Received January 30, 2017, accepted February 22, 2017, date of publication March 14, 2017, date of current version April 24, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2682108

The Dynamics of Data Packet in Transmission Session

ADAMU I. ABUBAKAR, (Member, IEEE), ELBARA ELDAW ELNOUR MOHAMED, AND AKRAM M. ZEKI, (Member, IEEE)

Department of Information Systems, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia

Corresponding author: A. I. Abubakar (adamu@iium.edu.my)

This work was supported by the International Islamic University Malaysia Research Grant through the Foundation Research Grant Scheme under Grant FRGS15-245-0486.

ABSTRACT This paper presents a coordinated framework of prepared data packet transmission in a transmission session. Automated guide vehicles (AGVs) with an onboard 1-km range APC transmitter are used to establish the network environment for testing the prepared packet transmission session. An evaluation technique of the dataflow through the network was carried out. The captured transmission parameters offer an inbound and outbound limit of dataflow through which the AGVs communicate with each other. The findings of this paper indicate that the ideal transmission rates at the frequency of 434 MHz and at a baud rate of 19200 bps was found to be 25 ms in between each 56-B data packets. The range of distance between the transmitter and the receiver was 0.5 to 970 m. The transmitter had a 1-km range limitation. As the applications for which the communication model was developed did not require long distances, the transmitter's range limitation was sufficient. Further tests validated these transmission session parameters values. The proposed techniques can be implemented as logistic and automation solutions in the manufacturing industry.

INDEX TERMS Data packet, transmission session, dataflow, APC transmitter.

I. INTRODUCTION

Coordination constitutes a key aspect of network systems which makes it highly versatile and scalable in any communication environment. Networked devices and programs provide distributed avenues that are used for industrial application where coordination constitutes a major task in the automation process, especially in the context of modern manufacturing operations [1]. All aspects of industrial automation are highly important and have to agree on a set of engineering tools [2]. In any manufacturing process automation is necessary, especially at the initial stage where raw materials are delivered to the processor(s) or at the final stage of manufacturing where the finished products are delivered to storage which requires the directed movement of loads [3]. In a big manufacturing company, the major goal is the efficient and speedy transportation of any type of material (raw or finished materials or work in progress) in and out of the company's environment. This activity requires a scalable and flexible coordinated network of carriers. There exists a high demand of automation in the pharmaceutical industry where it assists in driving labs into the automated world of industrial production lines [4]. The Automated Guide Vehicle (AGV) has become the most important transporter used in the manufacturing process in order to move materials from one place to another [5].

Companies utilize AGVs in their activities in order to increase productivity and minimize human. error [6]. This innovation has also created a better working environment. The main feature of the AGV consists of driverless transportation controlled by a computer program [7]. It can be used in both interior and exterior environments. In manufacturing, AGVs are used to transport all types of materials related to the manufacturing process. Being an unmanned vehicle, its major navigation technique can be either closed path or open path [8]. The closed path is based on a fixed outline drawn on the ground which the AGV sensor detects and follows as its path. This navigation technique has been judged as not sufficiently dynamic and flexible since any alteration of the fixed path requires a new outline on the ground [9]. Open path navigation has been invented in order to overcome the weakness of the closed path [10] as it does not depend on a ground outline on a physical path. Here, the open path the AGV follows is programed on the controlling unit which allows for dynamic path updates and changes. The controlling units mostly rely on certain preexisting dimensions assigned as AGV paths. Thus, high flexibility is achieved, and the AGVs assigned to any path can be redirected while on transit. However, although open path navigation is more flexible, it lacks the interactive harmonization among two or more AGVs [11] and it does not provide an interactive coordination among two or more AGVs. Crucial to this is the network of dataflow within the AGVs and the controlling program. Hence, under certain circumstances this may lead to risking data loss if multiple AGVs in the network are performing an important task [12]. In cases where there are many AGVs in a network and the tasks at hand are more than the number of AGVs, the processing of the dataflow certainly becomes larger and the centralized controlling program will find it difficult to route and broadcast to the individual AGVs. In cases where the manufacturing process is large and the number of load or unload points is high, the open path navigation process is likely to become extremely time consuming [11].

The internetwork communication among AGVs is aided by onboard wireless devices (transmitters). If the AGVs are able to communicate with each other, the effectiveness of flexible navigation is enhanced which in turn assists the industrial automated handling of materials being transported from one point to another. The major issue is how effectively and accurately the controlling program coordinates the dataflow among the individual AGVs. A major constraint constitutes the way in which the controlling program guarantees that the sent information arrives or is received within precise time-bounds. This problem has been part of the two major challenges in designing AGV systems in connection with transporter routings and flow-path networks. Both have to be tackled at the same time as one problem is linked to the other [13]. Owing to this fact, this paper proposes to design a coordinated AGV architectural framework and to develop a prototype central monitoring system that efficiently manages, monitors and controls the AGVs through wireless communication

II. RELATED WORK

Recent AGV research has focused mainly on the functions of moving materials and products. This has lead the way towards converting the role of the towing vehicle into complex material-handling transport vehicles ranging from mail handling AGVs to automatic trailer loading AGVs [14]. The various techniques applicable in this respect were studied, and more and more techniques for design and navigation technologies were proposed. A critical issue of the AGV functions is its flow system design [15] that is tied to its routing functions within the flow path and station location design. The first integer programming model for material flow path design was developed by Gaskin and Tanchoco [16]. Thereafter, an alternative model was proposed where the station locations were no longer fixed and instead restricted to the nodes on the boundary of the cells as described by Goetz and Egbelu [17]. A Bayesian approach was used to determine the loss function of moving a AGV routing network by Fazlollahtabar and Saidi-Mehrabad [12]. A cooperative co-evolution with a genetic algorithm was subsequently suggested to solve the problem of transporter routings and flow-path network of AGVs [13].

