Mechanisms to Enforce Dependability and Timeliness in Wireless Communications

Jeferson L. R. Souza, **Ricardo C. Pinto**, and **José Rufino**
Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Lisboa, Portugal
LaSIGE - Navigators Research Team
Email(s): jsouza@lasige.di.fc.ul.pt, ricardo.pinto@ciencias.ulisboa.pt, jmrufino@ciencias.ulisboa.pt

***Abstract*—The use of wireless networks to effectively support hard real-time communications in the extremely harsh conditions that avionics and aerospace systems and applications face aboard (unmanned) autonomous vehicles is still an open issue.**

This paper starts by defining and exploiting the properties of the Wireless network Segment (WnS), a broadcast space where all communicating devices are at one-hop distance of each other. Then we discuss how to exploit and secure the properties defined for the WnS and how those can be used to support low level (frame) error monitoring and failure detection functions and, more importantly, how such mechanisms can be used as building blocks to architect, above the Wns, Resilient Hard Real-Time protocols and services, such as reliable communication, node failure detection and membership, clock synchronisation and a global notion of time.

Such kind of protocols and services are extremely helpful to build wireless-based distributed applications, and the mechanisms and properties one introduce in the WnS definition constitute a first step towards the provision of similar guarantees in multi-hop interconnected WnS networks.

***Index Terms*—network architectures; real-time; dependability; timeliness; wireless communications; wireless sensor and actuator networks**

I. Introduction

The use of wireless networks in avionics and aerospace environments has been seen as a must to reduce the Size, Weight, and Power (SWaP) consumption, being at the same time essential to support critical intra-vehicular communication of on-board devices, such as sensors and actuators. Advances in microelectronics have enabled the development of small but extremely powerful low-power sensor and actuators devices, constituting the basis for the establishment of novel distributed real-time embedded wireless networks with potential for monitoring the whole environment and performing control functions. Wireless communications assume also a fundamental role for the communication and cooperation of distributed and mobile entities in the execution of complex tasks. This is the case of small satellite clusters flying in formation missions, as well as robotic teams for planetary exploration.

In a distributed hard real-time environment, where communications must be, at the same time, correct, dependable, and timely, the presence of errors is extremely challenging in the design, verification, and validation of the real-time guarantees offered by the communication platform. Theoretical and schedulability analysis often disregard the impact of low level (frame) errors in the operation of the whole networking protocol stack, namely in the analysis and provision of end-to-end real-time bounds [1], [2], [3], that ultimately can lead to the violation of such bounds and therefore to the failure of the whole system. The provision of hard real-time bounds within one-hop wireless communications is a first step towards the provision of end-to-end guarantees in multi-hop environments [4].

In this paper we explore how the design of simple but fundamental low levels extensions constitute the basis for the specification of robust, reliable, and timely communication protocols for wireless networks. We believe that these new mechanisms will be of fundamental importance to design and develop highly effective Resilient Hard Real-Time protocols and services, such as reliable communication, node failure detection and membership, clock synchronisation and global notion of time [5].

To present our contributions this paper is organised as follows: Section II presents the detailed description of an one-hop abstract communication model dubbed Wireless network Segment (WnS), with suitable and useful properties for the provision of dependable real-time guarantees; Section III presents the basis for frame error monitoring and failure detection; Section IV uses the previous mechanisms to advance towards the provision of Resilient Hard Real-Time Communication services, above the WnS; finally, Section V presents the conclusions and future research directions.

II. System Model

All networking communications described in this paper are performed within the scope of a physical and data link layer abstract networking model, dubbed Wireless network Segment (WnS). The WnS is a broadcast space where the networking nodes are able to communicate directly and sense transmission from each other (one-hop distance). This simple approach allows to achieve a first result: to exploit the broadcast nature of the shared wireless transmission medium, throughout which all networking nodes communicate. The terms networking node, wireless node, or simply node, are used interchangeably in this paper.

This work was partially supported by the EC, through project IST-FP7-STREP-288195 (KARYON); by FCT/DAAD, through the transnational cooperation project PROPHECY; and by FCT, through project PTDC/EEI-SCR/3200/2012 (READAPT), through LaSIGE Strategic Project PEst-OE/EEI UI0408/2014, and Individual Doctoral Grant SFRH/BD/72005/2010.

