Wireless Sensor Actor Networks For Industry Control

Yoshihiro Nozaki, Nirmala Shenoy
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY, USA
yxn4279@rit.edu, nxsvks@rit.edu

Qian Li
CAST-Telecommunications Engineering Technology
Rochester Institute of Technology
Rochester, NY, USA
qxl2571@rit.edu

***Abstract*— A robust and reliable architecture for wireless sensor actor networks for industry control is discussed and described in this paper. The stringent physical constraints in an industry environment are taken into consideration. A combination of MAC and routing protocol to support reliable and robust transportation of data is described.**

Keywords-Sensor Actor Networks; Industry Control; Robust and Reliable Architectures.

I. INTRODUCTION

Wireless Sensor-Actuator Networks (WSAN) comprise of wireless sensors and actuators (or actors). Sensors are low-processing, low-energy devices that sense data such as temperature, pressure and so on. The sensed data is gathered at a *sink* to be analyzed and acted upon. Typically sensors are low-cost disposable devices. Based on the sensed data, actuators make decisions and take action. Actuators have higher processing capacity and are not energy constrained. They may also perform the functions of a *sink*.

Significant technology advances have resulted in major cost reductions in sensors and actuators. This coupled with elegant techniques to overcome challenges in wireless transmissions make WSANs attractive and viable for many applications. Examples are environment / habitat monitoring and control, battlefield surveillance, industry control and automation. In WSAN for environment and habitat monitoring and control, and battlefield surveillance, a large number of sensors are randomly deployed in potentially inaccessible areas, hence they be disposable and highly energy conserving. Multi-hop data collection paths, self-configuration and self-healing are predominant features of WSAN in such applications. Importance of security in such WSANs depends on the applications.

Considering a *Wireless Sensor-Actuator Network for Industry Control* (WSANIC), high survivability and ability to support data, event and task prioritization are predominant requirements. Security is important because of the critical nature of the application. For example explosives, high power and chemical industries could have serious detrimental effects in terms of cost and/or human loss if tampered with. The fact that sensors and actuators could be placed in least human-frequented areas makes them highly vulnerable to security attacks.

In contrast to the distinctive features mentioned earlier for WSANs, in a WSANIC, sensors and actuators are manually placed, resulting in a more stationary and deterministic topology. Self-configuration and self-healing are required upon device failures or environmental changes. Devices may not be disposable and batteries can be charged or changed regularly. Thus, some issues that pose serious challenges in WSAN are less problematic in WSANIC [3]. Robustness, interference in communications and data reliability are of major concern in a WSANIC. To improve robustness one has to look for options other than using powerful antennas as high power transmissions pose danger in inflammable spaces and increase interference effects [2]. In an industry environment, *high electromagnetic fields* due to heavy electrical devices and power cables are normal to expect, which negates the use of low power transmissions by sensor and actors. Communications interference is also caused due to events such as environment conditions, moving people and objects all of which can impact timely data transmission. Data reliability is critical as corrupted data could result in improper control of machinery and processes, which could be catastrophic.

Section II describes current industry control networks. Related works that are addressing WSANIC issues is provided in Section III. Section IV describes about WSANIC. Section V introduces our proposed architecture and Section VI analyses the result of simulations. Section VII provides the conclusions.

II. CONTROL NETWORKS IN INDUSTRY

Wired *Control Networks* (CN) are adequately supporting industry control requirements today. However, in industries dealing with explosives, moving, or rotating machinery, some locations are inaccessible or highly inconvenient to monitor using wired systems. The cabling and conduits for wired sensors and actuators besides being vulnerable to damage can be cost prohibitive - ranging typically to as much as one third to one half of the total system cost [1]. Industrial sensors have seen a steady decrease in costs and the eventual driving cost factor becomes cabling rather than the sensor or actuator cost. A low cost wireless sensor-actuator system with reasonable battery life to provide reliable data collection spanning an entire industry plant, while meeting cost objectives could create a paradigm shift in industry maintenance and control [1]. Such systems would also allow computing power in locations that previously would have been cost-prohibitive [4].

