Performance Evaluation of Network Coding and Packet Skipping in IEEE 802.15.4-Based Real-Time Wireless Sensor Networks

Marc Aoun[1], Antonios Argyriou[2,3], and Peter van der Stok[1]

[1] Philips Research, High Tech Campus 34, 5656AE, Eindhoven, The Netherlands
{marc.aoun,peter.van.der.stok}@philips.com
[2] Department of Electrical and Computer Engineering,
University of Thessaly, Greece
[3] Center for Research and Technology Hellas (CERTH), Thessaloniki, Greece
anargyr@gmail.com

Abstract. In a number of application domains, the volatility of the monitored environment where Wireless Sensor Networks (WSNs) operate engenders timing constraints on the generation, processing, and communication of sensory data. In this paper, we primarily focus on the use of an information theoretic approach, namely network coding, to improve the on-time delivery of messages in IEEE 802.15.4-compliant networks. We further study the real-time gain that packet skipping can provide on its own and in combination with network coding. Subsequently, we investigate the potential benefits of introducing deadline-awareness into the coding mechanism through the use of an Earliest Deadline First (EDF) - based scheduling policy. Simulation results show that network coding and packet skipping are by themselves effective techniques to increase on-time goodput, with additional gain obtained when the two techniques are used concurrently. Our results further provide insight into the performance of deadline-aware scheduling in a network coding environment.

1 Introduction

The advent of wireless sensor networking is shaping a future where Ubiquitous Computing and Ambient Intelligence become a reality. 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 to report the state of such a 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snapshot* made through sensing by a deployed WSN remains valid for a limited amount of time, after which it becomes outdated and its underlying sensory data obsolete.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 state 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 patient's situation. 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 would practically mean that different data rates and deadline requirements will have to be dealt with by the same device. Hence the importance of studying real-time aspects under different load conditions.

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 meeting deadlines when 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) under different network loads, and the elimination of obsolete information in order to avoid false conclusions and reduce useless overhead.

In this paper we investigate a novel approach for improving real-time performance in sensor networks by considering recent information-theoretic advances. More specifically, we explore the idea of network coding [1] where routing elements in a network execute algebraic coding operations on packets besides simply forwarding them. Network coding increases the information content per packet transmission without incurring significant overhead [13]. In this paper we leverage this advantage that network coding provides in order to expedite the transmission of real-time packets through the network. Although we explore the above characteristic in a simple network topology, the presented results are transferable to more complex networks. The reason is that in generic multi-hop topologies, router nodes are the meeting point of packets coming from different neighboring nodes and are responsible of forwarding them towards their destinations. A router node can take advantage of the communication pattern between its neighbors and the broadcast nature of the wireless channel; instead of forwarding each packet separately, the router can opportunistically combine two packets through an algebraic XOR operation, and broadcast one resulting coded packet instead of two packets. Provided that nodes store copies of packets they have recently transmitted, the two concerned direct neighbors of the router can decode (through a XOR operation) the coded packet and retrieve the information content relevant to them. Fig. 1(a) demonstrates the basic concept, where node A wants to communicate packet 1 to node B, and C is communicating packet 2 to node D, with node R being a common routing point for both flows. The following communication sequence takes place: Node A broadcasts packet 1 and C broadcasts packet 2. nodes M and N rebroadcast packet 2 and packet 1, respectively. Node R codes the two packets (1 XOR 2) and broadcasts the resulting coded packet. Node M retrieves packet 1 by decoding the received packet ([1 XOR 2] XOR 2) and forwards it to node B. Node N performs a similar decoding operation ([1 XOR 2] XOR 1) and delivers packet 2 to node D.

Our contribution in this paper is multi-fold: 1) We investigate network coding from a real-time perspective, i.e. by assessing its impact on the delivery of

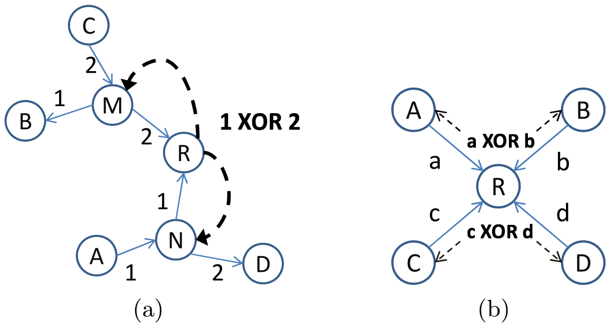


Fig. 1. Network coding of two and four packet flows

real-time data (goodput) that are associated with *per-packet deadlines*. This is in contrast to previous non-sensor networking work where coding was primarily employed with the purpose of improving throughput and delay optimization at the flow-level. 2) We investigate the impact that packet skipping (i.e. proactive message removal from transmission queues) can have on timeliness. 3) We investigate the joint real-time performance of the combination of coding and packet skipping, and assess their relative contributions for different loads and different timing requirements. 4) We provide insight into the performance of deadline-aware packet scheduling in a network coding environment, relative to the widely adopted and simpler to implement First Come First Served (FCFS) scheduling policy.