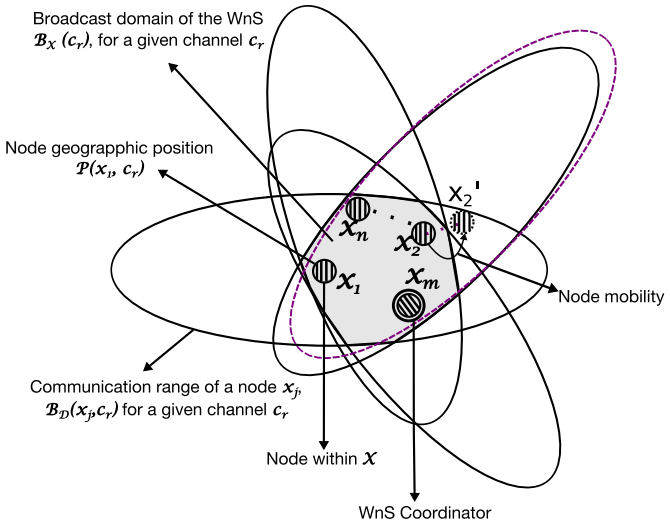


Fig. 1: A representation of the WnS abstraction

A. Wireless network Segment (WnS) abstraction

The utilisation of the WnS communication model, introduced in [5] and updated and extended in [4], [6], [7], can be formally expressed by an elementary tuple $WnS = \langle X, x_m, C \rangle$, where X is the set of the wireless nodes members of the WnS; x_m is the WnS coordinator, $x_m \in X$; C represents a set of non-overlapping radio frequency communication channels.

The set of WnS members is defined by $X = \{x_1, \ldots, x_n\}$, where the cardinality $\#X$ represents the number of nodes within the WnS. All communications are performed through a set of communication channels, $C = \{c_1, \ldots, c_z\}$, where each $c_r \in C$ is a unique channel, being $1 \leq r \leq z$. In case of a WnS using only one channel, $z = r = 1$, i.e., $\#C = 1$.

Every node $x_j \in X$ is included in the set of recipients of each transmitted frame, for each channel $c_r \in C$. The broadcast domain of the WnS, for a given channel $c_r \in C$, is defined by: $B_X(c_r) = \bigcap_{j=1}^{\#X} B_D(x_j, c_r), \ \forall x_j \in X$, where $B_D(x_j, c_r)$ is a geographic region that represents the communication range of a node x_j for a given channel c_r. Figure 1 illustrates a two-dimension representation of the WnS, where the ellipses roughly characterise the radiation diagram and the communication range of nodes for a given channel $c_r \in C$, being the grey area the characterisation of the broadcast domain for a given WnS. In practice, the radiation diagram of a node assumes irregular and complex forms [8].

Furthermore, let $P(x_j, c_r)$ represent the geographic position of node x_j transmitting on channel c_r. A node $x_j \in X$ if, and only if, $\exists c_r \in C$ where $P(x_j, c_r) \subseteq B_X(c_r)$. Otherwise, a node $x_j \notin X$ if, and only if, $\forall c_r \in C, P(x_j, c_r) \nsubseteq B_X(c_r)$. In Fig. 1, due to node mobility, node x_2 changes its geographical position from $P(x_2, c_r)$ to $P(x_2', c_r)$, located outside the broadcast domain of the WnS and puts itself into a situation, where it is within the communication range of all nodes in X but x_m, the WnS coordinator. Being unable to receive frames from x_m, node x_2 will be considered failed and therefore no longer member of the WnS.

B. Fault model

Networking components (e.g., a channel $c_r \in C$, or a node $x_j \in X$) either behave correctly or fail upon exceeding a given number of consecutive omissions (the component's *omission degree bound*), f_o, under a given observation criteria (e.g., the duration of a given protocol execution, $\mathcal{T}_{rd}$). Omission faults may be inconsistent (i.e., not observed by all recipients).