A. Wired Control Network

A *Process Control System* in an industry uses sensors to measure the process parameters and actuators to adjust the

 ISBN: 978-1-61208-256-1

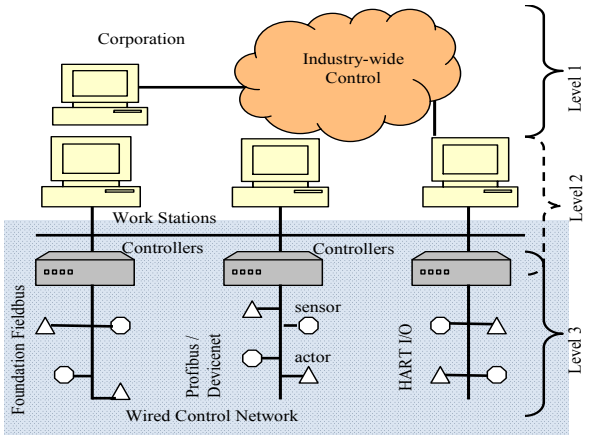


Figure 1. Wired Industry Control Network Architecture

operation of the process. Control action can be inbuilt into actuators or can be in separate entities called controllers. In industry control, it is convenient to have controllers separate from actuators as the controllers collect data from several sensors, make decision on an appropriate action to take (like *proportional, integral, derivative* or combinations of these) and actuate several actuators [3].

In Fig. 1, a typical wired industry-wide control network is shown. It has three levels of hierarchical control. The network at level 3 *that connects the sensors and actuators to the controllers* is of interest to us and we use the term ***wired CN*** for this segment. In this article, we analyze a ***wireless CN*** (WSANIC) that can replace the wired CN.

At level 3, *Foundation Fieldbus* (FF), *Profibus* and *Devicenet* are some of the wired CN industry standards being used [2]. The standards assume inherently *high predictability* and *reliability* as they operate over wired networks and target *real-time* data delivery. Real-time and reliable data delivery is very important in industry control, since loss of scheduled data could result in costly consequences [3]. Other performance affecting factors to consider are data rates, distance and transmission ranges. For example at the physical layer of FF, the official data rate is 31.25 Kbps. A process unit in a plant could span tens to hundreds of meters. Depending on the cable types and whether the controller is mounted close to the sensor/actuator or in a remote room, the distance range of FF is expected to be from 200 to 1900 meters [3]. *As a promising alternative to industry control, a WSANIC should have capabilities similar to the wired CN.*

B. Wireless Control Network

The frequency spectrum used in current wireless networks, can support high data rates. However, long transmission ranges are difficult to achieve given that high power transmissions are undesirable. In [4], Enwall T. provides statistics from studies conducted on suitability of major wireless network standards like 802.11g, 802.11s, Zigbee 802.15.4 and WiMax for industry control as per ISA-SP100. From the statistics it is clear that none of the above standards come close to doing what they need to do to fully support industrial applications. However, combining Zigbee with a *service broker* [4] improved its rating considerably, though it still fell short in several aspects such as network and messaging security, adequate reporting rates, quality of service in terms of timeliness, delivery ordering and recovery actions among others.

III. RELATED WORK

A survey of related literature reveals that there are few contributions that address WSANIC issues [1 – 4]. The prime focus in these articles are on how best to replace the FF or other similar wired CN [3] with a wireless counterpart.

From an industry and standards perspective, several wireless organizations are investigating solutions and pursuing adoption of wireless standards promoted by them. Of these WINA, Zigbee, ISA wireless system for automation, wireless HART are some major ones [2]. However none of these efforts takes into consideration industry environmental, placement and access restrictions.

In [8], the authors observe that "a WSAN should be robust to node failures and in general exhibit fast dynamic response to changes". In [9], researchers at *Massachusetts Institute of Technology* harnessed the robustness inherent in mesh topologies in a WSANIC test bed. These observations indicate that topology and architectural issues are important to consider in a WSANIC architecture. High survivability and security are of also very important. These are best addressed via suitable architectures and/or topology.