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. We developed a performance model that characterizes analytically the real-time performance of a single M/M/1 node with network coding in [2]. The complexities we faced in that work forced us to seek a simulation-based study that we present here.

2 Related Work

A fair amount of work has targeted real-time aspects in WSNs. Existing approaches try to address timeliness at specific layers separately [6,9,8,10,12], or with sophisticated cross-layer techniques [7,15]. Caccamo et al. designed an implicit prioritized protocol based on the Earliest Deadline First (EDF) Policy for hard real-time sensor networking [6]. A cellular topology is adopted and channel diversity is used among neighboring cells. Intra-cell communication is achieved by replicating the transmission schedule at all cell nodes and employing EDF to schedule channel access. Inter-cell communication uses a globally synchronized TDMA mechanism. He et al. proposed SPEED [10], a protocol that provides real-time unicast, area-multicast and area-anycast services. It achieves soft real-time end-to-end communication by maintaining a certain velocity for a packet travelling across the network. RAP, a cross-layer real-time communication architecture designed by Lu et al. [15], provides high-level services to applications and introduces the concept of Velocity Monotonic Scheduling (VMS), a packet scheduling policy that takes into account both the deadline of packets and the remaining distance they have to cover to reach their destination. More recently, Aoun and van der Stok [3] provided an in-depth schedulability analysis of real-time periodic messages in an overloaded point-to-point IEEE 802.15.4 setting, and analyzed the performance of different service rejection criteria, identifying the optimal strategies for different operational conditions. The idea of network coding has been applied in IEEE 802.11-based multi-hop wireless networks and throughput benefits in the order of 3-4 times over baseline 802.11 have been demonstrated for bulk data transfers [13,14,4]. 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.

3 System Model

The system under consideration is an IEEE 802.15.4 mesh network and is depicted in Fig. 1(b). The four nodes A, B, C, and D, referred to as source nodes, communicate with each other in pairs, acting concurrently as sources and sinks of information. The 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 [17]. Our topology choice of having the sources as direct neighbors of the router R is motivated by a need to reduce the complexity of the analysis, while maintaining representative results and shedding light on the fundamental performance of network coding.

Source nodes maintain two queues; a transmission queue (TxQ) and a decoding queue (DcQ). The TxQ holds messages that are generated by 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 n reception queues RxQ_i. A reception queue RxQ_i is associated with source node i and functions as forwarding queue; it stores messages that were received from that source node and need to be forwarded to its counterpart node. All the aforementioned queues are implemented at the network layer. No modifications are introduced to the IEEE 802.15.4 MAC.

3.1 IEEE 802.15.4

The IEEE 802.15.4 MAC/PHY standard [11] is the most popular standard for low data rate wireless PANs. The standard specifies four physical layers (PHY) with three of them being based on direct sequence spread spectrum (DSSS) techniques. We consider 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 adopt the beaconless mode.

Three configurable attributes define the functioning of non-slotted CSMA/CA Medium Access Control: *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 320μs), 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 (we assume broadcast transmissions without ACKs). As a consequence of the CSMA/CA mechanism, the time it takes to service a packet is variable.

4 Coding, Scheduling, and Skipping Mechanisms

Three mechanisms were implemented on top of the 802.15.4 MAC layer and are described below. Note that all these mechanisms are closely related with respect to their functionality and their impact on performance. In this paper we assume that new packets with firm deadlines are generated at the application layer of the source nodes. A generated packet is forwarded to the network layer and is subsequently presented for admission at the *TxQ*. If the queue is full, the packet is discarded (Drop-Tail policy). Otherwise, it is admitted. Similar admission behavior takes place at the router node for the *RxQ*s.

4.1 Packet Coding

Digital network coding, when enabled, occurs at the router node when both RxQs belonging to a pair of inter-communicating source nodes are populated, and the MAC layer is ready to service a packet. The payloads of the Head-of-Line (HOL) packets of the two queues are algebraically coded with each other, resulting in a coded packet. The sequence number of both packets and the IDs of the two source nodes are appended to the payload, resulting in a minor extra overhead of 4 bytes. Similarly to the Sliding Window Protocol, we assume periodic reuse of the available range of sequence numbers. In case only one RxQ is populated, its HOL packet is given to the MAC for servicing. In that sense, network coding is opportunistic in that it exploits the coincidental presence of two packets that can be coded. In essence, a synchronous presence is not required per se, but is leveraged when it occurs.