Other studies were conducted on AGV scheduling of navigation whereby several AGVs were required to work in a certain place while non-conflicting with each other and conflict-free routing techniques were proposed [18]–[20]. Later, in 2006, Kim *et al.* [21] proposed a deadlock detection and prevention algorithms for AGVs. The idea was based on the assumption that preexisting grid blocks were designed to prevent collisions and deadlocks among AGVs. The fuzzy neural network controller technique was used in Wuwei *et al.* [22], while Basnet and Mize [23] and Rachamadugu and Stecke [24] provided new insights into fuzzy logic control and the artificial potential field. Han and McGinnis [25] developed a real time algorithm which considered material handling transporters. Schriber and Stecke [26] showed to which extent the additional consideration of the material handling system and limited buffers degraded system performance. Sabuncuoglu and Hommertzheim [27], [28] highlighted the importance of material handling and compared several AGV dispatching rules. A combination of topological and grid maps plan layout procedures was used for navigation where the A* Algorithm generated low-level paths [9]. Parametric technology was also used in the AGV design [10]. A model of deadlock avoidance for automated manufacturing systems with multiple AGV was proposed in Wu and Zhou [8].

They also showed how the buffer capacity can affect the performance of the system. Routing flexibility (i.e., alternative machines and processing routes) was considered by Wilhelm and Shin [29], Chen and Chung [30], and Khoshnevis and Chen [31]. Their studies indicated that dynamic routing (i.e. a path determined dynamically during schedule generation) performed better than preplanned routing.

The predictive methods were aimed at finding optimal paths for AGVs, the conflicts were predicted off-line, the AGV routes being planned to avoid collisions and deadlocks [19]–[21].

III. METHODOLOGY

The aim of this work is to construct three AGVs with onboard wireless modules and to develop a controlling program for the master node (controller) and three slave nodes (AGVs). This is intended to be established based on certain real-life network scenarios where the transmission parameters in bytes or packets can be captured and analyzed. Specifically, all the instructions required for the remote functioning of the AGVs need to be determined. Among those critical instructions is the information involving decisions about the nodes the master node intends for each transmission and the hand scheduling and maintenance of the instruction flows. Furthermore, decisions on instructions from the master node which needs to be transmitted in sequence to each node can also be captured and analyzed. The constructed AVG

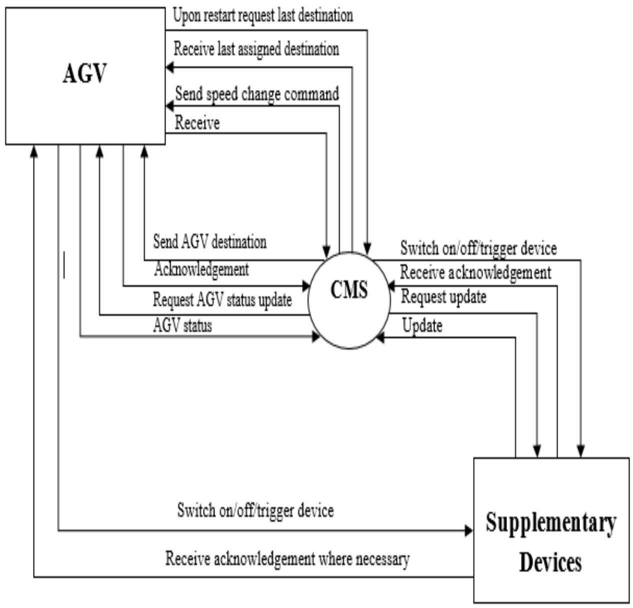

FIGURE 1. The proposed system architecture.

prototype is used for the networked environment to establish the scenarios outlined above.

A. SYSTEM ARCHITECTURES

The system architecture of the proposed coordinated AGVs networked environment (see Figure 1) consists of two main components which are as follows: Master node – a Java program to act as a central monitoring system on a computer and the slave nodes - AGV/AGV model/supplementary devices. The AGV to be constructed consists of the entire building block of the vehicle assembled from various components, namely chassis components, motors and wheels components, Arduino controller, wireless transmitters, SLA batteries, sensors or leds or buzzers. The Central Monitoring System (CMS) consists of the master node running on a windows-powered device with a wireless transmitter. The Java program is designed on the CMS for scheduling, managing, monitoring and controlling the entire networked environment. The supplementary devices are set up in the form of wireless enabled devices connected to the system that supplement the AGV system functions. These include alarm modules to alert an operator when the AGV reaches a station as well as auto gates that are triggered when the AGV crosses a walkway. All these devices are connected to the system wirelessly to receive a signal from the AGVs or CMS

The fact that the system architecture is designed for a coordinated networked environment, the CMS listens to four events from the AGVs, namely ''AGV status'', ''Acknowledgement'', ''Receives request'', and ''Repeat from last destination report''. The CMS responds to each operation it listens to from the AGV in sequence starting from ''Request AGV status update'' to ''Send AGV destination'', ''Send speed change command'' and ''Received last assigned destination''. Similarly, the CMS listens to supplementary devices in the same fashion as it does for the AGV. Furthermore, the AGVs and supplementary devices interact when the AGV is switched on or off or triggers a device, and the supplementary devices acknowledge the startup.

A Java programming platform is used to facilitate the creation of the CMS (master node) for managing the AGV models (slave nodes). This constitutes the central monitoring system as described in the previous section. The AGV model is constructed using the materials presented in Figure 3. In order to program the microcontrollers needed to interface with sensors and actuators and operate the AGV models, an Arduino complier is used. The most important components necessary to build the AGV model include sensors, actuators, electronics and drive modules. Thus, Arduino microcontrollers are used for operating the AGV models and other hardware components. APC 220 wireless transmitters (see Figure 3) are used as wireless transmitters for data transmission. Here, the APC220 transmitter serves as the wireless transmitter that allows communication between the AGV model (slave node) and the master node to manage communication as detailed above. The Arduino Uno Microcontroller is the microcomputer running on the AGV that allows the AGV to move, navigate and manage its wireless communication. A sealed lead acid battery (12 V) is used to power all components of the AGV model. A multicolor LED (Light Emitting Diode) module is used as a visual cue to display the AGV status, and a L293d motor driver is used to drive the motors of the AGV model. It allows the microcontroller to control motor speed and direction of rotation. The Digital Compass is a sensor in the AGV which shows the current orientation or direction that the AGV is facing. The RGB Color Sensor is used to detect color, more specifically color cards placed along the AGV path to issue commands to the AGV or indicate stops along its path. The Rotary Encoder consists of a sensor that is normally used to determine the speed of rotation or the angular displacement of a wheel. Motors and wheels constitute the driving unit in two sets that are used to drive the unit. The method of driving follows the differential navigating method. Finally, the robot body platform consists of the main platform which makes up the AGV prototype's body and upon which all components that make up the AGV are attached.