IV. WIRELESS SENSOR ACTUATOR NETWORKS FOR INDUSTRY CONTROL

We start with three main devices essential in a WSANIC, namely sensors, actuators and controllers and distinguish their functions in an industry control environment. Without loss of generality, it is assumed that sensors and actuators are distinct devices. Sensors are end devices that collect and transmit data while actuators are end devices that receive data and actuate a lever or valve. The controller, which we henceforth call an *Access Control Point* (ACP) is the data collection device that collects data from several sensors and is the source point of control data to several actuators. Inter-ACP communication required for industry wide control may be over wireless or wired links is out-of-scope in this work. ACPs will be limited in number and positioned at specific locations. Hence it may not be possible for all sensors and actuators to have line of sight communications path to an ACP. For robustness in connectivity it is further essential that sensors and actuators have routes to multiple ACPs.

A. The Architecture

To overcome the physical issues due to communications range, line of sight and to provision multiple paths between ACPs and sensor/actuators special devices called 'relays' are introduced. Relays forward data for other devices and will provide multiple paths of communications. It has been observed [5] that multiple types of devices result in complex management due to diversity in techniques, data collection methods and protocols. In the proposed architecture, multiple types of devices are necessary to provide robustness and adaptability. However complex communications and

 ISBN: 978-1-61208-256-1

management are avoided by using a set of medium access and routing protocols common to all devices.

The architecture comprising of ACPs, sensors, actuators and the relay mesh that emerges from the discussions thus, far is pictured in Fig. 2. The emphasis is on WSANIC at level 3 that will embed into the 3-level hierarchy from Fig. 1. As per the architecture, relays and an Access Control Point (ACP) are used besides sensors and actors. The ACP is responsible for implementing the proportional, integral and/or derivative control depending on the process. The control action is then conveyed to the actuators. The relays facilitate robust connectivity between the ACPs and actuators; ACPs and sensors by providing redundant paths. They are also useful to keep the transmission power low, and facilitate multi-hop communications when two nodes are distant to one another.

B. The Protocols

In a typical wired CN standard like the FF, the protocol stack is derived from the OSI 7 layer model, where only the lower two layers namely the physical and the data-link are specified; the network, transport and session layers are removed[3]. The proposed protocol stack for WSANIC also has two layers. The lower layer is the physical layer, which is not the focus of this article, and the layer above i.e. layer 2, has integrated medium access control (MAC) and routing functions that operate off a single header. This is very attractive in wireless networks as it reduces header overhead, processing requirements and its associated delays, while allowing MAC and routing functions to interwork closely.

C. The MAC Functions

A MAC protocol for WSANIC should provide timely and near-lossless data delivery that is comparable to wired CN. In wired CN, it is naturally assumed that priority data carrying vital information under alarm conditions will be delivered reliably and in time. However, this assumption is not valid in wireless networks and sensitive, urgent data has to be handled specially to facilitate timely and reliable delivery.

Timely delivery can be achieved through preemptive priority. Preemption requires abortion / delay of other transmissions or receptions on the arrival of high priority data. This capability can be provisioned through the use of a dual channel MAC (one channel to carry high priority data and another for normal data) where the MAC switches the local processing to handle high priority data on its arrival.

Reliability can be achieved through retransmissions on loss of acknowledgements, if accomplished within acceptable latency limits or in the routing functions through the use of concurrent multipath transmissions of critical data to increase the probability of its delivery.

Normally a scheduled MAC is considered suitable for reliable and timely delivery of data. However, we advocated a multi-hop mesh topology which makes it difficult if not impossible to implement scheduled MAC due to synchronizations issue. Moreover in industry environment, an unscheduled MAC will have more flexibility to provide combinations of periodic, event-based and query-based data collection / delivery. If an unscheduled MAC is used, reliability of data delivery has to be achieved via acknowledgements and retransmissions. Given the frequency spectrum used in current wireless networks, the data rates achieved are very high compared to a wired CN data rates (like FF) and retransmissions on loss of acknowledgements can be processed within acceptable latency limits. The routing scheme to be presented next also support timely and reliable data delivery, as it has the capability to send priority data concurrently on proactively maintained multiple paths.