In the context of networking communications, we define an omission as an error that destroys a frame. In this sense, errors derived from the presence of accidental faults are transformed into omissions, which are accounted for the purpose of monitoring networking components at different levels. For each received frame, every node $x_j \in X$ locally accounts the observed omissions. Consecutive erroneous frames received from the same channel input, i.e. a given source node $x_q \in X$, imply the signalling of a *node persistent failure* if exceeding a given bound, k_p; a *node crash failure* is signalled if no traffic is received from node x_q during an observation period, bounded by a given number of consecutive elementary monitoring intervals, k_c, each of limited duration, e.g., $\mathcal{T}_{rd}$.

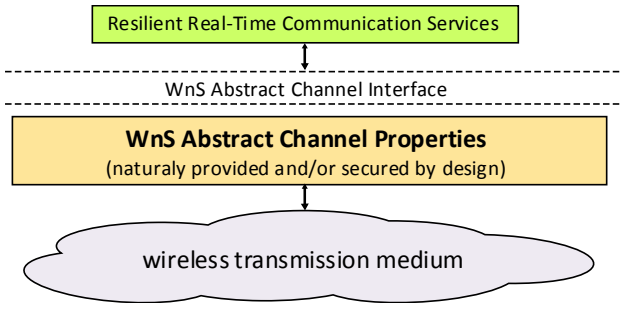
Despite of their importance we are not considering intentional faults, being such topic addressed in future work.

C. WnS abstract channel properties

The characteristics of the low level layers in the wireless networking protocol stack can be abstracted by a set of correctness, dependability, and timeliness properties, which are in essence independent of each particular networking technology. In our WnS abstraction such properties are offered through the facet of an abstract single communication channel we dubbed WnS abstract channel, as illustrated in Fig. 2.

Property WnS1 (*Broadcast*) formalises that it is physically impossible for a node $x_j \in X$ to send conflicting information (in the same broadcast) to different nodes [9], within the broadcast domain of the WnS, $B_X(c_r)$, for a given channel $c_r \in C$.

Property WnS2 (*Error Detection*) has both detection and signalling facets; the detection facet, traditionally provided by classical MAC sublayers, derives directly from frame protection through a frame check sequence (FCS) mechanism, which most utilised algorithm is the cyclic redundancy check (CRC); the signalling facet is provided by the FCS extension introduced in [7], which is able to signal omissions detected in frames received with errors. No fundamental modifications are needed to the wireless MAC standards, such as IEEE 802.15.4 [10]. The use of such unconventional extension is enabled by emerging controller technology, such as reprogrammable technology and/or open core MAC sublayer solutions, such as the transceivers and the MAC sublayers developed by ATMEL [11]. With the CRC polynomials used in wireless MAC sublayers, the residual probability of undetected frame errors is negligible [12], [13].



WnS1 - *Broadcast*: correct nodes, receiving an uncorrupted frame transmission, receive the same frame;

WnS2 - *Error Detection*: correct nodes *detect and signal* any corruption done during frame transmissions in a locally received frame;

WnS3 - *Bounded Omission Degree*: in a known time interval $\mathcal{T}_{rd}$, omission failures may occur in at most k transmissions;

WnS4 - *Bounded Inaccessibility*: in a known time interval $\mathcal{T}_{rd}$, a wireless network segment may be inaccessible at most i times, with a total duration of at most $\mathcal{T}_{ina}$;

WnS5 - *Bounded Transmission Delay*: any frame transmission request is transmitted on the wireless network segment, within a bounded delay $\mathcal{T}_{td} + \mathcal{T}_{ina}$.

Fig. 2: WnS abstract channel properties

Property WnS3 (*Bounded Omission Degree*) formalises for a channel, $c_r \in C$, the failure semantics introduced earlier in the fault model definition, being the abstract channel omission degree bound, $k \geq f_o$. The omission degree of a WnS abstract channel can be bounded, given the error characteristics of its wireless transmission medium [13], [14], [15].

The *Bounded Omission Degree* property is one of the most complex properties to secure in wireless communications. Securing this property with optimal values and with a high degree of dependability coverage may require the use of multiple channels. In [7], we have advanced on how this can be achieved by monitoring channel omission errors, and switch between channels upon detecting that the channel omission degree bound has been exceeded.