When a coded packet is received at a source node, the latter will first check whether it is one of the two intended recipients. It will subsequently check the appended sequence numbers and search its DcQ for a copy of its native packet used in the coding process. In case a copy is found, the payloads of the coded packet and the copy are XORed with each other, resulting in the payload intended for the source node. Otherwise, the decoding process fails and the packet is considered as lost.

4.2 Scheduling and Queue Management

The TxQs at the source nodes 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. The scheduling of packets at the router operates as follows: 1) A "horizontal" FCFS mechanism orders the packets in each RxQ in increasing order of queue admission time, 2) A "vertical" FCFS mechanism decides which RxQ (two of them when network coding is enabled) to serve next. This "vertical" FCFS is greedy: It does not consider a combination of admission times for two packets that can be coded. It bases its scheduling decision on one admission time (the smallest). In case no coding opportunity exists, the native FCFS chosen packet is passed directly to the MAC. The concept of the two-dimensional FCFS scheduling that we just described is illustrated in Fig. 2. To assess the impact of deadline-awareness in the coding process, we shall also investigate at a later stage in this paper the performance of an Earliest Deadline First (EDF)-based scheduling policy (both horizontal and vertical) at the router node.

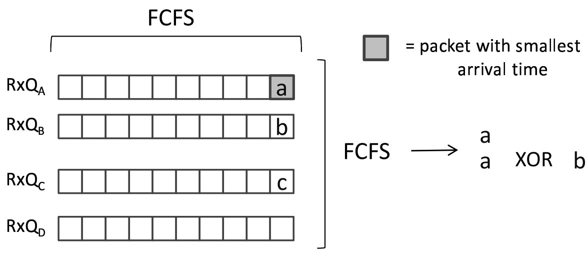


Fig. 2. Two-dimensional scheduling at the router node

The proposed scheduling and coding algorithms, as well as all transmission queues, are implemented at the network layer. No queue build-up happens at the MAC: At all nodes, packets are committed one at a time to the MAC thus allowing full decision control at the network layer over the precise packets that will be served and the packets that will be denied service (skipped). At a source node, a copy of a transmitted packet is stored in the *DcQ*. The copy is removed when the source node overhears R forwarding this packet (in native or coded form) or any other previously committed packet that has: 1) a higher sequence number (in case of FCFS at the router), or 2) a higher sequence number and a bigger absolute deadline (in case EDF is adopted at the router). When the DcQ is full, no copy is stored and the packet is flagged to inform node R that it should not be coded.

4.3 Packet Skipping

Packet skipping refers to the action of denying future service (i.e. transmission) to a packet that is already residing in a transmission queue. The procedure, when activated, proactively cleans the queues upon the arrival of new packets in order to potentially reserve the buffer space for packets that have higher probability of arriving on-time. Packet skipping is executed before coding and before any type of packet is committed to the MAC layer in order to avoid coding and serving packets that are expired or are unlikely to meet their timing constraints.

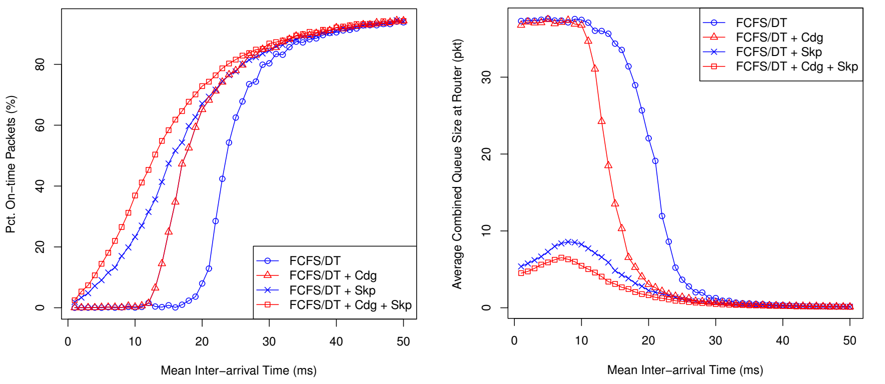
To decide whether to execute packet skipping for a particular packet, an estimate of the MAC service time is used. Estimation is needed due to the variable service time mentioned in section 3.1. The estimate is set to the mean of the service time values of the last 10 serviced packets. To account for the actual 2-hop communication path, source 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 (i.e. the remaining time until deadline expiration) 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.

