An Architecture for 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 (WASANIC) is discussed and described in this paper. The stringent physical constraints in an industry environment are taken into consideration. We proposed an architecture that allows efficient cross-layering between a semi-scheduled medium access control (MAC) protocol called the Neighbor Turn Taking MAC (NTT-MAC) and a routing protocol based on the Multi-Meshed Tree (MMT) routing algorithm that is suited to the WSANIC topology encountered in an industry. The proposed architecture also addresses survivability and security. The cross-layered approach, named NTT-MMT, supports reliable and robust transportation of data. Through simulations, the performance of NTT-MMT was compared with carrier sense multiple access with collision avoidance (CSMA/CA) MAC and dynamic source routing (DSR) protocol.**

Keywords-Sensor Actor Networks; Industry Control; Robust and Reliable Architectures; Cross Layering; Medium Access Control

I. INTRODUCTION

This paper is the extended version of the conference paper [1], and aimed at providing a deep insight into the integration between a Medium Access Control (MAC) protocol, the Neighbor Turn Taking (NTT) [2][3], and routing protocol, the Multi-Meshed Tree (MMT) [4]. Towards this, we describe the physical constraints encountered in a wireless industry environment and propose a suitable topology and an architecture that would address survivability and security. We then highlight 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.

Wireless Sensor-Actuator Networks (WSAN) comprise of wireless sensors and actuators (or actors). Typically, 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. In some cases, sensors are low-cost disposable devices. Based on the sensed data, actuators make decisions and take action. Actuators normally have higher processing capacity and are not energy constrained. They may also perform the functions of a sink.

Significant hardware and software 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 are disposable and should be 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 very 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. Due to 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 are rarely disposable and batteries can be charged or changed regularly. Thus, some issues that pose serious challenges in WSAN are less problematic in WSANIC [5]. 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 [6]. 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. Furthermore, in a WSANIC, some data may have to be transported with least latency, i.e., high priority and without loss, as they may need an immediate action to be taken.

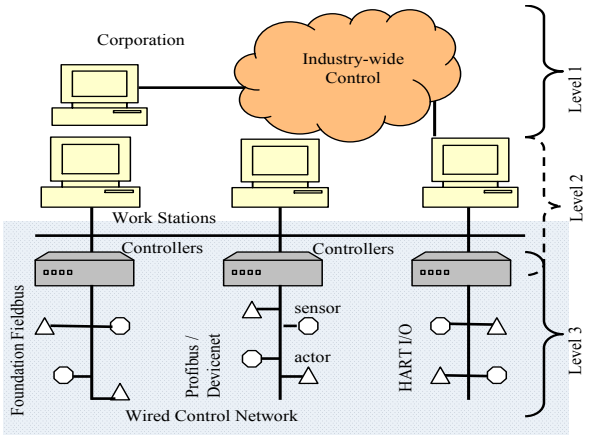


Figure 1. Wired Industry Control Network Architecture

To achieve close to real-time communications between sensors and actuators, a medium access control (MAC) protocol that allows for timely and reliable delivery is necessary. If multi-hop or multi path communications are required to introduce redundancy and robustness in the paths used to deliver data, then a reliable and robust routing protocol is equally important. The two protocols should operate with low complexity and interwork efficiently. We proposed an architecture presented in [1] that allows efficient cross-layering between a semi-scheduled MAC protocol called NTT-MAC and a routing protocol based on MMT algorithm that is suited to the WSANIC topology encountered in an industry. In this article, expanded cross-layering approach, multi-hop and multipath routing maintenance, and security concerns are discussed.

This paper has the following structure: 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 NTT-MAC and also discusses detailed cross-layering approach and Section VI analyses the result of simulations. Section VII provides the conclusions.

II. CONTROL NETWORKS IN INDUSTRY

Wired Control Networks (CN) have been adequately supporting industry control network requirements till date. However, in industries dealing with explosives, moving, or rotating machinery, some locations are inaccessible or highly inconvenient to monitor using wired sensor and actuator 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 [7]. Industrial sensors meanwhile have seen a steady decrease in cost and the eventual driving cost factor in wired industry control networks is the cabling cost rather than the sensor or actuator cost. A low-cost wireless sensor-actuator system with reasonable battery life that provides reliable data collection spanning an entire industry plant, while meeting certain cost objectives would create a paradigm shift in industry control and automation [7]. Such systems would also allow the penetration of computing capabilities in locations that previously would have been cost-prohibitive [8]. In the section below, we discuss some of the most adopted wired industry control network topologies and standards.

A. Wired Control Network

A *Process Control System* in an industry uses sensors to measure the process parameters and actuators to adjust the 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 [5].

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 propose an architecture and suitable protocols for a wireless CN (earlier termed the WSANIC) that can replace the wired CN and analyze the performance of such a WSANIC.

At level 3 in Fig. 1, Foundation Fieldbus (FF), Profibus and Devicenet are some of the wired CN industry standards being used [6]. The standards assume inherent high predictability and reliability as they operate over wired networks and hence the target of real-time data delivery should be achievable. Real-time and reliable data delivery is very important in industry control, since loss or untimely delivery of scheduled data could result in costly consequences [5]. 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 [5]. As a promising alternative to industry control, a WSANIC should have capabilities similar to the wired CN and address the critical targets set by the wired CN standards.

