This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

Network Coding and Service Reneging for Real-Time Communication in Sensor Networks

Marc Aoun*† and Antonios Argyriou‡
*Distributed Sensor Systems Group, Philips Research, The Netherlands
†Institute for Networked Systems, RWTH Aachen, Germany
‡Department of Electrical and Computer Engineering, University of Thessaly, Greece

***Abstract*—In a number of application domains, the volatility of the environment where Wireless Sensor Networks (WSNs) should operate engenders time constraints on the generation, processing, and communication of sensory data. In this paper, we investigate the use of network coding and service reneging for the timely delivery of messages between wireless sensor nodes that utilize the IEEE 802.15.4 MAC/PHY standard. More specifically, we study the real-time performance of algebraic coding and traffic regulation under different traffic loads and for different timeliness requirements. Our simulation results indicate that both techniques can be on their own effective measures to improve on-time delivery of messages in overload conditions. An additional gain in performance is obtained when coding is carefully coupled with a traffic regulation based on proactive cleaning of transmission queues.**

***Index Terms*—Sensor networks, network coding, real-time traffic, packet deadlines, queue management.**

I. Introduction

A key characteristic that differentiates Wireless Sensor Networks (WSNs) from other systems is their proximity to the environment they monitor. The main function of sensor networks is reporting the state of the monitored physical environment. The latter often being dynamic and volatile means that the *state snapshot* made through sensing by a deployed WSN remains valid for a limited amount of time, after which it becomes outdated and its underlying sensed data obsolete. For example, the vital signs defining the status of a patient are dynamic by nature. Related sensed values, measured by a Body Sensor Network (BSN), would provide the status of the patient at a particular point in time. They remain valid, as well as any conclusion based on them, only as long as the vital signs are stable. Any subsequent actuation (e.g. automatic update in drug administration) is correct as long as the information it is based upon is still a valid depiction of the status of the patient.

Additionally, it is expected that originally decoupled, application-specific networks, will cooperate and converge towards supporting multiple applications in a concurrent manner. Such a convergence, that leverages the existence of standards, would practically mean that different rates and types of information will have to be dealt with by the same device; data with no specific timing requirements, and information with different time-constraints concretized by specific hard, firm or soft deadlines. The mentioned convergence furthermore means that the network could be faced with transient or permanent overload conditions.

Antonios Argyriou would like to acknowledge the support from the European Commission through the STREP project CONECT (FP7ICT257616).

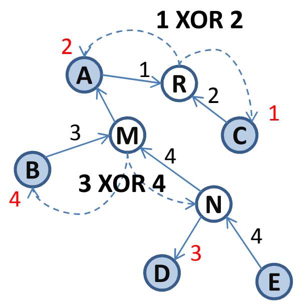

Fig. 1. Opportunistic network coding in a generic network topology.

Given the above emerging scenarios, we take a look at WSNs from a real-time perspective where the definition of correctness now spans two dimensions: correctness in computation/sensing/analysis, and timeliness in delivering information and performing actions. At the communication level, two aspects come into light: the timely delivery of information to achieve correct application behavior (e.g. timely and correct actuation) and the elimination of obsolete information in order to avoid false conclusions and reduce useless overhead.

Although a fair amount of work has targeted the timeliness aspect, research into real-time behavior and its enablers in wireless sensor networking is still in its infancy. This was to be expected in a domain where fundamental challenges, at for example the physical (PHY), medium access control (MAC) and network (NWK) layers had to first be addressed before focusing on Quality-of-Service and related issues. Existing approaches try to address timeliness at specific layers separately [1], [2], [3], [4], [5], or with sophisticated cross-layer techniques [6], [7]. In this paper we investigate a novel approach for improving real-time performance by considering recent information-theoretic advances. More specifically we explore the idea of network coding [8], where routing elements in a network can execute algebraic coding operations on packets besides simply forwarding them. In the case of wireless networks, network coding takes advantage of the broadcast nature of the channel in order to increase the information content per packet transmission [9].