D. Routing Functions

ACPs, sensors and actuators in WSANIC can be stationary or mobile. The set of relays that forward data from sensors to actuators can vary due to mobility of ACPs, sensors, and actuators; battery drain at relays or environmental changes. A single route is not advisable as data loss due to route failure could occur. Multiple routes from sensors to ACPs and ACPs to actuators can alleviate this problem. Delays due to new route discovery also cannot be tolerated in critical applications. Hence a robust proactive multipath routing scheme with low overheads would be ideally suited. The *Multi Meshed Tree* (MMT) routing [6] [7] has these desirable features.

E. MAC and Routing Protocols

The MAC protocol uses carrier sensing similar to 802.11, but adopts a more deterministic medium access approach. In this new approach, nodes take turns to access the media, based on neighbor knowledge and is called the *Neighbor Turn Taking* (NTT) MAC protocol [10]. This protocol has been previously shown via simulation to perform better than IEEE 802.11 CSMA/CA in terms of end-to-end packet latency and rate of successfully transmitted packets under saturated traffic conditions [11]. The proposed routing scheme sets up overlapping (meshed) trees originating at the ACPs and ending at the sensors and actuator. The meshed trees provide multiple robust routes. They also use neighbor knowledge and are based on the MMT algorithm.

V. IMPLEMENTATION

In this section, we describe the integrated NTT and MMT (NTT-MAC) operation.

A. The Semi-Automated Architecture

Fig. 2 shows the semi-automated architecture [12] with relays, sensors, actuators, and ACPs. In this architecture, sensors send data to ACPs, and collected data is processed at

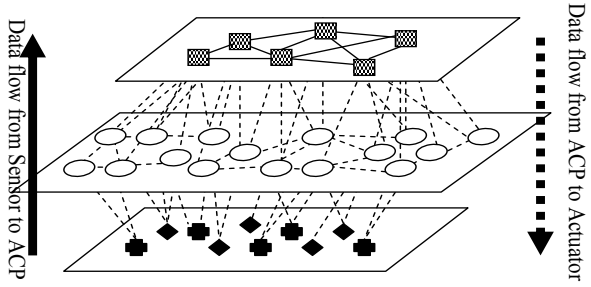


Figure 2. The Semi-automated Architecture

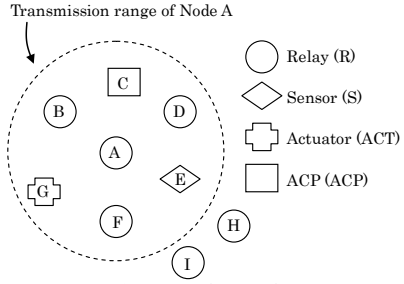
 ISBN: 978-1-61208-256-1

ACPs. The architecture shows 3 layers; the top layer is mesh of ACPs. The middle layer is a mesh of relay nodes, and the bottom layer comprises of sensors and actuators. All nodes in this architecture communicate over a wireless media except for the ACP mesh which could be wire connected. After the data is processed in ACPs, ACPs decide on the proper actuators that are to be activated and communicate to them. In the semi-automated architecture, route maintenance for both sensors-ACPs and ACPs-actuators routes is required. This will result in two way communications along the routes established. Hence, a MAC with low collisions low latency and a robust routing protocol are essentail.

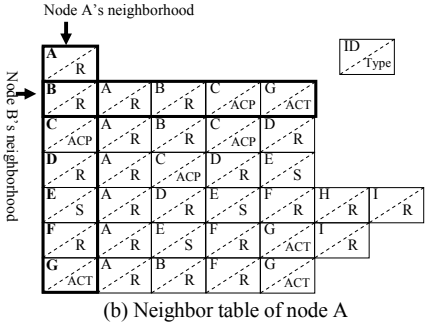
B. Neighbor Turn Taking Medium Access Control

NTT-MAC uses a distributed loosely scheduled approach based on neighbor knowledge and their activities. NTT operation requires two processes, 'neighbor sensing' and 'turn scheduling'. Because there are four different types of nodes sensors, relays, actuators, and ACPs, the NTT-MAC proposed in [10] was customized to the new architecture.