5 Performance Evaluation

5.1 Simulation Setup

We have built our performance assessment environment in OMNet++ [18], 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 accuracy of the models for timeliness evaluation has been validated by Rousselot et al. [16]. The MAC layer is configured with $macMinBE$=3, $macMaxBE$=5, and $macMaxCSMABackoffs$=4, which correspond to the default values specified in the standard [11]. The network consists of 4 source nodes [A,B,C,D] and a router R, as depicted in Fig. 1(b). The communication pattern is between A and B (flow 1), and between C and D (flow 2).The size of TxQ for each source node is equal to 10 packets and the DcQ has a size equal to 20 packets. Each of the RxQ queues at node R can hold up to 10 packets.

Source nodes generate packets following an exponentially distributed inter-arrival time with mean τ. The smaller τ is, the higher is the generation rate and thus the higher the network load is. Each node generates 10000 messages in total. The value of τ is gradually varied in the simulations to assess the system performance under different traffic loads. Packets have to traverse two



(a) On-time performance (b) Average buffer occupancy at node R

Fig. 3. (a) Percentage of packets received at the source nodes within their deadlines and (b) Average buffer occupancy at node R, for the baseline case, coding (Cdg), skipping (Skp) and the combination of Cdg and Skp

hops (e.g. A-R-B). Plain packets have a physical frame size of 60 bytes and coded packets 64 bytes. Every generated packet has a firm absolute deadline associated with it, after which its information content becomes obsolete. The absolute deadline is equal to the absolute generation time of the packet plus a relative deadline. In the first part of the evaluation, relative deadlines are uniformly distributed over the interval [10ms,100ms], thus with a mean value equal to 55ms and a width of 90ms. In a second phase, we shall vary the average relative deadline in order to assess the performance of coding and skipping for different timeliness requirements. 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 *Skp* refers to packet skipping. Four mechanisms are investigated: the baseline case (FCFS/DT), the baseline case plus network coding (FCFS/DT + Cdg), the baseline case plus packet skipping (FCFS/DT + Skp), and finally, the baseline case plus a combination of coding and skipping (FCFS/DT + Cdg + Skp).

5.2 Case 1: Constant Average Deadline

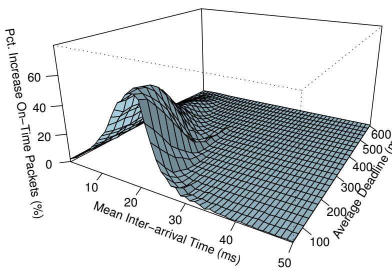
In this section, the relative deadline of generated packets is uniformly distributed between 10ms and 100ms. Fig. 3(a) presents, for all mechanisms, and as a function of τ, the percentage of packets that make it on time to their destination out of all 40000 packets generated in the network. Fig. 3(b) provides the average buffer space occupancy at the router node (total over all RxQs). We recall that the larger τ is, the lighter the network load is.

The results show that the timeliness performance of baseline FCFS/DT is the worst among the four mechanisms, in particular for high loads (1ms $< \tau <$ 30ms). For larger τ values, all methods have similar performance, since the network load is low and the queueing delays are small. Applying algebraic coding at node R provides a gain that reaches up to 57% over the baseline case. This performance improvement can be explained by studying Fig. 3(b). By XOR-ing payloads and servicing two packets concurrently, network coding considerably reduces the overall buffer usage at node R, relative to baseline FCFS/DT. The drop in buffer occupancy is visible in the same operation region where the on-time gain is noticed (10ms $< \tau <$ 30ms). A reduction in buffer space occupancy reduces the number of messages dropped at node R by the Drop-Tail policy. The ability of FCFS/DT + Cdg to service two packets concurrently further reduces the queueing time that packets have to endure at node R before having access to the MAC layer for servicing.

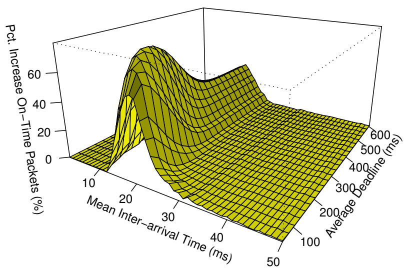
The relative goodput gain is nevertheless negligible for values of τ smaller than 10ms. There are two reasons behind this: 1) For such small inter-arrival times, a bottleneck exists at the source nodes, with a high percentage of packets being dropped at these nodes by the Drop-Tail mechanism or otherwise having to experience significant waiting times in the TxQ queues once admitted (the average queue size at the source nodes was measured to be equal to 9.5 packets) 2) Due to the high packet generation rate, The DcQ at the source nodes become considerably loaded, with a significant amount of sent packets being flagged as not valid for coding. This flagging process reduces the ability of node R to code packets, and its impact is reflected in the average buffer occupancy; as shown in Fig. 3(b), no difference in buffer occupancy between FCFS/DT and FCFS/DT + Cdg is noted for τ smaller than 10ms. When assuming a DcQ of infinite size, a drop of 25% in this same region was noticed.