Fig. 1 illustrates how opportunistic coding operates in a network; Nodes A and C want to communicate packets 1 and 2 to each other, respectively. Node R, acting as router between the two nodes, takes advantage of the simultaneous availability of packets 1 and 2, codes the bits of the two packets together through a simple XOR operation, and broadcasts the resulting coded packet, instead of two broadcasts. The successful reception of this packet by nodes A and C means that both packets 1 and 2 can be successfully retrieved by decoding at nodes C and A, respectively, provided that both nodes have maintained local copies of their transmitted packets. Fig. 1 also illustrates a more generic scenario; node B is communicating with node D, and node E is communicating with node B. Router node M applies network coding on the packet originating from B and the one originating from E, that was forwarded by N. It becomes straightforward to conclude that network coding is also applicable when multiple router nodes separate communicating nodes and that it can be applied repeatedly at the router nodes along the communication paths.

This simple, yet powerful idea has been applied in IEEE 802.11-based multi-hop wireless networks and throughput benefits in the order of 3-4 times over the baseline 802.11 have been demonstrated for bulk data transfers [9], [10], [11]. In this paper we take an additional step by studying its performance in the case where packets have specific delivery deadlines, i.e. when throughput and goodput hold different meanings.

Our contribution in this paper is three-fold: 1) We investigate network coding from a real-time perspective, i.e. by assessing its impact on the delivery of real-time data (goodput) that are associated with *per-packet deadlines*. This is in contrast with previous work where network coding was primarily employed with the purpose of improving throughput and delay optimization at the flow-level. 2) We investigate the impact that service reneging through proactive queue cleaning can have on timeliness 3) We investigate the joint real-time performance of the combination of coding and service reneging, and assess their relative contributions for different loads and different timing requirements.

To the best of our knowledge, our work is the first to consider network coding over IEEE 802.15.4 networks and to investigate coding as a method to improve real-time performance.

II. Terminology

For convenience, we shall define in this section the main terms used throughout this work.

A. Packet Related

- **Coded packet:** a packet whose payload resulted from a XOR operation on the payload of two other packets.
- **Plain packet:** a packet that does not carry XOR-ed data.

B. Functionality Related

- **Service reneging:** Removing a packet from a transmission queue to which it has been already admitted.

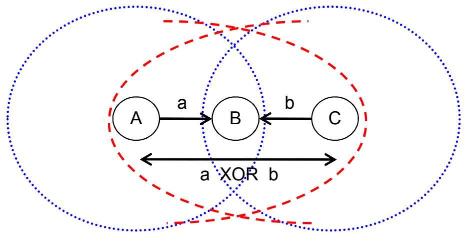

Fig. 2. Fundamental two-hop network topology for studying the performance of network coding and service reneging.

- **Packet servicing:** Processing of a packet at the MAC layer. Servicing starts when a packet is committed to the MAC, and ends upon successful transmission or overall failure (due to recurrent channel access failure).
- **Opportunistic coding:** Network coding operation that exploits the coincidental presence of two packets that could be coded. In essence, a synchronous presence is not required per se for the normal functioning of the network, but is leveraged when it occurs.

C. Timing Related

- **Generation time (g):** The moment in time when a packet is generated at the application layer.
- **Relative deadline (D):** The amount of time, relative to g, after which a packet's data becomes obsolete.
- **Absolute deadline (d):** The moment in time when a packet becomes obsolete. Therefore, $d = g + D$.
- **Lead time:** The amount of time, measured relative to the current time, after which the absolute deadline of the packet will be reached.
- **Queue admission time:** The moment in time when a packet is admitted to a queue.
- **Service Time:** The amount of time it takes the MAC layer to fully serve a packet, and send feedback to the upper layer upon successful or unsuccessful service completion.

III. System Model and Assumptions

The system under consideration is depicted in Fig. 2. In this paper we focus on this fundamental network coding scenario with 3 nodes for multiple reasons: It allows us to decouple the problem we want to study from other parameters (like the effect of multiple communicating nodes) and allows future real-hardware validation. We avoid the issues related to scheduling and coding multiple flows and focus on the fundamental performance contribution that network coding can achieve for timeliness. It is straightforward to deduce, through a comparison between Fig. 1 and Fig. 2, that our choice of a fundamental, reduced topology is in no way restricting or does not fully capture the impact of network coding on timeliness. With respect to the traffic pattern, the two edge nodes communicate with each other, acting concurrently as sources and sinks of information. Edge nodes are not within reliable communication range, thus requiring a router. This scenario is realistic since the transmission range is usually quite shorter than the carrier sensing range [12].