The time domain behaviour of a WnS is described by the remaining properties. Property WnS5 (*Bounded Transmission Delay*) specifies a maximum frame transmission delay, which is $\mathcal{T}_{td}$ in the absence of faults. The value of $\mathcal{T}_{td}$ includes the medium access and transmission delays and it depends on message latency class and overall offered load bounds [16], [17]. The value of $\mathcal{T}_{td}$ does not include the effects of omission errors. In particular, $\mathcal{T}_{td}$ does not account for possible frame retransmissions. However, $\mathcal{T}_{td}$ may include extra delays resulting from longer WnS access delays derived from subtle side-effects caused by the occurrence of periods of network inaccessibility [15]. The use of deterministic networking access protocols such as [3], [18] is fundamental to secure a bounded $\mathcal{T}_{td}$ value.

A period of network inaccessibility is a disturbance that may be induced externally by electromagnetic interference, or by glitches in the MAC sublayer operation, such as those that may result from the omission of a MAC control frame (e.g., beacon). The network cannot be considered failed; it only enters into a temporary state where the communication service is not provided to some or all of the nodes. Hence, nodes may experience a loss of connectivity within a WnS; the transient loss of connectivity due to heavy node mobility is also treated under the inaccessibility model. The bounded transmission delay includes $\mathcal{T}_{ina}$, as a corrective term that accounts for the worst-case duration of inaccessibility glitches, given the bounds specified by property WnS4 (*Bounded Inaccessibility*). The inaccessibility bounds depend on, and can be predicted by the analysis of MAC sublayer characteristics [15].

III. Low Level Dependability Enhancements

To secure WnS abstract channel properties some mechanisms were designed and introduced in network operation.

A. Securing WnS2: frame error detection and signalling

The frame check sequence (FCS) is an error-detection code appended to the frame content, and utilised to verify that its integrity was not compromised during the propagation through the communication channel. When the FCS of a received frame is checked, and an error is detected, the (traditional) MAC sublayer silently discards the erroneous frame. This is a very restrictive operational model.

If, in addition to a robust frame error detection, one will have the ability to signal the occurrence of such errors, that will open an whole set of possibilities in terms of error handling capabilities. Thus, to overcome the limitations imposed by the standard design of the MAC sublayer, we have proposed in [7] an extension to the traditional FCS mechanism, which generates a notification upon the reception of each frame (including those received with errors). This mechanism has already been successfully applied in [7] to design advanced channel monitoring, failure detection and channel switching upon failure functions.

The extension to the standard MAC sublayer is highly effective, though extremely simple, as show in the central section of Fig. 3: the FCS extension includes only the extraction of the frame's header and its signalling, together with the frame error status, at the WnS abstract channel interface. This can be easily implemented using currently available commercial off-the-shelf (COTS) technology, such as the ATMEL transceiver [11].

B. Securing frame header integrity

The content of an erroneous frame cannot be used since one cannot know which part of the frame was corrupted. However, some parts of the frame, such as the frame header, may provide relevant information concerning network and/or protocol operation. Thus, it will be extremely interesting if one have the ability to extract such information, even when the FCS verification reports the frame is erroneous. To guarantee this extract action provides correct results, the information to be extracted needs to be protected with a new integrity verification code, herein dubbed Header Check Sequence (HCS)