1) Neighbor Sensing: Each node overhears the neighbor nodes to calculate its turn to access the medium next. To accomplish this, all nodes in the network advertise themselves and their 1-hop neighbors periodically. Nodes thus, know their neighbor's neighbor information i.e. 2-hops neighbor information. In addition, node type such as sensor, relay, actuator, and/or ACP is also advertised. Fig. 3(b) shows an example of neighbor knowledge of the topology in Fig. 3(a). Nodes B, C, D, E, F, and G are neighbors of Node A. In Fig. 3(b), the left most column in the table represents Node A's neighbor list and each row represents each neighbor's neighbor list. For example, Node B's neighbors are nodes A, C, G and their node types are relay (R), ACP, and actuator (ACT).



(a) Example Topology



(b) Neighbor table of node A

Figure 3. Neighbor knowledge example

2) Turn Scheduling: Turn scheduling is achieved based on neighbor table and their activities as described next.

a) Neighbor Activities: Each node calculates its next turn based on the sender node's neighbor list which it overhears from its neighbors transmissions. For example, if Node B in Fig. 3 (a) sends a packet, all neighbors nodes A, C, and G hear the transmission of Node B. They will then calculate their next turn by looking up Node B's neighbor list. The neighbor list indicates the order of each node's turn. Therefore, the next sender from Node B will be Node C, and second sender will be Node G, third will be Node A. In order to synchronize their turns, the order in each neighbor list has to be the same with all neighbors. In this work, ACK is used for DATA, and hence each node computes their turn to transmit based on the type of message they overhear.

b) Node's activities: The turn calculation is based on a node's neighbor size. For example, Node B calculates its next turn to be 4^{th} because its neighbor size is 3.

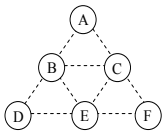
c) Updating: Each node has one next turn scheduled at any time. Thus, each node compares previous turn scheduling time and new turn scheduling time after every turn calculation, and applies the latest scheduled one.

C. Multi Meshed Tree Routing

For routing, the Multi-Meshed Tree (MMT) protocol is used to create logical meshed trees in the network. These trees are rooted at the ACP, and the ACTs and sensors are the leaf nodes. Since the semi-automated architecture has two-way data flow, sensor nodes need routes to ACPs and ACPs need routes to actuators. In addition, a sensor can communicate with any ACP and any ACP can communicate with any actuator. Hence, both sensors and ACPs are required to maintain routing information. As a result, route maintenance can become complicated and difficult. Most well-known routing protocols (proactive and reactive) in wireless ad hoc networks such as *Dynamic Source Routing* (DSR) and *Optimized Link State Routing* (OLSR) are required to maintain routing information at sender nodes. MMT requires only ACPs to maintain route information to ACTs. Sensors have the route information to ACPs, which is inherent in their allocated virtual IDs (VIDs). By nature of MMT, leaf nodes in the trees such as sensors and actuators can know routes to the root nodes of the trees once they joined the trees as this information is inherent in the assigned VIDs to the leaf nodes. Likewise, the root nodes such as ACPs know routes for both sensors and actuators. Therefore, sensors do not require to maintain routing information. Because the logical trees are meshed, MMT protocol provides not only overlapping coverage, but also route robustness while avoiding loops in the meshed topology. An optimized version of the MMT algorithm as presented in [7] is used to reduce control packets of MMT in this work.