Edge nodes maintain two queues; a transmission queue (*TxQ*) and a decoding queue (*DcQ*). The *TxQ* holds messages that are generated by the application layer of the node and need to be transmitted. The *DcQ* holds copies of messages that have already been transmitted. These copies are used to decode algebraically coded packets and retrieve their information content. The router node maintains two reception queues, RxQ_1, RxQ_2. A reception queue RxQ_i is associated with edge node i; it stores messages that were received from that node and need to be forwarded. All the aforementioned queues are implemented at the network layer. No modifications are introduced to the IEEE 802.15.4 MAC.

A. MAC/PHY Layer

At a lower layer, the IEEE 802.15.4 MAC/PHY was used [13]. The IEEE 802.15.4 is the most popular standard for low data rate wireless PANs. The standard specifies four physical layers (PHY) where three of them are based on direct sequence spread spectrum (DSSS) techniques. We considered the DSSS PHY working in the 2450 MHz band with a data rate of 250 Kbps. Regarding the MAC layer, the IEEE 802.15.4 specifies a beacon-enabled mode and a beaconless mode. In beacon-enabled mode, channel access is arbitrated through the use of a slotted CSMA/CA algorithm. Beaconless mode employs non-slotted CSMA/CA. In this work we adopted the beaconless mode.

Three configurable attributes define the functioning of non-slotted CSMA/CA: *macMinBE*, *macMaxBE*, and *macMaxCSMABackoffs*. The algorithm functions as follows: When a node has a packet to send, it first chooses a random number *n* of backoff periods (each equal to 320 μs), where n is uniformly distributed between 0 and 2^{BE}-1. The variable BE refers to the Backoff Exponent, that is initially set to *macMinBE*. Once *n* backoff periods elapse, the node checks whether the channel is free or not. If it is free, the node initiates the packet transmission. If on the other hand the channel is busy, BE is incremented by 1, unless it has already reached the value adopted by *macMaxBE*. In that latter case, BE is set to *macMaxBE*. The process of choosing a random number *n* of periods, waiting for them to elapse and then checking the channel is again repeated. In total, the MAC will try this process *macMaxCSMABackoffs* times, after which it will report a failure to the upper layer in case the channel is never found to be free. Otherwise, a success will be reported since in this paper we assume broadcast transmissions without ACKs.

IV. Mechanisms Behavior

Three mechanisms were implemented on top of the 802.15.4 MAC layer and are described as follows.

A. Handling of New Packets and Scheduling

New packets are generated at the application layer of the edge nodes and are subsequently forwarded to the network layer and presented for admission at the *TxQ* queue. If the queue is full, the packet is discarded (Drop-Tail policy). Otherwise, it is admitted to the queue. Similar admission behavior takes place at the router node for the *RxQ* queues. Queues are served using the First Come First Served (FCFS) scheduling policy, where the packet with the smallest queue admission time among all packets residing in the queue is committed first to the MAC layer for servicing. FCFS Drop-Tail is the most widespread used policy in wireless ad-hoc networks [14]. Note that no queue build-up is happening at the MAC layer. Packets are committed one packet at a time in order to have full control at the network layer over which packet to serve next and which packet to renege. At the edge nodes, a copy of a committed packet is stored in the *DcQ*. The copy is removed when node B is overheard forwarding this packet or any other previously committed packet that has a higher sequence number. In case DcQ is full, no copy is stored and the packet is flagged to inform node B that it should not be coded.

B. Algebraic Coding

Coding, when activated, takes place at the router node when both Rx queues are populated and the MAC layer is ready to service a packet. The payload of the Head-of-Line (HOL) packets of both queues are algebraically coded with each other, resulting in a coded packet. The sequence number of both packets is appended to the payload, resuling in a minor extra overhead of 2 bytes (assuming a periodic reuse of the possible sequence number values).

C. Service Reneging

The service reneging procedure, when activated, proactively cleans the queues upon the arrival of a new packet in order to potentially free buffer space. The procedure is also executed before coding and before any type of packet is committed to the MAC layer in order to avoid serving packets that are expired or are unlikely to meet their timing constraints.