The coordinated transmission algorithm was formulated considering a typical two or more AGVs (slaves notes) and the CMS in a transmission session. The transmission parameters (transmission status, transmission delay, transmission type and node ID) were defined for scenarios within a practical multi-APC220 transmitter wireless network, where each AGV node had a single antenna. Each APC220 transmitter of the AGV node was controlled by the CMS. Thus, the initiation of the data transmission was set ready to read transmission. Thereafter both send and received routines were set to be carried out.

B. DEVELOPMENT OF THE PROPOSED ALGORITHM

The algorithms formulated encompasses the behavior of the transmission session, the entire variables involve are capture during the transmission. Crucial variable in this algorithms is the degree of delays necessary in to maintain transmission between the network nodes for stable communication to be achieved. Thus, six algorithms were proposed. Algorithm 1 and 2 define the continuous transmission to determine the fastest rate while Algorithm 3 and 4 express transmissions with response, Algorithm 5 and 6 dwell on coordinated transmission with response. The algorithm leads APC (controller) running a master java program and a wireless USB module. The AGV model uses different capacities as slave nodes running their own programs for the duration of the transmission session.

The data packet used is a string containing the alphabet as follows: ''*ABCDEFGHIJKLMNOPQRSTUVWXYZ#''. This packet has to be sent between the transmitters and receivers. The master java program is acting as a wireless ''sniffer'' tool node placed in the network and act as a network analyzer in order to observe and analyze the data transmitted within the network. These parameters include information such as how many data are transmitted within a certain time as well as how many and how long are the delays between transmission sessions within a certain period of time. Data and information captured by the sniffer node and the PC are used to determine the fastest achievable packet transmission rate and hence the lowest delay in between packet transmissions. Variables that normally affect these are the distance between the two wireless communication modules (normally one would use the furthest distance to simulate the worst possible scenario). The baud rate of transmission and the frequency of the transmitters.

Algorithm 1 and 2: Continuous transmission to determine fastest rate:

Algorithm 1 and 2 were proposed to established a transmission rate and a threshold in a continued transmission medium in one-way, that is from the slave node to the master node. The major variables involved are ''distance'' and ''delay'', the distance set in the field experiments starting from 0.5, whereas the delay was initialized in line two of algorithm 1. The data send by algorithm 1 within certain distance from the slave node and master node, were verify by the data received by line 2 to 4 in algorithm 2. Hence, both algorithm 1 and 2 establish a one-way and acquire details involved in the transmission session.

Algorithm 1
(Slave Node)

1. ***Begin***
2. ***Initialize count = 0,transmit_delay = 25;*** *//the transmit_delay is adjusted everytime to find the ideal delay value*
3. ***if****(****count*** *<= 400)*
4. *wirelessSerial.print("****test_packets#****");* ***//transmit test packet***
5. *delay (****transmit_delay****);*
6. ***count++;***
7. ***end if***

Algorithm 2
(Master Node)

1. ***Begin***
2. ***while****(****wireless.available()****) //if wireless data is detected*
3. ***if****(****data is correct****) //verify data integrity*
4. ***process data for results;*** *//here another function processes data*
5. ***end if***
6. ***end while***

Algorithm 3 and 4: Transmission Test B – (Transmission with response)

Algorithm 3 and 4 are proposed in order to examine the transmission session behavior when the master node sent back the response to the slave node. Thus, at algorithm 3, data packets are sent from the slave node, and at algorithm 4, a response massage is sent. This allows to determine the delay on the side of the master node

Algorithm 3
(Slave Node)

1. ***Begin***
2. ***Initialize count = 0;***
3. ***while****(****wireless.available()****) //if wireless data is detected*
4. ***if****(****data is correct****) //verify data integrity*
5. *wirelessSerial.print(''****test_packets#****''); //transmit test* ***response packet***
6. ***end if***
7. ***end while***

Algorithm 4
(Master Node)

1. ***Begin***
2. ***Initialize count = 0;***
3. ***while****(****wireless.available()****) //if wireless data is detected*
4. ***if****(****count****<= 400)*
5. ***if****(****data is correct****) //verify data integrity*
6. *wirelessSerial.print(''****test_packets#****'');* ***//transmit test packet***
7. *counter++;*
8. ***process data for results;*** *//here another function processes data*
9. ***end if***
10. ***end if***
11. ***end while***

end;

Algorithm 5 and 6: Co-ordinated transmission with response

Algorithm 5 and 6 lead multiple slave nodes to send to each other and to the master node. Similar to algorithm 1 and 2, the main variables involved are ''distance'' and ''delay''.