III. RELATED WORK

The frequency spectrum used in current wireless networks can support high data rates. However, long transmission ranges are difficult to achieve as high power transmissions are undesirable in an industry typically those that handle explosives or highly inflammable material. In [8], 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 standards [9]. From the statistics it is clear that none of the above standards come close to doing what they need to do to fully support industry applications. However, combining Zigbee with a service broker [8] improved Zigbee's 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

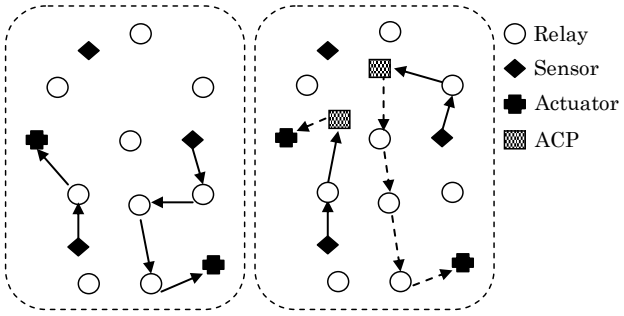


Figure 2. WSANIC architectures

among others.

A survey of related literature reveals that there are few contributions that address WSANIC issues [5] - [8]. The prime focus in these articles are on how best to replace the FF or other similar wired CN [5] 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. Among these, Wireless Industrial Network Alliance (WINA), Zigbee, International Society of Automation (ISA) wireless system for automation, and Wireless Highway Addressable Remote Transducer (WirelessHART) protocol are some major ones [6]. However, none of these efforts takes into consideration industry environmental, placement and access restrictions.

In [10], the authors observe that "a WSAN should be robust to node failures and in general exhibit fast dynamic response to topology or connectivity changes". In [11], 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 for WSANIC architecture. High survivability and security are equally important. The varied features are best addressed through suitable architectures and / or topologies.

IV. WIRELESS SENSOR ACTUATOR NETWORKS FOR INDUSTRY CONTROL

We start with three main devices that are essential in a WSANIC, namely sensors, actuators and controllers. We distinguish their functions in an industry control environment to aid in a suitable architecture design. Without loss of generality, it is assumed that sensors and actuators are distinct and separate devices. Sensors are end devices that collect and transmit data while actuators are end devices that receive data and actuate a lever or valve in an industry control process. 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 that controls the operations of several actuators. Inter-ACP communication required for industry wide control may be over wireless or wired links is not considered in this architecture. It is reasonable to assume that ACPs will be limited in number and positioned at specific locations. Hence, it may not be possible for all sensors and actuators to have a 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.

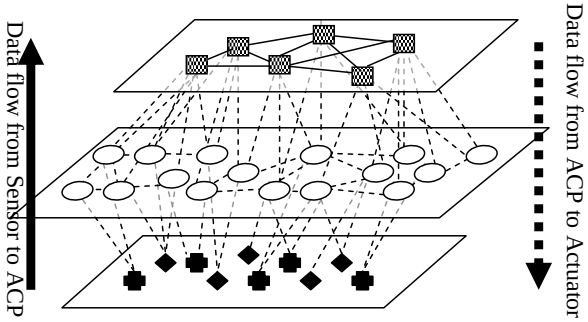


Figure 3. The Semi-automated Architecture

A. The Architecture

To overcome the physical constraints due to communications range, line of sight transmissions and to provision multiple paths between ACPs and the sensors / actuators special devices called 'relays' are introduced. Relays forward data for other devices and will also aid in setting up multiple paths of communications between ACPs and sensors / actors. It has been observed in [12] 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 management are avoided by using a set of medium access and routing protocols that is common to all devices.

The architecture thus designed for WSANIC should include consideration of ACPs, sensors, actuators and the relay mesh. Fig. 2 shows such typical architectures and the topology linking the different devices that can be used for the purpose. As seen in Fig. 2 (a), there is no ACP in the automated architecture because actuators can process collected data and make decision automatically thereby replacing the need for special devices to perform the action. The data flow in this architecture will be a one-way communication from sensors to actuators. While the automated architecture has one-way communication, the semi-automated architecture has two-way communications, i.e., between sensors to ACPs and ACPs to actuators [13], as seen in Fig. 2 (b).

Fig. 3 expands the architecture in Fig 2b, by positioning the devices that also shows the linking and connectivity between the different devices. In this architecture, sensors send data to ACPs, and the collected data from several sensors is processed at the ACPs. Fig. 3 shows the logical view of the architecture. Thus, there are distinct 3 layers, comprising of a top layer, which is a mesh of ACPs, a middle layer, which is a mesh of relay nodes, and a bottom layer, which comprises of sensors and actuators. In an industry, the physical location of the devices may not be separated as seen in Fig. 3.

Between each layer and among the middle layer entities, wireless links are assumed to be used for communications. After the collected data is processed in ACPs, the ACPs will

make decision for proper actions to be initiated in the actuators and forward the commands to the actuators. Since ACPs can be powerful computers and wired to each other, they can process much effective and take collaborative decisions as compared the automated architecture. However, the semi-automated architecture requires route maintenance between sensors and ACPs, and ACPs and actuators. There will be more transmissions than the semi-automated architecture due to the two-way communication. Therefore, the semi-automated architecture needs improved MAC in terms of less collision and latency and robust routing protocols that leverage the multiple paths and multi hops in the architecture.

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 [4]. The proposed protocol stack for WSANIC also follows the two-layer approach. 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 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 Medium Access Control 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 channel for normal data. The MAC switches the local processing to handle high priority data on its arrival. However, this requires increased performance capabilities in the wireless nodes.

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

A scheduled MAC is more suitable for reliable and timely delivery of data. However, as we advocated a multi-hop mesh topology a scheduled MAC is difficult to implement due to synchronizations issue. Moreover, in an industry environment, an unscheduled MAC will have more flexibility as it can provide combinations of periodic, event-based and query-based data collection and delivery. If an unscheduled MAC is used, then 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 the 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.