In order to decide whether to execute service reneging for a particular packet, an estimate of the MAC service time is used. This estimate is set to the mean value of the 10 most recent encountered service time values that resulted from transmitting the last 10 packets. To account for the actual 2-hop communication path, edge nodes multiply the calculated estimate by 2. Thus, they rely on a rather optimistic estimation of the waiting time plus service time that their packets will experience at the router node. The lead time of every packet in the queue is compared to the estimated service time. If the lead time is smaller, the packet is dropped from the queue. Otherwise, it is maintained.

V. Performance Evaluation

A. Simulation Setup

We have built our performance assessment environment in OMNet++[15], on top of simulation models of wireless propagation, multiple access interference, radio state machine and the IEEE 802.15.4 non beacon-enabled MAC protocol. The MAC layer is configured with $macMinBE$=3,

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

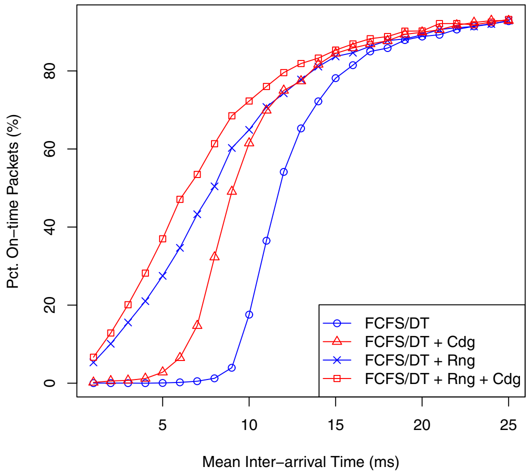

Fig. 3. Percentage of packets received at the edge nodes within their deadlines, for the baseline case, coding (Cdg), reneging (Rng) and the combination of Rng and Cdg.

$macMaxBE$=5, and $macMaxCSMABackoffs$=4, that correspond to the default values specified in the standard [13].

The two edge nodes A and C depicted in Fig. 2 generate packets following an exponentially distributed inter-arrival time with mean τ. Each node generates 10000 messages in total. The value of τ is increased gradually in the simulations to assess the system performance under different traffic loads. Packets have to traverse two hops (communication is end-to-end between nodes A and C). Plain packets have a physical frame size of 60 bytes and coded packets 62 bytes.

Every generated packet has a deadline associated with it. In the first part of the evaluation, deadlines are uniformly distributed over the interval [10ms,100ms], thus with a mean value equal to 55ms. In a second phase, the mean deadline is varied in order to assess the performance of coding and reneging for different timeliness requirements.

The size of *TxQ* for each edge node is equal to 10 packets and the *DcQ* has a size equal to 20 packets. Each of the two *RxQ* queues at node B can hold up to 10 packets.

B. Results and Analysis for Constant Average Deadline

In all presented figures, FCFS/DT refers to the simple baseline mechanism (First Come First Served with Drop-Tail). The acronym *Cdg* refers to network coding, and *Rng* refers to service reneging. The additional 2-byte overhead in coded packets is accounted for in the results where needed.

Fig. 3 shows, for different mechanisms, the percentage of packets that make it on time to their destination out of all 20000 generated packets. Fig. 4 provides the average buffer space occupancy at the router node. Fig. 5 illustrates the average MAC service time at node B.

Among the studied approaches, baseline FCFS/DT performs the worst, in particular for heavy loads (1ms $< \tau <$ 20ms). Applying algebraic coding at node B provides a gain that reaches up to 50% over the baseline case. This performance improvement can be explained by studying Fig. 4. By XOR-ing payloads and servicing two packets concurrently, network coding considerably reduces the overall buffer usage at node B. This in turn reduces the number of messages dropped at node B by the Drop-Tail policy. Servicing two packets concurrently further reduces the time that packets have to wait in the *RxQ* of node B before having access to the MAC layer for servicing. The relative gain in goodput of network coding alone is nevertheless negligible for values of τ smaller than 5ms. In such overload conditions, a bottleneck exists at nodes A and C, with a high percentage of packets being dropped at these nodes by the Drop-Tail mechanism or otherwise having to experience significant waiting times in the *TxQ* queues once admitted (the average queue size at the edge nodes was measured to be equal to 9.5 packets).