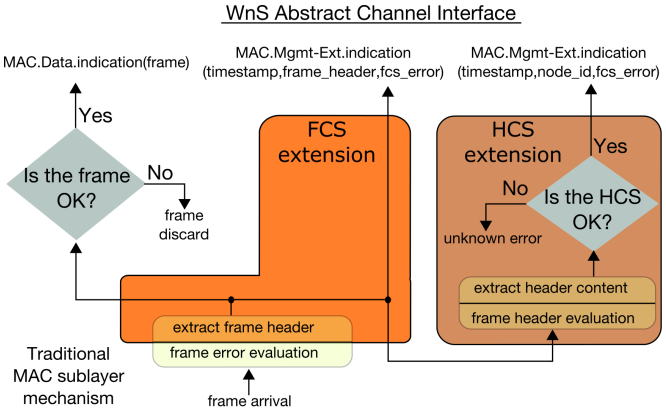


Fig. 3: MAC sub-layer FCS and HCS extensions

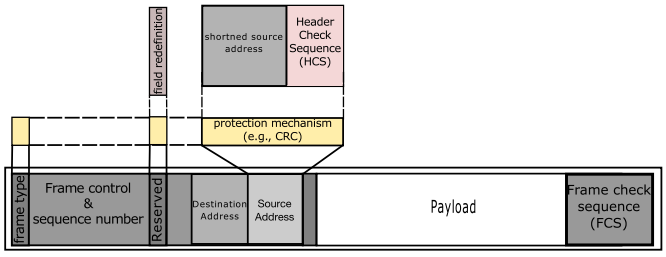


Fig. 4: Data frame, with the header structure defined in [4]

— see Fig. 3 and 4. The integrity of the information being extracted from the frame header content can now be verified and, if correct, delivered at the WnS abstract channel interface, through a management notification primitive, as illustrated in the rightmost part of Fig. 3.

In [4] we have successfully applied this technique to extract from the frame header: the frame type (directly identifying the header structure), a (redefined) reserved field (related with the protocol family using the frame) and a shortened version of the source node address. As illustrated in Fig. 4, in the place of the source address a compound data is stored, consisting in such shortened version of the source node address plus the HCS, which protects the frame type, the relevant reserved fields, and the (shortened) source address, altogether. The standard length of the source address field is unchanged, implying no extra overhead to the frame length [4], [10]. The use of this additional protection mechanism is started after receiving a short address assigned by the WnS coordinator, as specified in [4]. The usefulness of this mechanism consists in identifying the node at the channel input as possibly being the source of the error [4], [7].

C. Failure detectors

The low level mechanisms we have just introduced enable the monitoring of the different networking components: the channel $c_r \in C$, with an omission degree bound, k; a node at a given channel input, $x_q \in X$, to be reported as suffering from a persistent failure, if it exhibits more than k_p omissions in addition to possible omissions, due to channel operation; a node crash failure when no activity is detected from a node, during an entire observation period, multiple of a relevant elementary time interval. The node crash failure observation period has been set to $k_c \times (\mathcal{T}_{td} + \mathcal{T}_{ina})$, given the WnS timeliness properties described in Fig. 2, and the operation specified in [4].

Table I summarises the main characteristics (omission degree bounds and failure detection latencies) of the three failure detectors we have just enumerated. A detailed description of the operation of these failure detectors can be found in [4].

IV. Design of Reliable Protocols

This section discusses how the low level MAC mechanisms introduced earlier are useful in the design of Resilient Real-Time Communication services, such reliable message delivery (unicast, multicast, broadcast), to be built on top of the WnS abstract channel interface, as illustrated in Fig. 2. A first approach to the problem may consist of simply repeating the transmission of a message (data frame) $k+1$ times, being k the channel omission degree. However, such a solution will exhibit a very high overhead in terms of network bandwidth utilisation, and therefore is not considered in our analysis. The alternative is the use of transmit with response protocols.

A. P-ACK: the positive acknowledgement protocol

The traditional approach to the design of transmit with response protocols is the use of positive acknowledgements. Upon the reception of a correct data frame, correct recipients respond with a positive acknowledgement (P-ACK). If a response is missing from some recipient, a new round is started, upon the expiration of a timer, with the retransmission of the data frame. The foundation of such protocol operation is illustrated in Fig. 5.