Algorithm 5
(Slave Node)

1. ***Begin***
2. ***Initialize count* = *0,Agvid* = *x;***
3. ***while***(***wireless.available()***) *//if wireless data is detected*
4. ***if***(***data is correct***) *//verify data integrity*
5. ***if***(***data is intended for this Agvid***)
6. *wirelessSerial.print("****test_packets#***"); //transmit test* ***response packet***
7. ***end if***
8. ***end if***
9. ***end while***

Algorithm 6
(Master Node)

1. ***Begin***
2. ***Initialize count* = *0,AgvCount=4, Agvid=1;***
3. ***while***(***wireless.available()***) *//if wireless data is detected*
4. ***if***(*count* <= *400*)
5. ***if***(***data is correct***)
6. *wirelessSerial.print("****Agvid*** ****test_packets#***"); //transmit* ***test packet with Agvid attached***
7. *counter++;*
8. ***Agvid*** = *counter%* ***AgvCount*** *+**1;***
9. ***process data for results;****//here another function processes data*
10. ***end if***
11. ***end if***
12. ***end while***
13. ***end;***

IV. EXPERIMENTAL TEST

The experiment was carried out in an indoor area with a clear line of sight within three AGVs and a window tablet PC. All attempts were made to avoid electromagnetic interference where possible. The slave node was running the microprocessor code written in the Arduino language and is the constructed AGV customized for the tasks required in the experiments. The master node for these tests are laptop and tablet. The first is a windows 8.1 tablet with an Intel Atom Z3735F operating at 1.33 GHz with a wireless module attached to the USB port (see Figure 2). The second is a high performance laptop with an Intel i7-6500U processor capable of 3.1 GHz with the wireless module attached to the USB port (see Figure 2). The ram observed while running the application through the help of the windows task manager was found to not exceed the 150 megabyte mark after about an hour of usage of the application. Thus, any modern device should not have ram limitations as most devices come with over 1 gigabyte of available ram. The testing of two different types of devices here is important to see whether a device with faster processing capabilities is able to receive data at a faster rate or not and if so, whether it will be a significant amount to take note of or not. Both master nodes are running the java application that were used to conduct the tests.

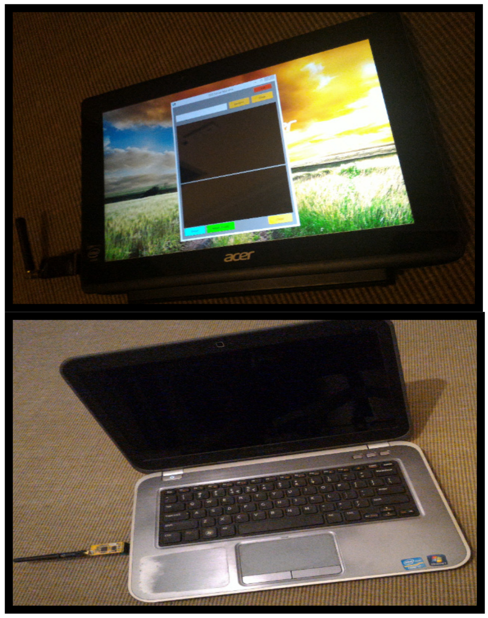

FIGURE 2. Master node tablet PC and laptop PC.

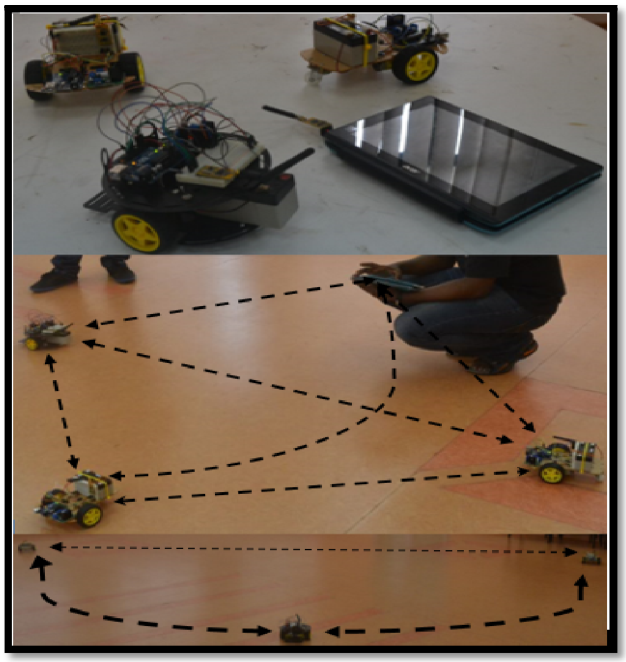

FIGURE 3. The coordinated transmission session.

The experimental setup is shown in Figure 3. In the upper section of Figure 3, are shown the three AGVs prototypes constructed and developed for the research. Each component ists of a APC220 wireless transmitter and a window tablet PC containing its controlling program. In the lower section of Figure 3 is shown the experimental setup containing the three AGVs in an indoor field. The middle section of Figure 3 contains the experimental scenario where the coordinated transmission scheme design on the controlling

TABLE 1. The result of experimental transmission session.

Test	D (m)	ET (kb)	Suc (kb)	TTT (sec)	MaxD (ms)	MinD (ms)	ATTD (ms)	PD (ms)	APPD (ms)	ADPS (kb)	TDT (kb)	TP (kb/s)
A1	0.5	400	68	18.502	282	44	276	15	261	58	3.944	0.213
A2	0.5	400	260	20.602	296	50	79	20	59	58	15.08	0.7281
A3	0.5	400	400	22.576	59	49	56	25	31	58	23.2	1.028
A4	100	400	400	22.583	61	53	56	25	31	58	23.2	1.027
A5	970	400	102	7.47	1360	46	73	25	48	58	5.916	0.793
A6	660	400	400	22.587	62	52	56	25	31	58	23.2	1.027
A7	0.5	400	400	22.584	74	38	56	25	31	58	23.2	1.027
A8	660	400	400	22.583	68	48	56	25	31	58	23.2	1.027

D(m) = Distance in meters, ET(kb) = No of expected transmissions in kilobits, Suc(kb) = No of successful transmissions in kilobits, TTT(sec) = Total time Taken in seconds, MaxD(ms) = Maximum detected delay in milliseconds, MinD(ms) = Minimum detected delay in milliseconds, ATTD(ms) = Average Total Transmission delay in milliseconds, PD(ms) = Programmed delay in milliseconds, APPD(ms) = Average Propagation and Processing delay in milliseconds, ADPS(kb) = Average data packet size in bytes, TDT(kb) = Total data transmitted in kilobytes and TP(kb/s) = Throughput in kilobytes per second.

program was launched. At stationary, a packet was send as an instruction to the AGVs, and the data within the transmission session were captured in the window tabled PC over various distances.