1) Multi-Meshed Trees (MMT)

As mentioned above, trees are grown from root nodes (ACPs) to leaf nodes (i.e. sensors and actuators) through relay nodes. Each meshed-tree can be viewed as a cluster and the ACP is the cluster head (CH) and all other nodes are the



	A	B	C	D	E	F
VID 1	1	11	12	111	112	123
VID 2		121	113	1211	122	1132
VID 3		1221	1123	1121	1212	1122
VID 4				1222	1131	1223
VID 5					1111	
VID 6					1231	

Figure 4. Example of MMT (Hop limit = 3)

cluster clients. 3-ways handshake is adopted by nodes when during the joining process. The ACP or CH initiates tree creation by broadcasting an advertisement (AD) containing its VID. On hearing the AD packet, neighbor nodes which want to join the tree will send a join request (JR) to the sender of AD packet i.e. the parent node. The parent then records the new VID into a JR message and forwards to the CH, which register the new VID to its cluster member. Because the child node can hear the forwarded JR message, the child can know the new VID assigned to it at the time. The CH replies with a join acceptance (JA) packet to the parent after registering the new VID. Finally, the parent sends the JA to the child. And then, the child node starts to advertise its new VID to its neighbors. The new VID for a child node is one additional digit appended to the parent's VID. Fig. 4 shows an example topology and VIDs in MMT. For example, if a CH node A has VID *1*, the child VID can be between *11* and *19*. So, Node B and C will get VID *11* and *12*. Since Node C has *12*, its child can be between *121 – 129*. In this manner, the VID carries the route information. The total number of digits in a VID indicates the hop distance from CH, and also route to CH. The process continues until the tree encounters defined limits such as maximum hop count, cluster size or reaching edge nodes.

To avoid loops in trees, VIDs are not assigned if there is already a child-parent relationship with a particular VID. This VID acceptance rule applies for not only direct parent-child, but also for any grandparents or grand children.

We include the knowledge from NTT into the joining process by combining JR and JA during the 3-ways handshake as shown in Fig. 5. Nodes B and C are neighbors of Node A which has VID *111*. After Node A broadcasts its VID, Node A calculates its next turn based on its neighbor table. Node B and C overhear Node A's AD packet and calculate their next turn based on Node A's neighbor table. As their turn scheduling is based on Node A's neighbor table, next turn scheduling time of Node B and C are the

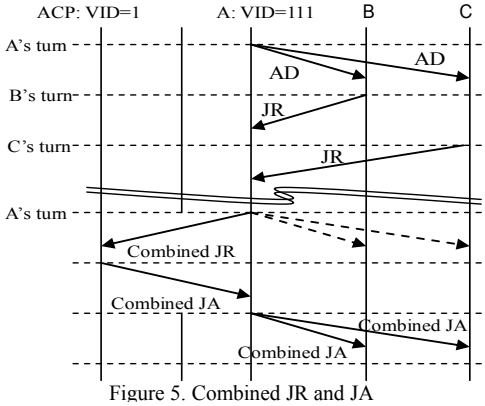
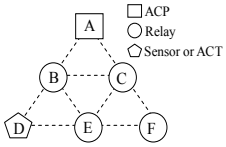


Figure 5. Combined JR and JA



	A	B	C	D	E	F
VID 1	1	11	12	111	112	
VID 2		121	113	1211	122	
VID 3				1121		
VID 4				1221		

Figure 6. MMT for the semi-automated architecture (Hop limit = 3)

time before Node A's next turn. Hence, join request from all Node A's neighbor can be received before Node A's next turn to transmit. Therefore, Node A can combine all JR messages from its neighbors ideally and assign new VIDs for all the children nodes when Node A gets its next turn and forward the combined JR to the CH. The CH node returns a combined join acceptance (combined JA) to Node A after new VIDs are registered.