The P-ACK protocol exhibits a significant bandwidth utilisation overhead, due to the transmission of positive acknowledgements, namely if the number of recipients (n_{list}) is high. The protocol allows the detection of recipients persistent/crash failures and the corresponding intra-protocol recipients list update. However, this action has, in the worst case, a higher latency than the corresponding inter-protocol recipients list

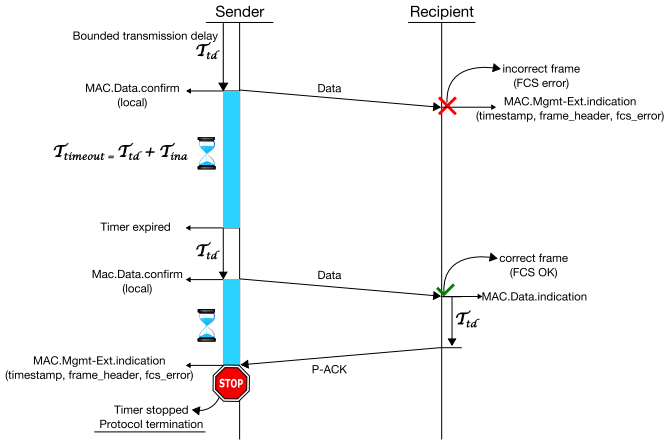


Fig. 5: Sketch of P-ACK protocol operation

Failure detector	Omission degree bound	Best-case failure detection latency	Worst-case failure detection latency
channel	k	$\mathcal{T}_{td}$	$(k+1) \times \mathcal{T}_{td} + \mathcal{T}_{ina}$
persistent	$k + k_p$	$(k + k_p + 1) \times \mathcal{T}_{td}$	$(k + k_p + 1) \times \mathcal{T}_{td} + \mathcal{T}_{ina}$
crash	k_c	$k_c \times (\mathcal{T}_{td} + \mathcal{T}_{ina})$	

TABLE I: Summary of failure detector characteristics

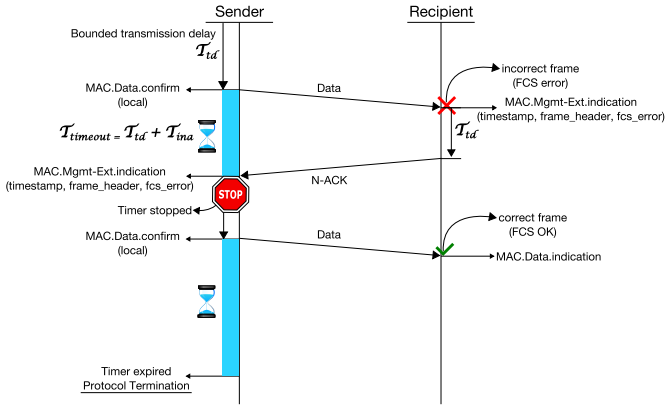


Fig. 6: Sketch of N-ACK protocol operation

update to be performed through the failure detectors specified in [4] and introduced in Section III-C. With this exception, the design of the P-ACK protocol, is rather classic and does not take a real advantage of the specific omission monitoring and failure detection mechanisms introduced in the WnS design. This situation completely changes with the protocol described next, in Section IV-B.

B. N-ACK: the negative acknowledgement protocol

Taking advantage of the frame error signalling management indications, provided at the WnS abstract channel interface, the N-ACK protocol, uses negative acknowledgements: the sender transmits a data frame; correct recipients receiving a correct data frame, take no action; correct recipients receiving an incorrect data frame, signal this event through a management indication, which leads to the transmission of a negative acknowledgement (N-ACK). This operation is illustrated in Fig. 6. The transmission of N-ACK frames is monitored by the sender and, if needed, may also be monitored by recipients.

The N-ACK protocol takes real advantage of the low level mechanisms introduced in Section III. In the absence of frame errors, the N-ACK protocol guarantees reliable message delivery without any network bandwidth utilisation overhead, as illustrated in the second round of the protocol operation sketched in Fig. 6. If no N-ACK is issued, a correct frame has been received by all correct recipients. That means, a totally ordered message broadcast has been successfully completed.

C. Protocol comparison and discussion

The Table II compares a series of characteristics for both P-ACK and N-ACK protocols, among which are the protocol overhead of network bandwidth utilization, the best and worst case protocol termination times, and the worst case intra-protocol recipient failure detection latency. Both protocols may benefit from the interaction with companion failure detectors [4], which allow a faster and accurate update of the recipients list, via inter-protocol updates.

