 1%, 2% and 5%).

The performance of non-cooperative compression strategy decreases with increase in d. However, the curves for cooperative method remain almost constant. This can be explained by the large cycle length in cooperative case. Due to the large values of N, the average values for BS, PEP and energy efficiency are not affected by the relatively small changed in d.

The higher packet loss rate results in higher payoff of micro cooperation. In the case of $PLR = 2\%$ and 5%, the cooperative strategy outperforms the non-cooperative one. However, in the case of good channel conditions (low values of PLR), the context update will be required seldom and, therefore, there is no need to invest the resources in cooperation. Terminals that experience very low PLR should sustain from cooperation and do not join cooperative groups.

The number of three cooperative terminals ($J = 3$) shows the lowest packet error probability. However, the bandwidth and energy efficiency for $J = 2$ is higher compared with $J = 3$. The later case corresponds to the larger compressed header size and, thus, more resources per bit of useful information are spent. We can conclude that the number of two cooperative terminals are optimal.

Considering the average energy consumed per packet, we observe that using header compression the terminals consume nearly half of the energy used in the case of no compression. This emphasizes the importance of compression for the battery-driven hand-held devices. Comparing results when compression is applied, one can note that the consumption is slightly higher for cooperative case. This is due to the larger packet sizes and energy spent in Bluetooth piconet communication. Additionally, we should take into account overhead in terms of energy consumption caused by formation of cooperative clusters. However, as calculations show the energy spent for short-range communication is much less compared with the energy consumed for reception over cellular link. The slight increase in energy consumption can be considered as a trade-off with the robustness of the scheme. The cooperative scheme ensures high robustness and it is able to sustain channel errors without loosing the context state.

V. CONCLUSIONS

In this paper we have introduced a novel communication architecture referred to as cellular controlled peer to peer (P2P) networking. In this work we have used this architecture to support IP header compression schemes. It's benefits are shown in terms of bandwidth savings, packet error probability, and energy consumption. The investigation was based on the assumption of a feedback channel to react on lost packets and reinitialize the decompressor state in case of lost packets. The feedback channel responsiveness was varied from low to large delays. This is needed to reflect the architecture of existing 2.5G and 3G, and future 4G networks. For a varying value of the responsiveness and packet error rates, we could show that header compression schemes (non-cooperative and cooperative) outperform the uncompressed communication in terms of energy consumption by consuming nearly half. The advantage of the cooperative scheme over the non cooperative one can be shown in terms of packet loss. While the non cooperative scheme is suffering by packet loss propagation due to non correct decompressor states, the cooperative scheme is often able to adjust the state by first aid information from the neighboring node.

REFERENCES

[1] F.H.P. Fitzek, S. Hendrata, P. Seeling, M. Reisslein. Chapter in Wireless Internet – Header Compression Schemes for Wireless Internet Access. CRC Press. 2004.

[2] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng, Robust Header Compression: Framework and four profiles, Request for Comments 3095, July 2001.

[3] F.H.P. Fitzek, M. Katz (Ed.). Cooperation in Wireless Networks: Principles and Applications. Springer. 2006.

[4] S. Casner and V. Jacobson, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, Request for Comments 2508, February 1999.

[5] F.H.P. Fitzek, T. K. Madsen, P. Popovski, R. Prasad and M. Katz. Cooperative IP Header Compression for Parallel Channels in Wireless Meshed Networks. In *IEEE International Conference on Communication (ICC)*, Seoul, Korea, 2005.

[6] Specification of the Bluetooth system, Std., Rev. 2.0. [Online]. Available: http://www.bluetooth.com

[7] E. Shih, P. Bahl, M. Sinclair, Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In Proceedings of MOBICOM'02. Atlanta, USA. September 2002.