Service reneging outperforms both baseline FCFS/DT and simple algebraic coding. By proactively dropping packets that have a high probability of missing their deadlines, reneging favors those packets which are more likely to satisfy their timing requirements. On the other hand, simple FCFS/DT and its network coding variant treat every single packet that is admitted to the transmission queue. Both remain oblivious to the fact that buffering and servicing short-deadlined and expired packets consumes both precious buffer space and MAC service time. This in turn increases the number of packets rejected by Drop-Tail and increases the waiting time of other admitted packets with more relaxed deadlines. The reduction in congestion obtained through service reneging is correlated with a decrease in average service time, as illustrated in Fig. 5, and with a reduction in buffer occupancy, as illustrated in Fig. 4. A similar reduction in buffer occupancy at the edge nodes was also observed.

The ability of network coding to simultaneously serve two packets at router B, adds to the efficient buffer space and MAC usage efficiency provided by service reneging. As shown in Fig. 3, a combination of coding and reneging performs the best among all tested cases. Applying network coding in addition to proactively cleaning queues reduces the waiting time of packets at node B and reduces further the average queue size at the router (Fig. 4).

Overall, applying a combination of network coding and service reneging can provide up to 70% more on-time packets than traditional FCFS Drop-Tail.

C. Results and Analysis for Varying Average Deadlines

We shall now study the real-time performance of reneging and its combination with coding for different timing requirements. For that purpose, we vary the average deadline of packets, while maintaining a uniform deadline distribution with constant standard deviation. The percentage increase in real-time goodput (relative to FCFS/DT) is plotted in Fig. 6 for service reneging. The figure shows a clear benefit of reneging for mean inter-arrival times τ smaller than 15ms and average deadlines up to 200ms. A peak increase of 52% is noted for an average deadline equal to 55ms. The higher the deadline is, the less the impact of proactive queue cleaning becomes. For

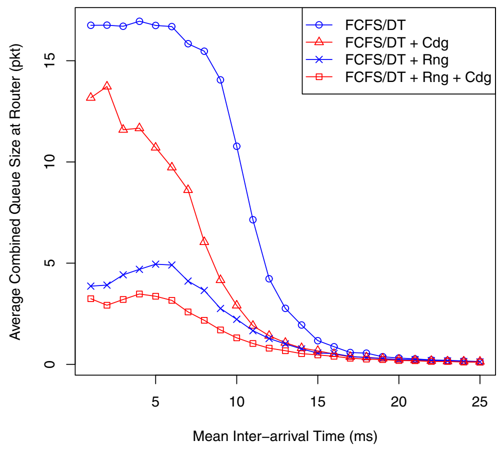

Fig. 4. Average combined size of RxQ_1 and RxQ_2 at node B.

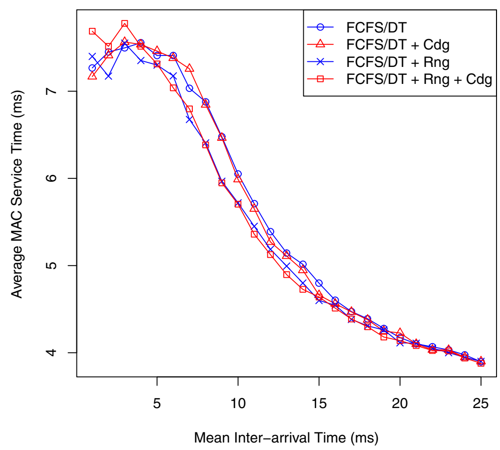

Fig. 5. Average MAC service time per packet at the router node B.

every average deadline value, the best performance occurs for τ equal to 7ms. We shall come back to this point hereafter.

To compare the additional gain that can be obtained when using network coding in combination with reneging, we plot in Fig. 7 the percentage increase of the combined approach *relative to reneging alone*. A clear benefit of network coding is witnessed in the same range of τ where reneging is active. The additional gain first increases with the absolute deadline. Indeed, the bigger the deadlines are, the less messages will be discarded by service reneging, and thus the more packets will make it through to the router and be coded. The gain then reaches a peak value which it maintains for all average deadlines bigger than 200ms. In this operation region, service reneging is practically inactive, with zero gain increase over FCFS/DT. The ability of coding to reduce buffer usage and thus reduce the number of dropped packets at node B provides this constant gain irrespective of the average deadline.