Special inaccessibility control techniques, applicable both to P-ACK and N-ACK protocols, and described in [5], allow an optimal and adptative dimensioning of protocol timers. Timers are always started with the optimal timeout value, $\mathcal{T}_{td}$, in the absence of inaccessibility; in the presence of inaccessibility events, timers are automatically and dynamically extented with t_{ina}, the actual duration of the inaccessibility event, instead of the worst case inaccessibility bound, $\mathcal{T}_{ina}$. Thus, the failure detection latencies of Table I and the protocol execution times inscribed in Table II can be significantly reduced.

V. Conclusion

This paper addressed the difficult problem of providing resilient real-time communications in wireless networks. This problem is even harder in extremely harsh environments with respect to electromagnetic disturbances, such as those found in avionics and aerospace systems and applications.

The approach taken exploits a fundamental set of correctness, dependability and timeliness properties, naturally exploited or imposed by design in the Wireless network Segment (WnS), a broadcast space where all communicating devices are at one-hop distance of each other. The WnS is then enriched with low level omission monitoring and failure detection functions, which are useful building blocks to architect, above the Wns, Resilient Hard Real-Time protocols and services, such as reliable communications, node failure detection and membership, clock synchronisation and global notion of time.

Such kind of protocols and services are extremely helpful to build wireless-based distributed applications, and the mechanisms and properties one have introduced in the WnS definition constitute a first step towards the provision of similar guarantees in multi-hop interconnected WnS networks.

References

[1] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Priority assignment for real-time flows in WirelessHART networks," in *23rd Euromicro Conference on Real-Time Systems (ECRTS)*, 2011, pp. 35–44.

[2] W. Shen, T. Zhang, M. Gidlund, and F. Dobslaw, "SAS-TDMA: A Source Aware Scheduling Algorithm For Real-Time Communication In Industrial Wireless Sensor Networks," *Wireless Networks*, vol. 19, no. 6, pp. 1155–1170, 2013.

Characteristic	Protocol	
	P-ACK: Positive Acknowledgement	**N-ACK:** Negative Acknowledgement
Summary of protocol operation	sender transmits a data frame; correct recipients receiving a correct data frame respond with a positive acknowledgement (P-ACK); correct recipients receiving an incorrect data frame do nothing.	sender transmits a data frame; correct recipients receiving a correct data frame do nothing; correct recipients receiving an incorrect data frame respond with a negative acknowledgement (N-ACK).
Protocol overhead (bandwidth utilisation)	medium/high (n_{list} P-ACK responses)	none (in the absence of errors)
Timeout value	$\mathcal{T}_{timeout} = \mathcal{T}_{td} + \mathcal{T}_{ina}$	
Timer expiration	starts a new round (due to lack of responses in current round)	protocol termination (data delivery to correct nodes)
Successful last round: $(k+i+1)^{th}$	protocol termination (all responses received)	protocol termination (no responses received: timer expires)
Unsuccessful last round: $(k+i+1)^{th}$	lack of responses (recipient failure detected: persistent/crash)	N-ACK response received (recipient failure detected: persistent)
Node persistent failure	inaccurate (reported as crash failures upon $(k+i+1)^{th}$ round)	detected upon $(k+i+1)^{th}$ round
Node crash failure	detected upon $(k+i+1)^{th}$ round	cannot be detected
Worst case protocol execution time	$(k+i+1) \times (\mathcal{T}_{td} + \mathcal{T}_{data} + \mathcal{T}_{timeout}) = (k+i+1) \times (2.\mathcal{T}_{td} + \mathcal{T}_{data} + \mathcal{T}_{ina})$	$(k+i) \times (2.\mathcal{T}_{td} + \mathcal{T}_{data} + \mathcal{T}_{N-ACK}) + \mathcal{T}_{td} + \mathcal{T}_{data} + \mathcal{T}_{timeout} = (k+i) \times (2.\mathcal{T}_{td} + \mathcal{T}_{data} + \mathcal{T}_{N-ACK}) + 2.\mathcal{T}_{td} + \mathcal{T}_{data} + \mathcal{T}_{ina}$
Best case protocol execution time	$2.\mathcal{T}_{td} + \mathcal{T}_{data} + n_{list}.\mathcal{T}_{P-ACK}$ (all responses received: first round)	$2.\mathcal{T}_{td} + \mathcal{T}_{data} + \mathcal{T}_{ina}$ (no responses received: first round)
Intra-protocol recipients list	yes (removal of failed nodes: persistent and crash)	partial (removal only of persistent failed nodes)
Inter-protocol recipients list (persistent & crash failure detectors)	yes (faster recipient list update)	yes (complete and faster recipient list update)
Parameters	$k; i; \mathcal{T}_{td}; \mathcal{T}_{ina}$ - fundamental WnS parameters defined in Fig 2; n_{list} - number of nodes in the recipients list; $\mathcal{T}_{data}$ - the duration of a data frame transmission; $\mathcal{T}_{\{P,N\}-ACK}$ - the duration of the $\{P, N\}$ ACK frame transmissions	