One may argue that in order to prove the feasibility of the proposed algorithm, the experimental designed should consider the issue of scalability, in terms of several tens of vehicles for a real-life case, like in a warehouse which usually has more than several tens of vehicles in operation. Technically, the numbers do not matter, since the controlling program is not the main focus of this study but rather the flow and behavior of data within the transmission session and the technical constraints of the transmission medium. Once a transmission session is initiated from a master node to the slave nodes, the feedback of this experiment is to capture the dynamics of the instructions sent in a window tabled PC over various distances

Experiment test A1 relies on the implementation of Algorithm 1 and 2. The transmission is described as the simplex transmission, because is a one-way transmission session established by the slave node. Packets data are transmitted to the master node from the slave node, and the master node captures the transmission session parameter. This transmission test is a repeated string (400 times), transmitted by the slave node to the master node where the master node verifies each transmission instance and displays the results of the test runs. The transmitted data packet string are as follows:

$$^{*}1101, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z\#$$

The processing of transmission session parameters in the master node involved calculation of transmission delays, amount of data received, time taken etc. The battery powered AGV (slave node) is placed in a stationary position and switched on. The user then carries the Laptop/Tablet (master node) running the java application for the transmission session tracking to the different distances and initiates the test using the following command string to initiate the test:

$$^{*}\mathbf{1101}, \mathbf{start1}, \mathbf{20\#}$$

The value "20" at the end indicates the desired delay, this string is received and interpreted by the slave node and processed and the stated delay is incorporated in the ensuing test which begins immediately after the string is received. This method allows the user to conduct the test from multiple locations easily and rapidly while varying the delay variable without having to go back to the slave node.

When the test begins, the user then waits with the master node stationary at the chosen distance and observes the screen on the laptop or tablet until transmission stops, many transmission session parameters has been captures in many different distance, however, some of the sample captured transmission session are collected. The transmitted data packet string sent to the slave nodes in the coordinated duplex transmission session, experiment is as follows:

$$^{*}\mathbf{1101}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{I}, \mathbf{J}, \mathbf{K}, \mathbf{L}, \mathbf{M}, \mathbf{N}, \mathbf{O}, \mathbf{P}, \mathbf{Q}, \mathbf{R}, \mathbf{S}, \mathbf{T}, \mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{Nx\#}$$

Where the "**x**" in **Nx** will be replaced with the node number

V. PRESENTATION OF THE RESULTS

The entire algorithms proposed were implemented and experimental analysis was carried out to determine their impact on various degrees of transmission sessions. In this regards, various aspects of centralized, multi-nodal communication model associated with congested network environments are established. This has been possible with the use of some builds AGVs as the network nodes equipped with APC transmitters. The packet data used for transmission test are string of alphabets. The packets were send between the transmitters and receivers, in one-way and two-ways. A java implementation of the entire algorithm act as a network analyzer in order to observe and analyze the data transmitted within the network. After the experimental analysis performed, the summary of the gathered results was presented in Table 1.

The result of the first three experimental test (Test A1-A3), which were carried out at same distance (0.5m) within the slaves and master node, but with different programed delays, indicates that the expected packet data supposed to be received by the master node which were sent from slave node are not received in the first and second test, despite undertaken the experiment in a different programmed delays. Out of 400 packet data expected to be received by the master node, only 68 were received for the first test when programmed delay is 15ms, and 260 for the second test when programmed delay is 20ms. This means, the test fails here. The received data window at the master node showed that the data came at a high rate and ended up being jumbled together. Whereas, the result of the experimental test 3, indicate that the entire 400 packet data send were received under a programmed delay of 25ms. This means all packets were transmitted correctly and transmission was successful. This can be concluded that, the higher the delay the more reliable data packet will be received. Even though, the distances of the slaves and master nodes were increased up to 100m, and the programmed delay remains at 25m, the entire packet data sent were received, this has been seen in from experimental test results of test 4 to test 8. This also means that the minimum transmission delay at this distance should be 25 milliseconds, the next phase of tests is to slowly increase this distance until it is at the max transmission distance of the transmitted and observing the results and the minimum delays achievable.

The 25 ms delay works the same at 100 meters, all expected packets were received. While maintaining the same 25ms delay at about 970 meters' failures are detected in transmission. Since the advertised distance for the transmitter was 1 km we jumped to that distance after a few tests that showed 25 milliseconds to be the necessary delay. But going to 1km only revealed that although the transmitter is capable of about 1 km, not all data reaches successfully even when varying the delay. So we moved down from that distance. It is also important to note that at this point when we achieved the ideal 25 millisecond transmission delay, there was no change necessary to achieve successful transmission even though there is a change in the distance. These experimental test yielded successful results at the distance obtained out of the eight experiments that were undertaken. Those key distances identified are 0.5 meter and 660 meters, hence, these are suitable distances for testing the atom tablet as a master node. The results are almost identical to the results obtained with the high performance processor. The third and fourth algorithms (Algorithm 3 and 4) were tested in an experiment to determine how it takes for the responses to reach when either the slave or master node send packet data to each other. Hence, two experimental test were carried out, the experimental results are presented in Table 2.

The results of the test, using the 660 meters' distance obtained from the previous experiment when the programmed delay between transmissions was set 0 milliseconds because the transmissions depend on acknowledgement from receiver were gathered. There was a slight change in the total time taken for all transmissions as compared with experimental test 1, which could be due to the difference in distance and gives a 943 millisecond propagation delay for the total transmission time taken. This gives an average propagation delay difference of approximately 2.4 milliseconds given that 400 transmissions had taken place.

TABLE 2. Transmission session with response.

Parameters Name	Test B1	Test B2
No of successful transmissions	400	400
No of expected transmissions	400	400
Total time Taken	98.371	99.314
Max detected delay	250	252
Min detected delay	245	245
Average Total Transmission delay	246	248
Programmed delay	0	0
Average Propagation + Processing delay	246	248
Average data packet size	58	58
Total data transmitted	23.2	23.2
Throughput	0.236	0.234

TABLE 3. The results of experimental transmission session.