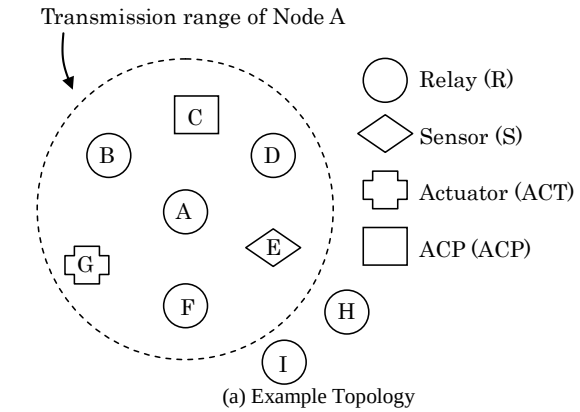
D. Routing Functions

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 (which is rare), sensors, and actuators; battery drain at relays or environmental changes which can impact the wireless link between a pair of devices. In this case, a single route is not advisable as data loss due to route failure has a high probability of occurrence. 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 such critical situations. Hence, a robust proactive multipath routing scheme with low overheads would be ideally suited. Routing based on the Multi Meshed Tree (MMT) algorithm [4] [14] has these desirable features.

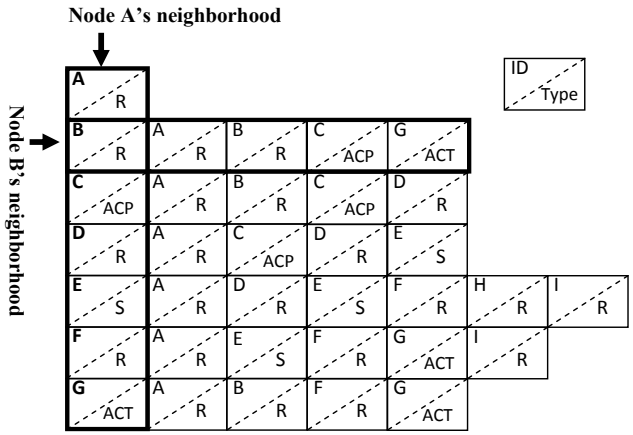
V. IMPLEMENTATION

We stated earlier that the MAC and routing functions would be integrated and operate off a single protocol header. Hence, in this section, we first describe the operational details of the Neighbor Turn Taking (NTT-MAC) and then the operation of the MMT routing protocol. This is followed by the details of integrating the two operations.

The NTT-MAC protocol uses carrier sensing similar to IEEE 802.11 CSMA/CA [15], but adopts a more deterministic medium access approach. In this new approach, nodes take turns to access the media, based on neighbor knowledge and hence is called the Neighbor Turn Taking MAC protocol [2]. 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 [3]. The MMT based routing sets up overlapping (meshed) trees originating at the ACPs and ending at the sensors and actuator. The meshed trees provide proactively established multiple robust routes. MMT algorithm also uses neighbor knowledge for its operation. Thus, the cross-layering approach adopted in the proposed architecture integrates the functions.



(a) Example Topology



(b) Neighbor table of node A

Figure 4. Neighbor knowledge example

A. *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 [2] has been customized to the new architecture with the four different types of devices. We now explain the different operations in NTT-MAC.

1) Neighbor Sensing: Each node overhears messages sent by its 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. Thus, nodes know their neighbor's neighbor information, i.e., 2-hops neighbor information. In addition, node types such as sensor, relay, actuator, and/or ACP is also advertised. This advertisement is also used as a hello protocol [16] to detect any change in the 2-hop neighbors. Fig. 4 (b) shows an example of neighbor knowledge of the topology in Fig. 4(a). Nodes B, C, D, E, F, and G are neighbors of Node A. In Fig. 4 (b), the left most column in the table represents Node A's neighbor list and each row represents each neighbor's neighbor list including itself. For example, Node B's neighbors are nodes A, C, G and their node types are relay (R), ACP, and actuator (ACT). These neighbors' node type information is included in the advertisement.