Packet skipping outperforms both baseline FCFS/DT and simple algebraic coding. By proactively dropping packets that have a high probability of missing their deadlines, skipping favors those packets which are more likely to satisfy their timing requirements. On the other hand, simple FCFS/DT and its network coding variant FCFS/DT + Cdg treat every single packet that is admitted to the transmission queue. Both remain oblivious to the fact that buffering and servicing expired packets and packets with short lead times 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 packet skipping is correlated with a reduction in buffer occupancy, as illustrated in Fig. 3(b). A similar reduction in buffer occupancy at the source nodes was also observed.

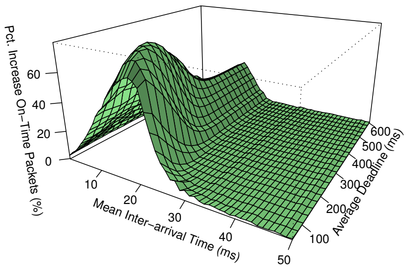
The ability of network coding to simultaneously serve two packets at router R, adds to the efficient buffer space and MAC usage provided by packet skipping. As shown in Fig. 3(a), a combination of coding and skipping performs the best among all tested cases. Applying network coding in addition to proactively cleaning queues reduces the waiting time of packets at node R and reduces further the average queue size at the router (Fig. 3(b)). Overall, applying a combination of network coding and packet skipping can provide up to 65% more on-time packets than traditional FCFS Drop-Tail. Finally, it is also important to mention



(a) FCFS/DT + Skp



(b) FCFS/DT + Cdg



(c) FCFS/DT + Cdg + Skp

Fig. 4. Percentage difference in on-time packets for coding, skipping, and their combination, relative to baseline FCFS/DT

that the higher performance of both skipping and coding is also coupled with a decrease in transmitted messages, and thus comes at less transmission effort.

5.3 Case 2: Varying Average Deadlines and Packet Arrival Rates

In this section we investigate the real-time performance of coding, skipping, and their combination, for different timeliness requirements and different loads. For that purpose, the average relative deadline of generated packets is varied from 55ms to 605ms in steps of 25ms, while maintaining a uniform deadline distribution with a constant width of 90ms. The network load is also varied by varying the mean inter-arrival time of packets at the source nodes from 1ms to 50ms in unit steps. The difference in on-time packets between each of the three methods and baseline FCFS/DT is plotted in Fig. 4. Note that the percentage values are relative to the total 40000 generated packets.

We shall first consider the impact of solely applying packet skipping (i.e. FCFS/DT + Skp). Fig. 4(a) shows a clear performance benefit of skipping for mean inter-arrival times τ between 1ms and an upper limit τ_l. The value τ_l for which performance improvement is still witnessed decreases when the average deadline is increased; the higher the deadline is, the narrower is the range of τ values for which packet skipping outperforms baseline FCFS/DT. Indeed, higher deadlines implies that less packets will get expired or be assessed as outdated, which further implies that less packets will be denied service. The peak increase, that initially reached up to 60% for an average deadline of 55ms, gradually fades away for higher deadlines values, with less than 20% peak improvement left at a deadline of 350ms, and 5% for a deadline of 500ms.

Moving to the case of FCFS/DT + Cdg, a number of performance-related aspects can be deduced from Fig. 4(b). The most striking difference between FCFS/DT + Skp and FCFS/DT + Cdg is that, unlike skipping where the gain gradually decreases for higher deadline values, coding has the ability to maintain a constant gain for large average deadlines, with a constant peak gain of more than 30% for deadlines above 400ms. The ability of coding to reduce buffer occupancy time and thus reduce the number of dropped packets at node R provides this constant gain irrespective of the average deadline. Whereas skipping is able to provide improvement for small τ values (e.g. τ less than 10ms for a 55ms average deadline), coding alone does not deliver gain over baseline FCFS/DT under these high arrival rates, for the same reasons mentioned in section 5.2. The actual lower limit of τ above which coding starts to provide performance gain is deadline-dependent. A further aspect where coding and skipping differ is in the evolution of the peak performance relative to the deadline value; for all values of τ, skipping provides the highest improvement for the lowest considered average deadline (55ms), whereas the performance of coding first increases when the deadline is increased, before reaching a maximum value beyond which it gracefully decreases to the observed constant gain value.