Parameters	Test C1	Test C2	Test C3
No of successful transmissions	60 data packets	60 data packets	60 data packets
No of expected transmissions	60 data packets	60 data packets	60 data packets
Total time Taken	14.756 secs	14.764 secs	14.744 secs
Max detected delay	251 millisecs	251 millisecs	251 millisecs
Min detected delay	249 millisecs	249 millisecs	249 millisecs
Average Total Transmission delay	250.0 millisecs	250.0 millisecs	249.0 millisecs
Programmed delay	0 millisecs	0 millisecs	0 millisecs
Average Propagation + Processing delay	250.0 millisecs	250.0 millisecs	249.0 millisecs
Average data packet size	58.0 bytes	58.0 bytes	58.0 bytes
Total data transmitted	3.48 kilobytes	3.48 kilobytes	3.48 kilobytes
Throughput	0.236 kb/s	0.236 kb/s	0.236 kb/s

The last two algorithms (Algorithm 5 and 6) were put into test in an experiment to determine the threshold of transmission session involving a two-way multiple-slave's single master node networked environment. Hence, this could let to an establishment of a point at which when it reached congestion might occur. Hence, the result of the experiment is presented in table 3.

The first experimental test (Test C1) was carried out with 3 nodes clustered at approximately 2m distance from the master node and with no programmed delay since the transmissions depend on acknowledgement 120 transmissions and a total network data of 6.96 kb has been obtained. This session only retransmits after verifying, which means it will not be overwhelmed with a continuous string before processing successfully. The second experiment is similar with the first, only that it was carried out at a distance of 20m with same notes with Test C2. The results indicate that the average data packet size received is 58.0 bytes and the total data transmitted is 3.48 kilobytes. Experimental results of test C3 which was carried out with three nodes placed at distances of 5m, 50m and 300m respectively, from the master nodes is presented. There was no programmed delay as each of the receiving nodes, only retransmits after verifying which means it will not be overwhelmed with a continuous string before processing successfully.

From the results received, especially with the continuous transmission tests conducted in "Transmission test A" the most ideal delay between transmissions is 25 milliseconds in order to ensure that the receiving end is not overwhelmed by the transmitted data. This type of transmission is best conducted when a large amount of data needs to be pushed and has to be broken into multiple parts and transmitted as quickly as possible. This delay was found to not matter whether a high performance device is used or not and a low performance Intel Atom powered device was found to be sufficient and as such anything with equivalent or higher performance should be good enough to utilize the 25 millisecond delay determined here.

When transmitting with an acknowledgement as in "Transmission test B" the propagation and processing delays result in a higher average delay in between transmissions with a slight 2.4millisecond delay increase observed between transmitting at 0.5 meters and at 660 meters. The average total transmission delay in between transmission was found to be 246 milliseconds. This is useful because it allows us to know the minimum delay to wait for an acknowledgement from a node before declaring a timeout.

It is important to note that this technique and the values captured here are for the transmitters used in these tests and other transmitters or mediums may have different results so the variables captured here should be made adjustable in the final proposed communication model. The two key variables are the 246 millisecond delay for the acknowledgement based transmissions and the 25 milliseconds for the continuous one sided transmission.

VI. DISCUSSION

Distances among the devices (master and slaves) constitute an important constraint to transmission status and delay. Thus, it was found to that data packets captured in sequences of inter-arrival times were based on distance. Each packet transferred within the AGVs and the controlling program and among the AGVs were traced and captured. Effective traffic performance was understood based on various distances among the transmitters while stationary when a sent instruction was invoked. This sending determined the fastest achievable packet transmission rate and hence the lowest delay in between packet transmissions. The test clearly indicated that the distance between the wireless communication modules, baud rate of transmission, and the frequency of the transmitters had a significant effect on the transmission session.

Although it was observed the even at the farthest possible distance from each AGVs node proper clear transmission of data occurred, yet that only when there was low or no EMF interference. This test was repeated until the best baud rate which facilitated clear loss-free transmission was achieved, a packet containing data that was transmitted at different times with a known incremental variation. Integrity check was performed for the content at each iteration after each set of transmissions ensured that all data instances were received. The baud rate was then modified and the test was repeated until the best possible rate was recorded. The proposed baud rates tested were the common standard rates (this was also limited to the transmitter support and the APC modules support, a max of 19200 bps): and 1200 bps, 2400 bps, 4800 bps, 19200 bps were the testbed. In this fashion the optimal baud rate was determined, and the test then tried at different frequencies to see if different results could be observed. The frequency increments of 50 hz, from the lowest to the highest were supported by the transmitter, and the superior frequencies or a range of frequencies were identified.

The lowest achievable inter-packet transmission delay was also determined. The behavior of these wireless communications with different parameters and the delays necessary in between transmissions were captured. The AGV navigation techniques were required to be flexible which would enhance efficiency in the manufacturing operations of a factory or warehouses. Coordinated networked AGVs were designed to allow for the frequent changes and adjustments required in the efficient communication. This was achieved by incorporating a high degree of onboard tasks (Martínez-Barberá and Herrero-Pérez, 2010). As previously stated that the data packet used for this study are strings analysis, while on transmission session, looking from the master node, they are in the following form:

***1101**, **2|3|2**, **N3**, **commandstring**, **R#**

Where the commas that separate each portion of the data packet string represent the boundaries of each packet. Star is use to indicate the beginning of the data packets and hash sign is used to indicate the ending point of the packets. These are the characters that determine the beginning and the end of the data packet.