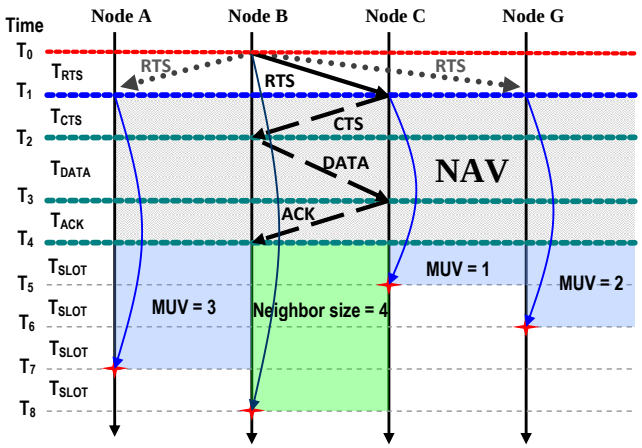


Figure 5. Example of Turn Calculation

2) Turn Scheduling: In NTT-MAC, a turn slot time (T_{SLOT}) is allocated to each node after the computations by the turn scheduling algorithm. Turn scheduling is achieved based on the 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. 4 (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. Fig. 5 illustrates the turn calculation initiated by Node B's activity namely a data transmission by Node B. A ready-to-send (RTS), clear-to-send (CTS), DATA, and an acknowledgement (ACK) packet are used for data transmission. When Node B has data to send to Node C, Node B sends RTS to Node C. Since Nodes A and G are neighbors of Node B, they also hear the RTS packet at time T_1 in Fig. 5. Then, all neighbors of Node B calculate their next turn. Because there will be a sequence of packet transmissions between Node B and C, the total transmission time is the sum of time to send CTS (T_{CTS}), DATA (T_{DATA}), and ACK (T_{ACK}) transmissions. This is called a network allocation vector (NAV). In addition to the *NAV* time, each node calculates its next turn based on the position in the sender's neighbor list, named medium user value (*MUV*). According to Node B's neighbor list in Fig. 4 (b), the turn taking order is Node A→ B → C → G. When Node B is taking a turn, MUV of Node C, G, and A are 1, 2, and 3 respectively. The time T_{SLOT} is greater than or equal to the time to transmit RTS (T_{RTS}), to provide chance to send an RTS packet. Total wait time (T_{WAIT}) for each node at time T_1 in Fig. 5 can then be calculated as:

$$NAV = T_{CTS} + T_{DATA} + T_{ACK} \quad (1)$$

$$T_{WAIT} = NAV + (MUV \times T_{SLOT}) \quad (2)$$

Based on the type of packet received, the value of *NAV* will be updated. For example, *NAV* at time T_2 will be:

$$NAV = T_{DATA} + T_{ACK} \quad (3)$$

And at time T_3:

$$NAV = T_{ACK} \quad (4)$$

Therefore, the first sender after Node B will be Node C at T_5, and the second sender will be Node G at T_6 if Node C did not send any packets. If Node C sends a packet at time T_5, all neighbors recalculate their next turn based on the types of packet they overhear. In order to synchronize their turns, the order in each neighbor list has to be the same with all neighbors.

In WSANIC, data from a specific sensor and ACP may have higher priority than others. In this case, these nodes can get more chance to send data by adding a duplicate entry for themselves in their neighbor list and advertise it. Thus, they can take turns more frequently.

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

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

B. Multi Meshed Tree Routing

For routing, the Multi-Meshed Tree (MMT) algorithm is used to create logical meshed trees in the network. These trees are rooted at the ACPs. 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) [17] and Optimized Link State Routing (OLSR) [18] 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). Inherently in 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 available 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 maintenance of routing information. Because the logical trees are meshed, MMT routing protocol provides not only overlapping coverage, but also route robustness while avoiding loops in the meshed topology. Loops are avoided due to the path-vector like property of the VIDs. An optimized version of the MMT algorithm presented in [4] is used to reduce control packets of MMT in the proposed architecture.

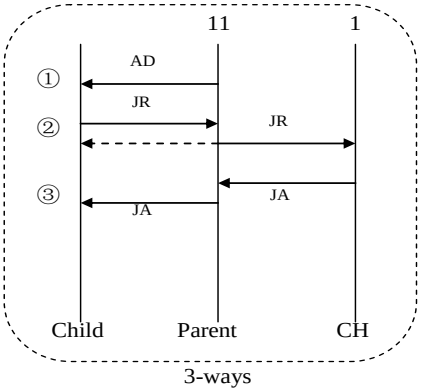
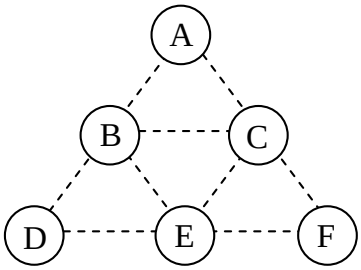


Figure 6. 3-ways handshake in MMT joining process



	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 7. Example of MMT (Hop limit = 3)

1) Multi-Meshed Trees (MMT)

As mentioned above, in the proposed architecture, trees are grown from root nodes (ACPs) to leaf nodes (i.e., sensors and actuators) through the relay nodes. Each meshed-tree can be viewed as a cluster and the ACP as the cluster head (CH) and all other nodes are the cluster clients because data flows in the semi-automated architecture are from sensors to ACPs and ACPs to ACTs. A 3-ways handshake is adopted by nodes to join the meshed tree.

Fig. 6 shows the 3-ways handshake used by a node during the joining process in MMT. The ACP (CH) node initiates tree creation by broadcasting an advertisement (AD) containing its VID. In general, a node on hearing an AD packet and wants to join the tree will send a join request (JR) to the sender of the AD packet who then becomes the parent node eventually to the joining node. The parent then records the new VID in a JR message and forwards to the CH, which register the new VID to its cluster member list. Because the child node can hear the forwarded JR message, the child can