D. Interaction Between NTT and MMT

Since MMT uses neighbor knowledge for optimized cluster joining process, MMT interacts with NTT to look up neighbor table. Each node maintains neighbor knowledge which includes not only node ID but also node types. MMT helps set up routes between sensor to ACP and ACP to actuator. If HOP_LIMIT is 5 and a parent VID is *1111*, the parent is located 3 hops from the CH 1. If a child node joined this VID, the child node will be at the 4th hop from CH 1 and the child's child node will be at the 5th hop (last hop). Therefore, if the child node does not have SENSOR or ACT node in its neighbor, the child VID will be meaningless and would use up one node cluster client position wastefully. Thus, a child node which does not have any SENSOR or ACT (actuator) in its neighbors' neighbor will not send JR to the parent if the parent VID is already HOP_LIMIT - 2. Fig. 6 shows optimized MMT for the sample topology. SENSOR and ACT do not allow having child node, so Node D does not have child VID. Because Node F does not have any SENSOR and ACT in its neighbor and Node C and E have already reached 2 hops from the CH, Node F does not join any tree. As a result, total number of control packets is reduced significantly because total number of VIDs is reduced. On the other hand, NTT interacts with MMT to identify sender and destination nodes from VIDs and to calculate turn scheduling from neighbor table and own VIDs.

VI. ANALYSIS RESULTS

A. The Topology

Fig. 7 is the topology used in the OPNET simulations [13]. The topology shows relative placement of the sensors and actuators with respect to the ACPs which is similar to semi-automated industry architecture.

The topology places the relays, sensors and actuators around the ACPs but with the relays between the sensors / actors and ACPs. Several simulations were conducted by varying the number of sensors / actors and with different simulation seed values and the results averaged. The tests were repeated using DSR and 802.11 CSMA/CA for comparison between them. At the ACPs and the sensors, data was generated at the rate of one packet in 0.05 seconds. The data packet size was maintained at 500 bits.

 ISBN: 978-1-61208-256-1

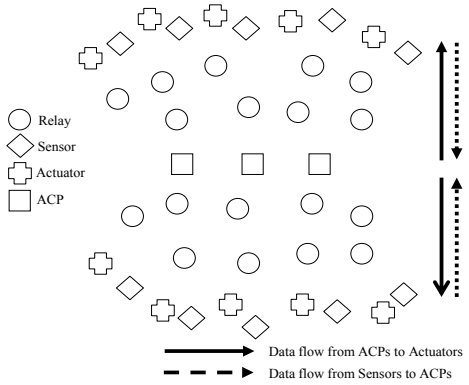


Figure 7. Relative placement of sensors, actuators, relays, and ACPs

B. Performance Metrics

1) Average End to End Latency is the time taken from transmission of a data packet at the sender to its reception at the receiver.

2) Success rate is calculated as the ratio of total number of packets received correctly at the destination node to the total number of packets sent by the sender node.

Two different situations for MMT based systems in each scenario were simulated. One is with 'route salvage' option which has salvage function. The other one doesn't have route salvage. DSR has 'route salvage' implemented. As can be seen in table 1, in all scenarios MMT/NTT based solutions has a consistently higher success rate of over 98%.

The latency was also recorded against the number of hops between the sending and receiving nodes. In most cases, the average end to end delivery latency for MMT/NTT is lower than DSR/CSMA/CA despite the fact that the MMT/NTT delivered more packets.

VII. CONCLUSIONS

The aim of this article was to provide insights into architectural and design issues that could affect the design of a wireless sensor-actuator networks for industry control. Towards this we described the physical constraints encountered in a wireless industry environment and proposed a suitable topology and an architecture that would address survivability and security. We then highlighted MAC functions essential to handle data, task and event prioritization, which is vital for wireless industry control. Lastly we identified a secure routing scheme that complements and integrates into the MAC, to provide the requisite connectivity robustness.

The NTT-MAC is contention based but uses a loosely scheduled medium access scheme that does not require strict time synchronization or a central server because it schedules based on neighbor activity. The main performance aspect we targeted when we developed NTT-MAC scheme was to achieve reduced latency, higher success rate and fairness in medium access among contending users. We also introduced a routing protocol based on the MMT algorithm, which is a proactive routing protocol at layer 2 along with the NTT MAC. MMT is developed to support high route robustness with a quick and easy forwarding approach based on virtual