*1101,2|3|2,N3,commandstring,R#

The numbers in black color, represent the command count, which determines the number of individual commands in the command portion of the string. This is useful when the master node is transmitting more than one command in one data packet to a master node. Each command is normally separated by the delimiter. To reduce congestion, if there is only 1 command being transmitted this number is omitted, for example if an intended slave node or AGV within the network required to send a data packet into transmission session, let say N3

∗1101, 2|3, N3, commandstring, R#

The node ID N3 is the ID of the intended slave node or AGV within the network that this data packet should go to:

*1101,2|3|2,**N3**,commandstring,R#

Thus, the count indicator will be:

*1101,**2|3**|2,N3,commandstring,R#

These numbers are useful when more than one packet of a group of packets is sent to the same slave node. The first number determines which packet of the group, in this is the second number determines how many packets are in the group. Each packet is sent separately (separated by the 25 millisecond delay)

If more than 3 packets are there, the numbers would be:

First packet: 1|3

Second packet: 2|3

Third packet: 3|3

It is important to note that in most situations a single packet is needed and so in order to reduce data packet length and thus reducing network congestion, when one packet is sent this portion of the data packet does not need to be there so instead of a packet being sent in this way:

∗1101, 1|1|2, N3, commandstring, R#

It can be sent in this way:

∗1101, 2, N3, commandstring, R#

Where the "1|1" are omitted.

Multiple packets are necessary where many commands need to be sent and the data packet size exceeds a pre-set size. The general pre-set size that is chosen for the research is 40 bytes which is 40 characters. If the data packet exceeds that size, it is split into two or more and transmitted, with each not exceeding the 40 byte size. When multiple commands are transmitted, the concept of the sliding window protocol is utilized where if acknowledgement is needed, it is received after 5 packets are transmitted in order to reduce congestion.

The group ID:

*__1101__,2|3|2,N3,commandstring,R#

This is a unique number that identifies all nodes communicating within one group. This is useful when you have separate groups of devices in one area that have to be communicating only among their groups.

Command String:

*1101,2|3|2,N3,**commandstring**,R#

The command string can be a single command or multiple commands. An example of a single command sent to AGV 3 to send agv to station 3.

***1101, N3, Ds3, R#**

Where "Ds3" is the command for the agv to move to new destination, station 3. Another example of three commands. Sending AGV 3 to a new destination, station 3 and to switch on AGV lights and to enable AGV Melody alert:

***1101, 3, N3, Ds3, L1, M1, R#**

Where M1 means switch on melody and L1 means switch on lights. The retransmit status are:

*1101,2|3|2,N3,commandstring,**R**#

This is a character in the packet that determines whether this command needs an acknowledgement from the slave node or not. If a command is sent with the R in it, it indicates that it is an important command and the master node will wait for an acknowledgement to confirm that the AGV has received the command. If no response from the AGV comes to indicate that it has received the command within 100 milliseconds, the master node will retransmit the same command packet again up to 5 times before declaring a timeout if the AGV does not acknowledge and an error flag is raised.

For data packets that do not require acknowledgement, or if it is not important whether the slave nodes receive the command or information or not, the character "R" can be removed from the transmission packet as follows: ***1101,1|1|2,N3,commandstring,#,** hence from the slave node, it will be ∗**1101**,**2|3|2,M1**,**information**,**R#**:

VII. CONCLUSION

The AGV prototype has been constructed to implement a framework of coordinated communication environment aimed at providing the most efficient mode of transferring many types of materials in and out of a factory or a manufacturing setting. This ensures a scalable and flexible coordination of movement of materials from one point to another. This kind of system where AGVs are required to move products is in high demand in the manufacturing industry as it assists in driving products easier and faster. This research built an AGV model with an onboard wireless device (transmitter) and created a network communication environment for individual AGVs that communicate with each other and equipped with controlling program using a wireless transmitter. The controlling program coordinates the dataflow among the AGVs and ensures that information is sent, arrives and is received within precise time-bounds. This paper proposes the design of a coordinated AGV architectural framework and has developed a prototype controlled by a central monitoring system that manages, monitors and controls the AGVs through wireless communication. The three AGVs prototype consists of a APC220 wireless transmitter and a window tablet PC containing the controlling program with the APC220 wireless transmitter and has been tested in an experimental setup. It has been established that a coordinated transmission scheme can be used while the AGVs moves as well as when in stationary mode. The data packets have been captured in sequences of inter-arrival times. Each packet's transfers within AGVs and the controlling program and among the AGVs have been traced and captured. Thus, the effective traffic performance based on various distances among the transmitter while stationary and on the move has been established.

The aim of this research is to develop a software implementable communication model capable for efficient delivery of packet data to and fro. The research methodological approach involves formulating algorithms, developing a prototype, and experimental validation. The algorithmic design dwells on the proposed communication model. Autonomous Guided Vehicle models are developed as the prototype for implementing and evaluating the algorithm proposed. The key issue with the prototype development lies with the choice of a transmitter, intended for use in the industrial and product development context. The experimental validation exerts quantitative data and constraints of the transmission medium. This has been obtained through field experiments measuring transmission performance (rate/speed) from different scenarios involving a master and slaves' nodes, with given technical constraints. Sufficient amount of tests was performed in order to receive a reasonable quantitative amount of feedback.

REFERENCES

[1] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," *Microelectron. J.*, vol. 40, no. 9, pp. 1322–1336, Sep. 2009.

[2] M. Eisenstein, "Living factories of the future," *Nature*, vol. 531, no. 7594, pp. 401–403, 2016.

[3] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proc. 9th IEEE Int. Conf. Ind. Inform. (INDIN)*, Jul. 2011, pp. 410–415.

[4] T. Chapman, "Lab automation and robotics: Automation on the move," *Nature*, vol. 421, no. 6923, pp. 661–666, 2003.

[5] S. G. M. Hossain, M. Y. Ali, H. Jamil, and M. Z. Haq, "Automated guided vehicles for industrial logistics—Development of intelligent prototypes using appropriate technology," in *Proc. 2nd Int. Conf. Comput. Autom. Eng. (ICCAE)*, vol. 5. Feb. 2010, pp. 237–241.

[6] D. Herrero-Perez and H. Martinez-Barbera, "Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 2, pp. 166–180, May 2010.

[7] L. Schulze and A. Wullner, "The approach of automated guided vehicle systems," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Format. (SOLI)*, Jun. 2006, pp. 522–527.