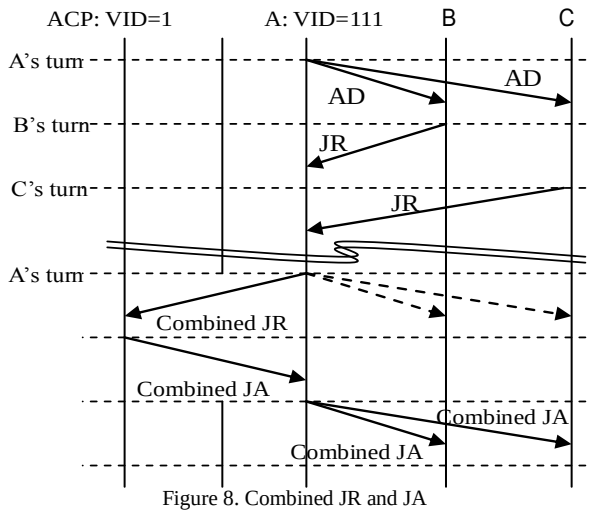


Figure 8. Combined JR and JA

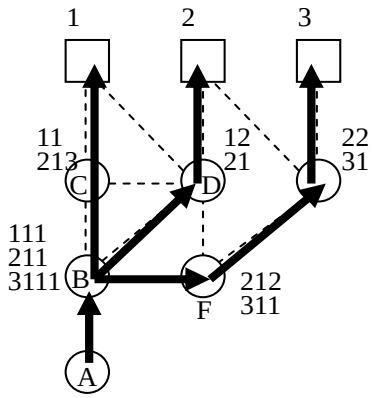


Figure 9. Joining process for multiple CHs

knows 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. 7 shows an example topology and the VIDs allocated using MMT to create the meshed trees. 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 the route to the CH. The process continues until the tree encounters defined limits such as maximum hop count, cluster size or edge nodes are reached.

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 and grandchildren nodes.

As part of the integrated operation of NTT-MAC and MMT routing, the knowledge acquired under the NTT-MAC is used in the MMT joining process by combining the JR and JA during the 3-ways handshake as shown in Fig. 8. Nodes B

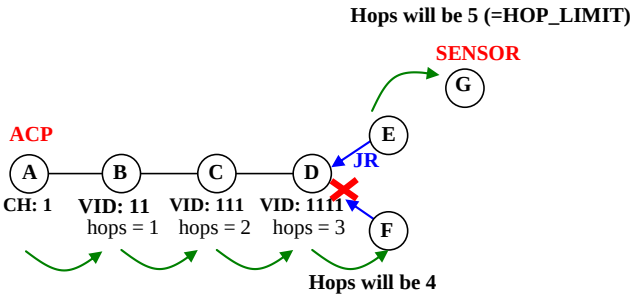


Figure 10. Neighbor Knowledge and Hop Limit

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. Nodes 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 Nodes B and C are the time before Node A's next turn. Hence, the JR 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 JA to Node A after the new VIDs are registered.

The proposed joining process thus allows the request for multiple VIDs with a single 3-ways hand shake process not only for the same CH, but also for VIDs under different CHs. Fig. 9 shows the scheme for joining different CHs using a single 3-ways handshake. If Node A wants to join all VIDs of Node B namely 111, 211, and 3111, Node A sends a single JR, which contains the request to join all VIDs included. Then, Node B assigns new VIDs under all requested VIDs and broadcasts them in JR message. All neighbors of Node B overhear the JR and look into the requested VIDs. If the VIDs contain their direct child VID, they will forward the JR to their CH. For example, Node C will forward the JR because the JR contains request for VID 111, which is a direct child of Node C's VID 11. Likewise, Node D and Node F will forward the JR packet because VIDs 211 and 3111 are their child VIDs.

C. Interaction between Multi Meshed Tree and Neighbor Turn Taking algorithms

Since MMT uses neighbor knowledge for optimized cluster joining process, MMT interacts with NTT to look up neighbor table. This cross-layering approach is thus named as MMT-NTT. Each node maintains neighbor knowledge, which includes not only the node's VID but also the node type. MMT helps set up routes between sensor to ACP and ACP to actuator. Fig. 10 shows an example scenario. When Node E and F receive an AD packet that contains VID 1111 from Node D, they will make decision whether they should send a JR to request a new VID or not. The VID 1111 is 3 hops away from an ACP (Node A). If Node E and F joined this VID, they will be at 4 hops away from the ACP. If the HOP_LIMIT is set to 5, their neighbor nodes will be at the 5th hop (the last hop allowed under the configuration).

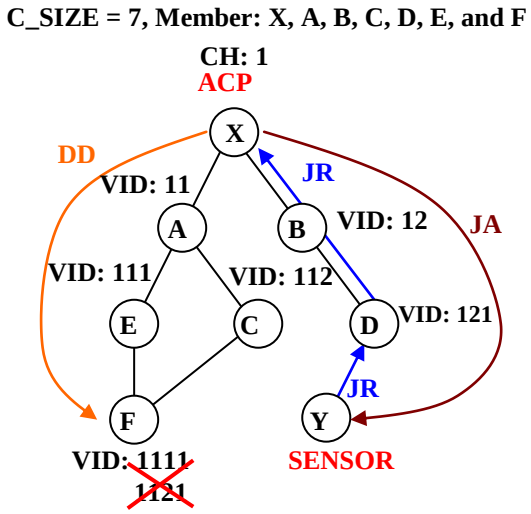


Figure11. Neighbor Knowledge and Cluster Size

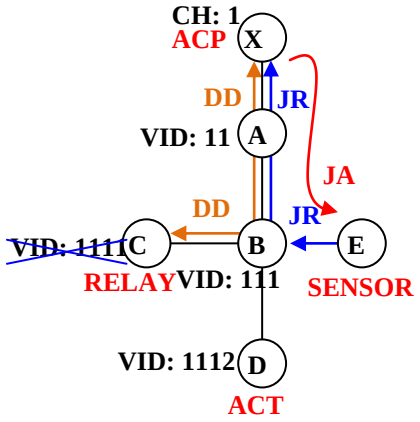
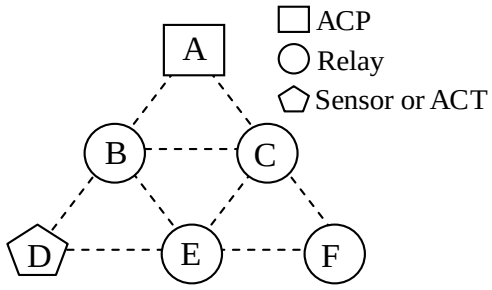


Figure 12. Neighbor Knowledge and Child Size

Therefore, if they do not have a sensor or ACT node in their neighbor list, the new VID will be meaningless and would use up the limited cluster size wastefully. Thus, a node that does not have any sensor or ACT in its neighbors' neighbor list, it will not send a JR if the joining VID is already at HOP_LIMIT−2. In Fig. 10, Node F will not send a JR to Node D because it does not have sensor or ACT in its neighbor list. On the other hand, Node E can join the VID 1111 because a sensor Node G is its neighbor and its newly acquired VID will be 1111x (4hops) and the new VID for Node G will be 1111xx (5hops).