TABLE I. PERFORMANCE COMPARISON MMT-NTT VS DSR

Metrics \ Scenarios			5 ACT/SENSOR	10 ACT/SENSOR
Success Rate	MMT-NTT (route salvage)		99.104930	99.802880
	MMT-NTT (no route salvage)		98.310980	99.568000
	DSR		95.330350	92.823010
Packet Latency	MMT-NTT (route salvage)	1-hop	N/A	N/A
		2-hop	0.000917	0.002599
		3-hop	0.002643	0.005812
		4-hop	0.002836	0.009116
		5-hop	N/A	0.006059
	MMT-NTT (no route salvage)	1-hop	N/A	N/A
		2-hop	0.001000	0.002100
		3-hop	0.002754	0.005468
		4-hop	0.002820	0.007900
		5-hop	N/A	N/A
	DSR	1-hop	N/A	0.000759
		2-hop	0.001719	0.002600
		3-hop	0.004089	0.004930
		4-hop	0.007069	0.010240
		5-hop	0.008872	0.014348

IDs. In industry control, Wireless Sensor-Actuator Ad-hoc Network using NTT-MAC algorithm and MMT-routing algorithm will provide high quality of performance. The performance metrics focused were success rate and packet delivery latency. The simulation results show improved performance of MMT-NTT in terms of success rate and end to end latency than DSR operating with 802.11 MAC.

REFERENCES

[1] T. Brooks, "Wireless technology for industrial sensor and control networks," Sensors for Industry, 2001. Proceedings of the First ISA/IEEE Conference, 2001, pp.73-77, doi: 10.1109/SFICON.2001.968502.

[2] J. Song, A. K. Mok, D. Chen, and M. Nixon, "Challenges of wireless control in process industry," in Workshop on Research Directions for Security and Networking in Critical Real-Time and Embedded Systems, San Jose, CA, 2006, http://moss.csc.ncsu.edu/~mueller/ftp/pub/mueller/papers/cps06.pdf. (accessed March 2013)

[3] D. Chen, M. Nixon, T. Aneweer, R. Shepard, and A. K. Mok, "Middleware for wireless process control systems," Workshop on Architectures for Cooperative Embedded Real-Time Systems, 2004, http://wacerts.di.fc.ul.pt/papers/Session1-ChenMok.pdf. (accessed March 2013)

[4] T. Enwall, "Deploying Wireless Sensor Networks for Industrial Automation and Control," http://www.eetimes.com/design/industrial-control/4013661/Deploying-Wireless-Sensor-Networks-for-Industrial-Automation-Control.(accessed March 2013)

[5] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," Ad Hoc Networks, Volume 2, Issue 4, October 2004, pp. 351-367.

[6] N. Shenoy, Y. Pan, and V. G. Reddy, "Quality of Service in Internet MANETs", Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on , vol.3, 2005, pp. 1823-1829.

[7] N. Shenoy, Y. Pan, D. Narayan, D. Ross and C. Lutzer, "Route robustness of a multi-meshed tree routing scheme for Internet MANETs," Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE, vol.6, 2005, pp. 3351-3356.

[8] B. P. Gerkey and M. J. Mataric, "A market-based formulation of sensor-actuator network coordination," in Proceedings of

the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems, Palo Alto, California, March 25-27 2002, pp. 21-26.

[9] P. Robert, "Wireless Mesh Networks", http://www.sensorsmag.com/networking-communications/standards-protocols/wireless-mesh-networks-968. (accessed March 2013)

[10] N. Shenoy, C. Xiaojun, Y. Nozaki, S. Hild and P. Chou, "Neighbor Turn Taking MAC - A Loosely Scheduled Access Protocol for Wireless Networks," Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on, 2007, pp. 1-5.

[11] E. F. Golen, Y. Nozaki and N. Shenoy, "An analytical model for the Neighbor Turn Taking MAC protocol," Military Communications Conference, 2008. MILCOM 2008. IEEE, 2008, pp. 1-7.

[12] L. Barolli, T. Yang, G. Mino, F. Xhafa and A. Durresi., "Routing efficiency in wireless sensor-actor networks considering semi-automated architecture," J. Mob. Multimed, vol.6, 2010, pp. 97-113.

[13] OPNET modeler, http://www.opnet.com/. (accessed March 2013)

 ISBN: 978-1-61208-256-1