An interesting result can be seen in Fig. 6 and Fig. 7; the maximum gain is obtained for a value of τ equal to 7ms. By taking a close look at Fig. 5, one can notice that for τ equal to 7ms, the service time is slightly less than 7ms (for the Cdg + Rng case). The same value was observed at the edge nodes. From a queuing theory perspective, this (7ms,7ms) operating point corresponds to the equilibrium point for all three nodes in the network. More specifically, for the edge nodes one packet is received every 7ms on average, and one packet is sent also every 7ms on average. For node B, two packets (one from A and one from C) are received every 7ms, and due to coding, two packets on average can be serviced every 7ms. The equilibrium point is the optimal operation point when network coding is employed. This optimality is due to the fact that the queues in the system become stable.

Fig. 8 and Fig. 9 depict the total MAC layer effort in terms of *over-the-air* transmitted packets (N_{tx}). In particular for the Cdg case we show the ratio:

$$\frac{N_{tx}^{FCFS/DT}(\tau) - N_{tx}^{Cdg}(\tau)}{N_{tx}^{FCFS/DT}(\tau)} \times 100\% \tag{1}$$

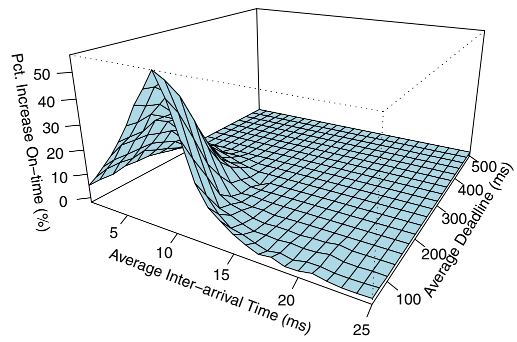

Fig. 6. Percentage increase in number of on-time packets for reneging (Rng) relative to FCFS/DT.

This aspect of the system performance is important; both reneging and coding reduce the amount of physically transmitted packets, which translates into a more efficient usage of the wireless channel. The most interesting observation from these figures is that the maximum percentile reduction in terms of transmitted packets occurs for τ equal to 11ms and not 7ms; for baseline FCFS/DT, the service time at node B At τ = 11ms is approximately 5.5ms (around 6.5ms for the edge nodes), and the witnessed arrival rate at node B is approximately 1 packet per 5.5ms. In other words, τ = 11ms is the equilibrium point of the network when baseline FCFS/DT is used. For smaller values of τ, the queues at node B become unstable and more packets will be dropped by the Drop-Tail policy instead of being forwarded. This naturally equates to less transmissions, and thus less visible effort reduction for Cdg+Rng relative to FCFS/DT. At τ = 11ms, the Drop-Tail effect is lessened, and the number of forwarded packets increases, resulting in more *over-the-air* transmissions. For the Cdg+Rng case, the queues at node B are also stable at τ = 11ms, albeit with a reduced average number of packets when compared to the τ = 7ms equilibrium point at which the peak coding performance is achieved. Nevertheless, eventhough coding opportunities are less for τ equal to 11ms than for τ equal to 7ms, the amount of

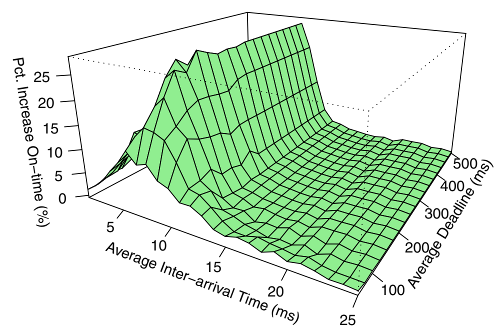

Fig. 7. Percentage increase in on-time packets for coding (Cdg) relative to Rng.