To assess the combination of coding and skipping, we plot in Fig. 4(c) the percentage on-time difference for the combined FCFS/DT + Cdg + Skp approach. We also plot in Fig. 5 the gain difference between FCFS/DT + Cdg + Skp and FCFS/DT + Skp. The general characteristics mentioned for FCFS + Cdg still hold for the combined approach, with one major exception; FCFS/DT + Cdg + Skp provides gain even for the smallest range of τ. Fig. 5 shows that it is the contribution of packet skipping that accounts for this difference. Interestingly, this contribution is shifted to lower values of τ compared to the FCFS/DT + Skp case. This is explained by the fact that network coding operates in the same region as skipping, thus reducing the impact that skipping would otherwise have had if it were applied alone. The impact of skipping remains in the region where coding was not initially providing significant gain, i.e. for the highest network loads.

5.4 Peak Performance and Optimal Operation Point

The results plotted in Fig. 4 indicate that for every considered average deadline, there exists a specific value of τ for which the maximum gain is achieved. This value will be referred to as τ_{max}. We shall try to elucidate the existence of such an optimal τ using FCFS/DT + Cdg as illustrative example. Fig. 6(a) plots the measured value of τ_{max} for every average deadline. It additionally plots the value of τ, referred to as τ_p, for which the maximum increase in delivered throughput

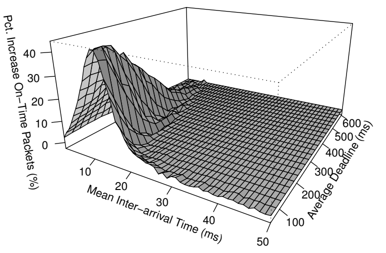
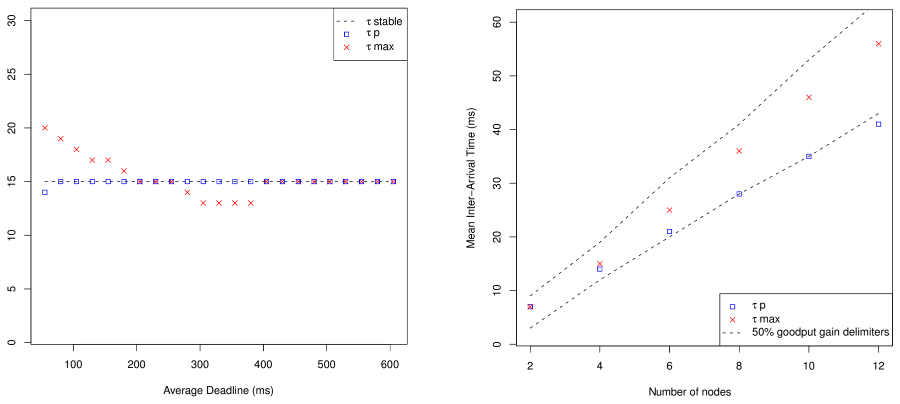


Fig. 5. On-time difference between FCFS/DT + Cdg + Skp and FCFS/DT + Skp

at the sink nodes occurs (regardless of whether the packets made it within their deadline or not). Finally, the stability point τ_{stable} of the queueing system at the router node is also plotted. Stability, from a queueing theory perspective, is the operation regime where a queueing system reaches and maintains an equilibrium state, i.e. when the arrival rate of packets at the queue is smaller or equal to the average departure rate. The stability point was identified by measuring the average packet arrival rate and average departure rate at node R, and identifying the smallest value of τ for which the ratio of these two rates is equal to 1. Note that node R has the highest load among all nodes in the network. As such, source nodes are also in a stable state at τ_{stable}.

For all deadlines, τ_p remains equal to τ_{stable}, meaning that the relative *throughput* increase is always maximized at the stability point. Indeed, the stability point offers the maximum number of opportunities for coding, while reducing the number of packets dropped by the Drop-Tail mechanism. A major observation is that for deadlines bigger or equal to 400ms, τ_{max} and τ_p are exactly equal to τ_{stable}; for high deadline values, the peak increase in on-time packets relative to FCFS/DT is achieved at the stability point of the network. On the other hand, for deadlines up to 200ms, the goodput is maximized for τ_{max} values bigger than τ_{stable} and τ_p. For such deadlines, even though the relative throughput is still maximized at τ_{stable}, a lower network load is needed to reduce the waiting time of packets in the queues. For deadlines between 300ms and 400ms, the shift between τ_{max} and τ_{stable} is due to the fact that the goodput increase is measured as a difference to the baseline case, which unlike coding, still does not provide any positive gain for this deadline range.

Finally, to present insight into the behavior of the system for higher node densities and loads, Fig. 6(b) plots, for a fixed deadline of 200ms and an increasing number of nodes: τ_p, τ_{max}, and the range of τ for which the provided goodput increase is at least 50% of the increase provided at τ_{max}. An observation we can make here is that the bigger/more loaded the network is, the more τ_{max} diverts from τ_p, and the wider the 50% range of τ values becomes. The aforementioned results validate our position of looking at network coding from a real-time goodput perspective, versus looking at it from a throughput perspective.