[8] N. Wu and M. Zhou, "Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.

[9] H. M. Barberá, J. P. C. Quinonero, M. A. Z. Izquierdo, and A. G. Skarmeta, "i-Fork: A flexible AGV system using topological and grid maps," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2. Sep. 2003, pp. 2147–2152.

[10] G. Kaloutsakis, N. Tsourveloudis, and P. Spanoudakis, "Design and development of an automated guided vehicle," in *Proc. IEEE Int. Conf. Ind. Technol.*, vol. 2. Dec. 2003, pp. 990–993.

[11] H. Martínez-Barberá and D. Herrero-Pérez, "Autonomous navigation of an automated guided vehicle in industrial environments," *Robot. Comput.-Integr. Manuf.*, vol. 26, no. 4, pp. 296–311, Aug. 2010.

[12] H. Fazlollahtabar and M. Saidi-Mehrabad, "Risk assessment for multiple automated guided vehicle manufacturing network," *Robot. Auto. Syst.*, vol. 74, pp. 175–183, Dec. 2015.

[13] R. Chiba, J. Ota, and T. Arai, "Integrated design for routing and network in AGV systems using co-evolution," in *Proc. IEEE Int. Conf. Robotics, Intell. Syst. Signal Process.*, vol. 1. Oct. 2003, pp. 318–323.

[14] M. G. Kay, "Material handling equipment?" Dept. Ind. Syst. Eng., North Carolina State Univ., Raleigh, NC, USA, Tech. Rep. 65, 2012.

[15] W. L. Maxwell and J. A. Muckstadt, "Design of automatic guided vehicle systems," *AIIE Trans.*, vol. 14, no. 2, pp. 114–124, 1982.

[16] R. J. Gaskins and J. M. Tanchoco, "Flow path design for automated guided vehicle systems," *Int. J. Prod. Res.*, vol. 25, no. 5, pp. 667–676, 1987.

[17] W. G. Goetz, Jr., and P. J. Egbelu, "Guide path design and location of load pick-up/drop-off points for an automated guided vehicle system," *Int. J. Prod. Res.*, vol. 28, no. 5, pp. 927–941, 1990.

[18] N. N. Krishnamurthy, R. Batta, and H. Karwan, "Developing conflict-free routes for automated guided vehicles," *Oper. Res.*, vol. 41, no. 6, pp. 1077–1090, 1993.

[19] C. W. Kim and J. M. A. Tanchoco, "Conflict-free shortest-time bidirectional AGV routeing," *Int. J. Prod. Res.*, vol. 29, no. 12, pp. 2377–2391, 1991.

[20] C. W. Kim and J. M. A. Tanchocoj, "Operational control of a bidirectional automated guided vehicle system," *Int. J. Prod. Res.*, vol. 31, no. 9, pp. 2123–2138, 1993.

[21] K. H. Kim, S. M. Jeon, and K. R. Ryu, "Deadlock prevention for automated guided vehicles in automated container terminals," *OR Spectr.*, vol. 28, no. 4, pp. 659–679, 2006.

[22] C. Wuwei, J. K. Mills, and S. Wenwu, "A new navigation method for an automatic guided vehicle," *J. Robot. Syst.*, vol. 21, no. 3, pp. 129–139, Mar. 2004, doi: 10.1002/rob.20004.

[23] C. Basnet and J. H. Mize, "Scheduling and control of flexible manufacturing systems: A critical review," *Int. J. Comput. Integr. Manuf.*, vol. 7, no. 6, pp. 340–355, 1994.

[24] R. Rachamadugu and K. E. Stecke, "Classification and review of FMS scheduling procedures," *Prod. Planning Control*, vol. 5, no. 711, pp. 2–20, 1994.

[25] M.-H. Han and L. F. McGinnis, "Control of material handling transporter in automated manufacturing," *IIE Trans.*, vol. 21, no. 2, pp. 184–190, 1989.

[26] T. J. Schriber and K. E. Stecke, "Machine utilizations achieved using balanced FMS production ratios in a simulated setting," *Prod. Ratios Simul. Setting, Ann. Oper. Res.*, vol. 15, no. 1, pp. 229–267, 1988.

[27] I. Sabuncuoglu and D. L. Hommertzheim, "Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system," *Int. J. Prod. Res.*, vol. 30, no. 5, pp. 1059–1080, 1992.

[28] I. Sabuncuoglu and D. L. Hommertzheim, "Experimental investigation of an FMS due-date scheduling problem: Evaluation of machine and AGV scheduling rules," *Int. J. Flexible Manuf. Syst.*, vol. 5, no. 4, pp. 301–324, 1993.

[29] W. E. Wilhelm and H.-M. Shin, "Effectiveness of alternate operations in a flexible manufacturing system," *Int. J. Prod. Res.*, vol. 23, no. 1, pp. 65–79, 1985.

[30] I. J. Chen and C.-H. Chung, "Effects of loading and routeing decisions on performance of flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 29, no. 11, pp. 2209–2225, 1991.

[31] B. H. Khoshnevis and Q. M. Chen, "Integration of process planning and scheduling functions," *J. Intell. Manuf.*, vol. 2, no. 3, pp. 165–175, Jun. 1991.

ADAMU I. ABUBAKAR received the B.S. and M.S. degrees in geography and computer science from the School of Science, Bayero University Kano, Nigeria, in 2004 and 2006, respectively, and the Ph.D. degree in information technology from the Department of Information System, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia. Since 2012, he has been with the Department of Computer Science, International Islamic University Malaysia, where he is currently as Assistant Professor. His current research interests include machine learning and parallel computing.

ELBARA ELDAW ELNOUR MOHAMED is currently pursuing the M.Sc. in computer science with the Department of Computer Science, International Islamic University Malaysia. His research interest lies in the area of network communication protocol.

AKRAM M. ZEKI received the Ph.D. degree in digital watermarking from the University of Technology at Malaysia, Malaysia. He is currently with the Department of Information System, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, an Associate Professor. His current research interests include digital watermarking.