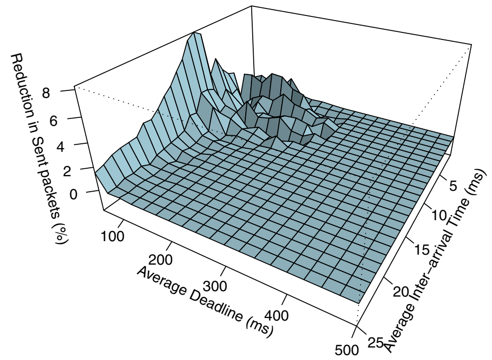

Fig. 8. Percentage reduction in over-the-air packet transmissions for reneging (Rng) relative to FCFS/DT.

coded packets is still considerable. Coupled with the increase in transmissions at node B for the baseline case, the overall result is that the peak relative effort reduction for Cdg+Rng relative to baseline FCFS/DT is witnessed for τ equal to 11ms.

VI. Conclusions

In this paper we proposed and investigated the use of opportunistic network coding and service reneging for improving the real-time performance in sensor networks. Our results show that both coding and reneging, and especially their combination, can be effective techniques that significantly increase the number of on-time received packets. The actual on-time gain depends on both the network load and the timing constraints of packets. Nevertheless, unlike service reneging, network coding is able to maintain a constant gain for increasing deadline values when the network load is in the equilibrium region. In addition to the novel approach that looks at network coding from a real-time perspective and investigates coding on top of IEEE 802.15.4 networks, this work opens the door to a number of new directions. More specifically, there is a need for analytical queuing models that characterize the performance of service reneging and network coding in general networks. In the domain of real-time scheduling, a comparative analysis between different scheduling algorithms in the presence of coding and reneging would be an interesting step forward in the topic, especially in multi-flow and multi-hop scenarios.

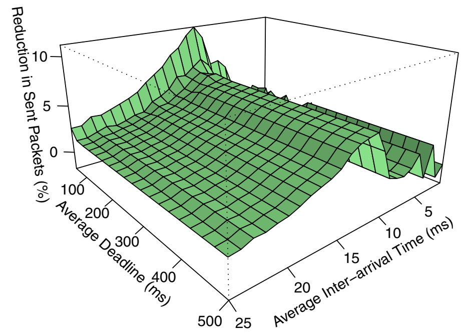

Fig. 9. Percentage reduction in over-the-air packet transmissions for coding (Cdg) relative to FCFS/DT.

References

[1] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," in *IEEE Real-Time Systems Symposium*, Austin, TX, USA, December 2002, pp. 39–48.

[2] J. Francomme, G. Mercier, and T. Val, "A simple method for guaranteed deadline of periodic messages in 802.15.4 cluster cells for control automation applications," in *11th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, Prague, Czech Republic, December 2006, pp. 270–277.

[3] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "Speed: a stateless protocol for real-time communication in sensor networks," in *23rd International Conference on Distributed Computing Systems*, Providence, RI, USA, May 2003, pp. 46–55.

[4] E. Felemban, C.-G. Lee, and E. Ekici, "Mmspeed: Multipath multi-speed protocol for qos guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.

[5] K. Karenos and V. Kalogeraki, "Real-time traffic management in sensor networks," in *27th IEEE Real-Time Systems Symposium*, Rio de Janeiro, Brazil, December 2006, pp. 422–434.

[6] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He, "Rap: A real-time communication architecture for large-scale wireless sensor networks," in *8th IEEE Real-Time and Embedded Technology and Applications Symposium*, San Jose, CA, USA, September 2002, pp. 55–66.

[7] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in *14th IEEE International Workshop on Quality of Service*, New Haven, CT, USA, June 2006, pp. 83–92.

[8] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[9] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Médard, "The importance of being opportunistic: Practical network coding for wireless environments," in *Allerton Conference*, Monticello, IL, September 2005.

[10] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *SIGCOMM*, Pisa, Italy, September 2006.

[11] A. Argyriou, "Wireless network coding with improved opportunistic listening," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 2014–2023, April 2009.

[12] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge Uviversity Press, 2005.

[13] "Ieee std 802.15.4-2996, september, part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans), ieee," 2006.

[14] B. Walke, S. Mangold, and L. Berlemann, *IEEE 802.11 Wireless Systems*. John Wiley and Sons, 2006.

[15] A. Varga, "The omnet++ discrete event simulation system," in *15th European Simulation Multiconference (ESM'2001)*, Prague, Czech Republic, June 2001.