(a) Optimal operation point for varying deadlines.

(b) Optimal operation regime vs number of source nodes.

Fig. 6. Peak performance and optimal operation points

5.5 Deadline-Aware Coding Using EDF Scheduling

We have till now assessed the performance of network coding while maintaining the same scheduling policy (FCFS). A yet unexplored dimension is the order with which packets are serviced at the router R. The following section assesses the impact of servicing queues based on the absolute deadlines of packets. Towards that end, FCFS is replaced by an Earliest Deadline First (EDF) variant: The packet P with the smallest absolute deadline among all packets residing in the RxQs is chosen first for service. In case a packet exists in the second RxQ belonging to the same flow, this packet can be opportunistically coded with P. Two scenarios are considered: In the first scenario, the average relative deadline of both flows is increased from 55ms to 605ms. In the second scenario, one flow has a fixed average deadline of 55ms, while the average relative deadline of the second flow is varied from 55ms to 605ms. This scenario is suitable for studying the impact of the vertical scheduling component. Given the insight obtained in the previous sections, we shall restrict ourselves to a comparison between FCFS/DT + Cdg + Skp and EDF/DT + Cdg + Skp.

Fig. 7 conveys the results of the first scenario (on-time difference for EDF relative to FCFS). Little difference is observed. Only minor improvements of 1-2% exists, for very limited operation points. In fact, for τ smaller than 10ms and high deadlines such as 400ms, we observed a decrease between 2-5% for EDF, that we found to be due to a less effective DcQ cleaning in the case of EDF than in the case of FCFS (As mentioned in section 4.2, for FCFS, the sequence number of an overheard packet is enough to clean smaller sequence-numbered packets from the DcQ, whereas for EDF, both the overheard sequence number and the absolute deadline should be bigger than those of a stored packet to safely remove the latter). The obtained result might seem striking at first, given the proven optimality of EDF in real-time systems. In fact, EDF has been proven

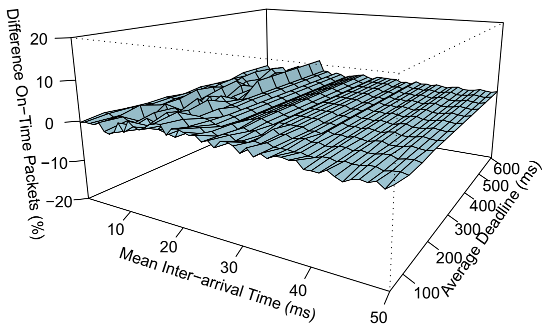
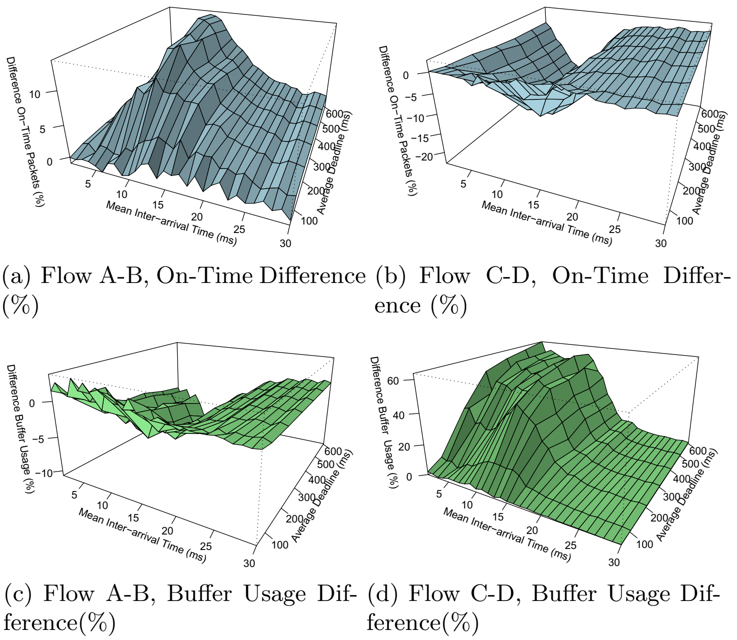


Fig. 7. Deadline-aware coding: On-time difference relative to FCFS

to be optimal *only in the case where all tasks (in our case all packets) can be serviced in an order such that all the deadlines are met*, i.e. if a feasible schedule exists [5]. This is obviously not the case here, because of the load conditions, the limited available buffer space, the stochastic service time introduced by the CSMA/CA mechanism and the transmission failures that occur.