TABLE II: Analysis of P-ACK versus N-ACK protocol characteristics

[3] Y.-H. Wei, Q. Leng, S. Han, A. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications," in *34th IEEE Real-Time Systems Symposium (RTSS)*, December 2013, pp. 140–149.

[4] J. L. R. Souza and J. Rufino, "Low level error detection for real-time wireless communications," in *13th International Workshop on Real-Time Networks (RTN) - In conjunction with the 26th Euromicro International Conference on Real-Time Systems (ECRTS)*, Madrid, Spain, July 2014.

[5] ——, "An approach to enhance the timeliness of wireless communications," in *The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, Lisbon, Portugal, november 2011.

[6] ——, "Towards resilient real-time wireless communications," in *25th Euromicro Conference on Real-Time Systems (ECRTS-WiP)*, Paris, France, July 2013.

[7] ——, "Analysing and reducing network inaccessibility in IEEE 802.15.4 wireless communications," in *38th IEEE Conference on Local Computer Networks (LCN)*, Sydney, Australia, October 2013.

[8] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *in MobiSYS 04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM Press, 2004, pp. 125–138.

[9] O. Babaoğlu and R. Drummond, "Streets of Byzantium: Network Architectures for Fast Reliable Broadcasts," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 6, Jun. 1985.

[10] IEEE 802.15.4, "Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs) - IEEE standard 802.15.4," 2011.

[11] ATMEL, *ATMEL AVR2025: IEEE 802.15.4 MAC Software Package - User guide*, ATMEL Coorporation, May 2012.

[12] T. Fujiwara, T. Kasami, A. Kitai, and S. Lin, "On the undetected error probability for shortened hamming codes," *IEEE Transactions on Communications*, vol. 33, no. 6, Jun. 1985.

[13] D. Eckhardt and P. Steenkiste, "Measurement and analysis of the error characteristics of an in-building wireless network," in *Annual Conference of Special Interest Group on Data Communication (SIGCOMM)*, 1996.

[14] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance study of IEEE 802.15.4 using measurements and simulations," in *Proceedings of the Wireless Communications and Networking Conference (WCNC 2006)*. Las Vegas, NV, USA: IEEE, Apr. 2006, pp. 487 – 492.

[15] J. L. R. Souza and J. Rufino, "Characterization of inaccessibility in wireless networks - a case study on IEEE 802.15.4 standard," in *IFIP 3th International Embedded System Simposium IESS*, September 2009.

[16] I. Ramachandran, A. K. Das, and S. Roy, "Analysis of the contention access period of IEEE 802.15.4 MAC," *ACM Transactions on Sensor Networks*, vol. 3, March 2007.

[17] M. Hameed, H. Trsek, O. Graeser, and J. Jasperneite, "Performance investigation and optimization of IEEE 802.15.4 for industrial wireless sensor networks," in *IEEE 13th International Conference on Emerging Technologies & Factory Automation (ETFA)*, September 2008.

[18] E. E-López, J. V-Alonso, A. M-Sala, J. G-Haro, P. P-Mariño, and M. Delgado, "A wireless sensor networks MAC protocol for real-time applications," *Personal Ubiquitous Computing Journal*, January 2008.