In addition to the hop limit, ACP (CH) can also limit its cluster size (C_SIZE) based on the topology and number of total nodes in the network. Therefore, an ACP will prioritize inclusion of more sensors and ACTs within the C_SIZE limit. MMT-NTT considers priorities to achieve this. If the total number of nodes in the cluster has reached C_SIZE limit, the CH can replace low priority node such as relay nodes to accommodate high priority node such as sensor and ACT. In this case, a CH creates a Direct Delete (DD) packet to tell the low priority node to delete all VIDs related to that



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

TABLE I. MMT CONTROL OVERHEADS COMPARISON

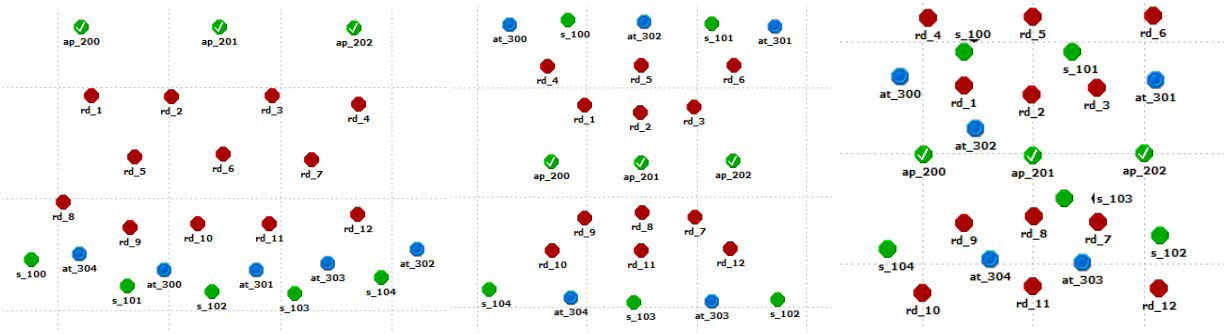
Total number of	VID	AD	JR	JA	Overheads
Original MMT	21	8	30	30	68
MMT-NTT with combined JR JA	11	5	13	10	28

Topology used in Fig. 7 and Fig.13 is used for this comparison

CH. Fig. 11 shows an example scenario for it. When the C_SIZE of the ACP Node X is limited to 7 and it has already reached this value with the member Nodes X, A, B, C, D, E, and F. Sensor Node Y wants to join the cluster, Node Y sends a JR to Node D, and then Node D forwards the JR to the CH, Node X. This JR is accepted by the CH even though it has already reached C_SIZE because Node Y is sensor node and it has higher priority than relay nodes in the cluster. Node X (CH) sends a DD packet to Node F to remove it from the cluster. Thus Node F removes VIDs 1111 and 1121 from its subsequent AD packets. Meanwhile, Node X sends a JA to Node Y.

There is also limit to the maximum number of child nodes under a relay node, based on node density, given by the variable CHILD_SIZE to reduce the number of VIDs. The rationale for this assumption also arises from the fact that if a node has too many children, it could result in a bottleneck. A relay node will give priority in accommodating more sensor and ACT nodes within the CHILD_SIZE limit. Fig. 12 shows an example scenario. When Node B has already accepted the maximum number of children under VID 111 and a sensor Node E wants to join the VID 111, Node B checks Node C's neighbor list and finds that there are no ACT and sensor in this list. Node B can then send a DD message to Node C under the VID 111. Node B will also send a DD packet to Node X to deregister the VID assigned to Node C. Meanwhile, Node B accepts the JR from Node E.

Fig. 13 shows optimized MMT for a sample topology. Sensor and ACT do not support child nodes, so Node D does not have a child VID. Because Node F does not have any sensor and ACT in its neighbor list 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



a) Layered b) Middle c) Random

Figure 14. Relative placement of sensors (s_number), actuators (at_number), relays (rd_number), and ACPs (ap_number)

reduced significantly because total number of VIDs in use is reduced. On the other hand, NTT interacts with MMT to identify sender and destination nodes from the VIDs and to calculate the turn scheduling from neighbor table and its own VIDs.

Table I shows the total number of VIDs assigned by the original MMT and the MMT-NTT that combines JR and JA messages based on the identical topologies in Fig. 7 and Fig. 13. Total number of control packets, i.e., AD, JR and JA is significantly reduced in the MMT-NTT that combines JR and JA messages. Reducing control packets in WSAN and WSANIC reduces number of collision and also achieves low latency in data transfer.

D. Security

As the trees of MMT are rooted at an ACP, the ACP can be used to authenticate the sensors, actuators and relays as they join its tree. In addition, MMT has the following security features.

1) Route spoofing [19] is a common security issue faced by reactive routing protocols. MMT being a proactive routing protocol does not face this issue. Furthermore, during MMT route setup, each node has to register with the ACP, which can employ efficient authentication schemes before admitting nodes to join its tree.