(a) Flow A-B, On-Time Difference (%)

(b) Flow C-D, On-Time Difference (%)

(c) Flow A-B, Buffer Usage Difference(%)

(d) Flow C-D, Buffer Usage Difference(%)

Fig. 8. Per-flow on-time performance and buffer space usage difference between EDF and FCFS

Fig. 8 illustrates, for the second scenario, the per-flow on-time and buffer usage difference of EDF relative to FCFS. By taking the absolute deadlines into account, vertical EDF favors the communication flow A-B since it has, on average, smaller deadlines. This results in an improved on-time performance relative to deadline-oblivious FCFS (Fig. 8(a)) and is correlated with a decrease in the buffer space usage at the router node for the RxQs that hold packets from A and B (Fig. 8(c)). The gain in the A-B flow comes at the expense of the communication flow C-D, as shown in Fig. 8(b). Due to the favoring of flow A-B, the packets of flow C-D experience more waiting time at node R. This reflects itself in an increased buffer occupancy for this flow, as conveyed in Fig. 8(d), which in turn results in a higher dropping rate of packets by the Drop-Tail mechanism.

6 Conclusion

This work is the first to propose and investigate the use of network coding for improving the real-time performance in IEEE 802.15.4-based wireless sensor networks. Our results show that coding, packet skipping, 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 packet skipping, network coding is able to maintain a constant gain for increasing deadline values. 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 models that characterize the performance of packet skipping 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 skipping would be an interesting step forward in the topic, especially for varying load conditions.

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

References

1. Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W.: Network information flow. IEEE Transactions on Information Theory 46(4), 1204–1216 (2000)
2. Aoun, M., Beekhuizen, P., Argyriou, A.: An analytical study of network coding in the presence of real-time messages. In: IEEE International Symposium on Network Coding (NetCod), Toronto, Canada (June 2010)
3. Aoun, M., van der Stok, P.: Overloading an IEEE 802.15.4 point-to-point connection with real-time messages. In: 31st IEEE Real-Time Systems Symposium (RTSS), San Diego, CA, USA, pp. 225–235 (December 2010)

4. Argyriou, A.: Wireless network coding with improved opportunistic listening. IEEE Transactions on Wireless Communications 8(4), 2014–2023 (2009)
5. Buttazzo, G.C.: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. Kluwer Academic Publishers, Dordrecht (2000)
6. Caccamo, M., Zhang, L.Y., Sha, L., Buttazzo, G.: An implicit prioritized access protocol for wireless sensor networks. In: 23rd IEEE Real-Time Systems Symposium (RTSS), Austin, TX, USA, pp. 39–48 (December 2002)
7. Chipara, O., He, Z., Xing, G., Chen, Q., Wang, X., Lu, C., Stankovic, J., Abdelzaher, T.: Real-time power-aware routing in sensor networks. In: 14th IEEE International Workshop on Quality of Service, New Haven, CT, USA, pp. 83–92 (June 2006)
8. Felemban, E., Lee, C.G., Ekici, E.: MMSPEED: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. IEEE Transactions on Mobile Computing 5(6), 738–754 (2006)
9. Francomme, J., Mercier, G., Val, T.: 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, pp. 270–277 (December 2006)
10. He, T., Stankovic, J., Lu, C., Abdelzaher, T.: SPEED: a stateless protocol for real-time communication in sensor networks. In: 23rd International Conference on Distributed Computing Systems, Providence, RI, USA, pp. 46–55 (May 2003)
11. IEEE Std 802.15.4-2996, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE (September 2006)
12. Karenos, K., Kalogeraki, V.: Real-time traffic management in sensor networks. In: 27th IEEE Real-Time Systems Symposium, Rio de Janeiro, Brazil, pp. 422–434 (December 2006)
13. Katti, S., Katabi, D., Hu, W., Rahul, H., Medard, M.: The importance of being opportunistic: Practical network coding for wireless environments. In: Allerton Conference, Monticello, IL, USA (September 2005)
14. Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M., Crowcroft, J.: XORs in the air: Practical wireless network coding. In: SIGCOMM, Pisa, Italy (September 2006)
15. Lu, C., Blum, B., Abdelzaher, T., Stankovic, J., He, T.: 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, pp. 55–66 (September 2002)
16. Rousselot, J., Decotignie, J.D., Aoun, M., van der Stok, P., Oliver, R.S., Fohler, G.: Accurate timeliness simulations for real-time wireless sensor networks. In: 3rd UKSim European Symposium on Computer Modeling and Simulation, Athens, Greece (November 2009)
17. Tse, D., Viswanath, P.: Fundamentals of Wireless Communication. Cambridge University Press, Cambridge (2005)
18. Varga, A.: The OMNeT++ discrete event simulation system. In: 15th European Simulation Multiconference (ESM 2001), Prague, Czech Republic (June 2001)