2) Impersonation [20] is easily detected in MMT due to the locality property of the VIDs. If a malicious node A is close to B and learns B's VID by eavesdropping; there are limited ways in which A can use B's VID. If A assumes B's VID in its vicinity, B would recognize this and report (via one of its alternate routes using a different VID) to ACP and the ACP could challenge A. If A takes B's VID and moves away from B to use it, then the VID is invalid because of the locality property of the VIDs. Node A could wait till B moved away and then use the VID, but when B moves away it will acquire and report a new VID to the ACP, and the ACP will know that A is misusing B's VID.

3) Denial of Service (DoS) attacks [21] can be acted upon if the DoS origination point can be located. The affected area can then be quarantined to restrict adverse effects in the rest of the network. In the MMT-based approach, DoS due to flooding or jamming will result in several route failure reports to the ACP. Based on the failure reports in the affected area, the ACP can determine a virtual boundary (of VIDs) of the affected area and isolate that area.

4) Black hole [22] problems are encountered when malicious nodes do not forward incoming packets. An explicit acknowledgement may not resolve this problem as the malicious node can send an acknowledgement for every received data packet without forwarding it. MMT builds routes on links that are bidirectional. At the MAC, forwarding of a data packet can be used as an implicit acknowledgement to the previous sender of the packet and this type of acknowledgement can be used till the packet reaches the destination node, at which point an explicit acknowledgment has to be used. When a node repeatedly fails to forward packets, the parent node reports this to the ACP, which declares that route obsolete by using alternate routes to inform the sensors and actuators that have a route via the defaulting node. A further advantage of using implicit acknowledgements is reduction in the number of transmitted messages.

TABLE II. SIMULATION SETS

	ACT	Sensor	Relay	ACP
NTT-MMT (5)	**5**	**5**	12	3
802-DSR (5)				
NTT-MMT (10)	**10**	**10**	12	3
802-DSR (10)				

TABLE III. SIMULATION SETTINGS

Data size	Data rate	Duration	Transmission	Data Generator
500 bits	0.05 sec	5.0 sec	11 Mbps	ACP, Sensor

VI. ANALYSIS RESULTS

A. The Evaluation Topology and Simulation Scenarios

Fig. 14 shows the topologies used in the OPNET simulations [23] to evaluate the proposed scheme. The

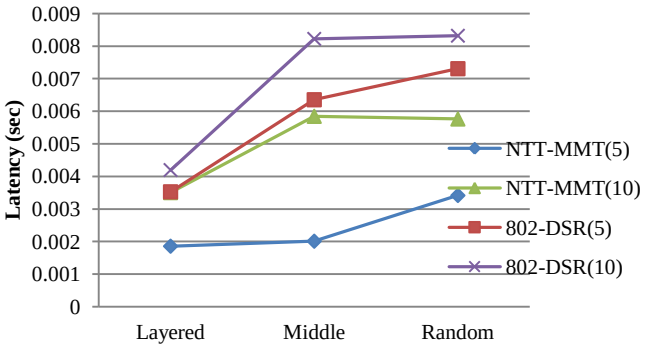


Figure15. End-to-end Latency

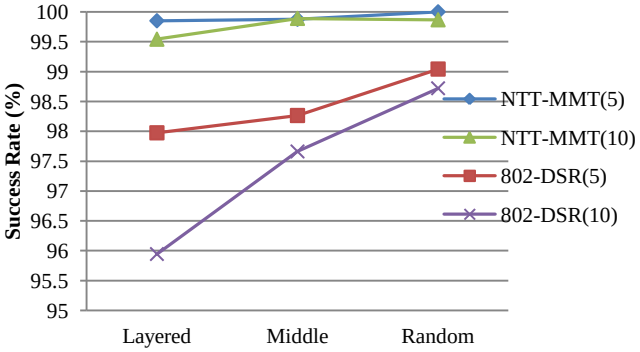


Figure16. Success Rates

topologies show relative placement of the sensors and actuators with respect to the ACPs, which is similar to the semi-automated industry architecture discussed earlier. Node name starting with s_, *at_*, *rd_*, and *ap_* are sensors, ACTs, relays, and ACPs respectively. Nodes in Fig. 14 (a) are placed in a layered manner where the top layer has ACPs, middle layer has relays, and bottom layer has sensors and ACTs. In Fig 14 (b), ACPs are placed in the middle and sensors and ACTs are placed at the edges. In Fig 14(c), all nodes except ACPs are placed randomly.

Several sets of simulations runs were conducted and each set is recorded in Table II. Each set was conducted on the three topologies described in Figure 14(a), (b) and (c). Each simulation was run for 5 seconds, and was repeated for 5 different seeds. Table III records the simulation setting. At the ACPs and the sensors, data was generated at the rate of one packet in 0.05 seconds, with a packet size of 500 bits. The transmission data rate was set to 11 Mbps. Data from sensors were sent to one of the three ACPs and data from ACPs were sent to all of the ACTs. Thus, in the 5 sensors and 5 ACTs scenario, a total of 2000 data packets (5 seconds / 0.05 packets * 5 sensors + (5 seconds / 0.05 packets) * 3 ACPs * 5 ACTs) can be transmitted and a total 4000 data packets can be transmitted in the 10 sensors and 10 ACTs scenario if routes between sensors and ACPs, and ACPs and ACTs are fully maintained by routing protocol.

The proposed architecture, which supports an integrated NTT-MAC and MMT routing protocol, called NTT-MMT is compared with a similar architecture using 802.11 CSMA/CA MAC and DSR routing protocol, called 802-DSR in the plots.

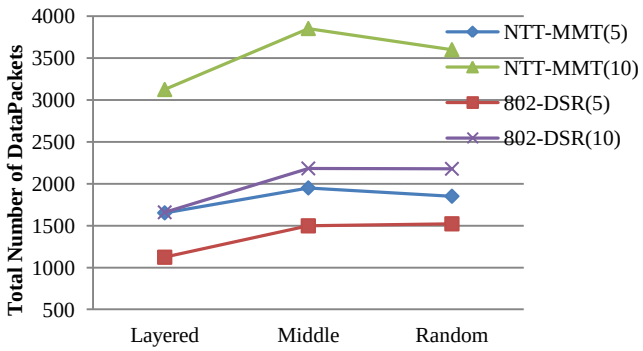


Figure17. Total number of Data packets

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) Average 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.

3) Average total number of sent data is also recorded to provide data on the traffic loads in the scenario.

C. *Simulation Results*

Fig. 15 shows results of average end-to-end latencies. Latencies of both NTT-MMT and 802-DSR have increased in the 10 sensors and 10 ACTs scenario as compared to the 5 sensor and 5 ACT scenario. This is because the number of nodes and generated data packets are increased for the same simulation field size. Therefore, the 10 nodes scenario is more congested than the 5 nodes scenario.

Fig. 16 shows results of average success rates. Both the 5 nodes and 10 nodes scenario show high success rates in the NTT-MMT scheme. The reason that the random topology has higher success rate compared to others is because the number of hops between sensors and ACPs, ACPs and ACTs are smaller in this topology.

Fig. 17 captures average total number of sent data packets. NTT-MMT sent 1.5 times more data packets than 802-DSR because NTT-MAC has a better utilization of the wireless medium compared to IEEE 802.11 CSMA/CA and MMT routing maintains more routes than DSR routing. Based on the data rate of packet generation, 2000 and 4000 data packets can be generated in 5 seconds simulation for the 5 nodes and the 10 nodes topologies. NTT-MMT could process 91% and 88% of the maximum number of data packets generated for the 5 nodes and the 10 nodes scenario respectively. On the other hand, 802-DSR could process only 69% and 50% of the maximum number of data packets generated.

The NTT-MMT achieves high success rate and low latency at the same time. In addition, NTT-MMT could send more data packets. Robustness in NTT-MMT is high because success rates of NTT-MMT remains high irrespective of the

different topologies and in highly congested network situations.

VII. CONCLUSIONS

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 and higher success rate. We also introduced a routing protocol based on the MMT algorithm, which is a proactive routing protocol along with the NTT-MAC. MMT is developed to support high route robustness with a quick and easy forwarding approach based on virtual IDs. In industry control, Wireless Sensor-Actuator Ad-hoc Network using NTT-MAC and MMT-routing will provide superior performance. The performance metrics focused were success rate, packet delivery latency, and number of delivered data packets. The simulation results show improved performance of NTT-MMT in terms of success rate and end to end latency compared to DSR operating with IEEE 802.11 CSMA/CA MAC.

REFERENCES

[1] Y. Nozaki, N. Shenoy, and Q. Li, "Wireless sensor actor networks for industry control," ICNS 2013, The Ninth International Conference on Networking and Services, pp. 172-178, 2013.

[2] 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, PIMRC 2007, IEEE 18th International Symposium on, pp. 1-5, 2007.

[3] E. F. Golen, Y. Nozaki and N. Shenoy, "An analytical model for the neighbor turn taking MAC protocol," Military Communications Conference, MILCOM 2008, IEEE, pp. 1-7, 2008.

[4] 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, GLOBECOM 2005, IEEE, vol. 6, pp. 3351-3356, 2005.

[5] 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 December 2013)

[6] 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 December 2013)

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

[8] 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 December 2013)

[9] "ISA-100 wireless system for automation," http://www.isa.org/MSTemplate.cfm?MicrositeID=1134&CommitteeID=6891. (accessed December 2013)

[10] 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, pp. 21-26, 2002.

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

[12] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," Ad Hoc Networks, vol. 2, pp. 351-367, 2004.

[13] L. Barolli, T. Yang, G. Mino, F. Xhafa and A. Durresi., "Routing efficiency in wireless sensor-actor networks considering semi-automated architecture," Journal of Mobile Multimedia, vol. 6, pp. 97-113, 2010.

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

[15] IEEE Computer Society LAN MAN Standards Committee. "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1997.

[16] V. C. Giruka and M. Singhal, "Hello protocols for ad-hoc networks: overhead and accuracy tradeoffs," World of Wireless Mobile and Multimedia Networks, WoWMoM 2005, 6th IEEE International Symposium on, pp. 354-361, 2005.

[17] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks," Ad Hoc Networks, vol. 5, pp. 139-172, 2001.

[18] T. Clausen, and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626, IETF Networking Group, 2003.

[19] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," Wireless Communications, IEEE, vol. 14, no. 5, pp. 85-91, 2007.

[20] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks," Network Protocols, 2002. Proceedings. 10th IEEE International Conference on, pp. 78-87, 2002.

[21] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," Computer, vol. 35, no. 10, pp. 54-62, 2002.

[22] M. Al-Shurman, S. M. Yoo, and S. Park, "Black hole attack in mobile ad hoc networks," In Proceedings of the 42nd annual southeast regional conference, ACM, pp. 96-97, 2004.

[23] "OPNET modeler," http://www.riverbed.com/products-solutions/products/network-performance-management/network-planning-simulation/Network-Simulation.html. (accessed December 2013)