Proceeding of the IEEE
International Conference on Information and Automation
Yinchuan, China, August 2013

# An Experimental Evaluation of the RT-WMP Routing Protocol in an Indoor Environment

Matthew Tan[1,3], Ming Liu[1,2], and Roland Siegwart [1]

[1]Autonomous Systems Laboratory, ETH Zurich, Switzerland
[2]The Hong Kong University of Science & Technology, Hong Kong
[3]Imperial College London, United Kingdom

***Abstract*—The RT-WMP routing protocol allows a user to determine which data streams ought to have a higher priority for transmission over a wireless mobile ad-hoc network. This paper provides a benchmark of the RT-WMP protocol in a typical indoor environment. The benchmarking results will reveal that the RT-WMP protocol is able to handle multiple data streams well over moderate distances of up to 10m. Above this distance, with a scarcer bandwidth, the protocol is still able to allow continued transmission of at least one data stream, which may not necessarily be the data stream with the highest priority. The results will lay the base for the development of a resource allocation strategy that allows for continued data transmission of all data streams, and not just the data stream with the highest priority only.**

## I. Introduction

WITH the advent of wireless communication methods, there has been a huge boom in this field of research. More recently, the number of applications for mobile ad-hoc networks (MANETs) have been increasing, ranging from surveillance by law-enforcement agencies, the transmission of files in a classroom via a Personal Area Network (PAN) to disaster relief operations[1]. In surveillance operations, MANETs can work in tandem with sensor networks to obtain data of a given area that is hazardous for humans to enter (for example, measuring radiation near a nuclear reactor that has suffered a meltdown). In disaster relief operations, [2] reveals that MANETs can provide last-mile communications among first responders of disaster relief organisations at a low cost. Due to the MANETs ability to provide multi-hop communications, the range of the network can be extended. The team's research also reveals that disaster relief organisations (such as the Red Cross, Oxfam etc) look forward to having MANETs that are able to send images and, more importantly, have voice communications over the MANET. Other schemes used in Urban Search and Rescue Operations, such as the schemes described in [3] and [4], as well as other applications such as colocalisation [5], [6] and topological navigation [7], require reliable network connections, especially timely message delivery, to function correctly and efficiently.

Although there is a potentially huge market for this form of multi-hop technology, it is still not all-pervasive yet. [8] reveals that although the IEEE 802.11 protocol has the mechanisms to implement ad-hoc networks, they are not suited for multi-hop wireless ad-hoc networks, similar to those found in MANETs, due to high packet collisions and exponential back-offs. Routing protocols have since been developed to work in tandem with the IEEE 802.11 protocol and overcome the problems faced by the latter protocol. The routing protocol to be used for this paper is the Real Time- Wireless Multi-hop Protocol (RT-WMP), which is able to fulfil Hard Real-Time (HRT) traffic requirements[9]. This is further described in Section II.

e-mail: mtan@ethz.ch, ming.liu@mavt.ethz.ch, rsiegwart@ethz.ch.

In addition, there has been much research into the performance of wireless multi-hop networks and their respective protocols. However, many of them have performed their tests and evaluations on simulators (for example [10], [11]), and not under real-life conditions. These simulators have pre-set "environmental" conditions, which do not take into account other forms of influences that are commonly found in a realistic environment. In other words, the system under test is not fully exposed to a real-life environment, and hence its behaviour in reality may deviate from that shown in simulations. This paper presents a benchmarked performance of the RT-WMP protocol under a typical, real-life office environment. It is organised as follows: Section II gives an overview of the RT-WMP protocol to be used in this evaluation; Section III describes how the RT-WMP protocol is implemented for this evaluation; Section IV describes the motivation and the experimental setup of the benchmarking process; Section V presents the results of the evaluation; Section VI concludes this paper with conclusions drawn and an insight into future work to be done.

## II. Real Time - Wireless Multi-hop Protocol [9]

The RT-WMP routing protocol is a novel token–based protocol that works over the existing IEEE 802.11 protocol and provides hard real-time traffic support. End-to-end message delay has a bounded and known duration and is able to manage global static message priorities. Multi-hop communications is also possible for a relatively small group of mobile nodes (approximately 10 – 20 units maximum), thereby allowing the network coverage to be extended. Since it runs on the existing IEEE 802.11 protocol, it can run on existing hardware too. More importantly, the protocol has the ability to prioritise data traffic in the network, using 128 priority levels. Messages with

978-1-4799-1334-3/13/$31.00 ©2013 IEEE

the same priority are stored in FIFO order. In this way, the network is able to fulfill hard real-time requirements.

The protocol has three stages, in the following order: *Priority Arbitration Phase, Authorisation Transmission Phase, Message Transmission Phase*. Each of these phases is described in more detail below:

### *A. Priority Arbitration Phase (PAP)*

To determine which node holds the message with the highest priority, a token containing information on the priority level of the Most Priority Message (MPM), as well as the identity of the node that has the MPM, is used. At the start, the initiating node stores the message with the highest priority in its queue as the MPM, as well as its identity, into the token and passes it to the next node. Upon receipt of the token, each node checks the priority of the messages in its own queue. If the current node ascertains that one of its messages has a higher priority than the MPM stated in the token, it overwrites the information in the token and passes it on to the next node. Otherwise, the node does nothing to the token and simply passes it on. This process repeats for all nodes in the network. Once the token reaches the last node, the PAP ends and the next phase starts.

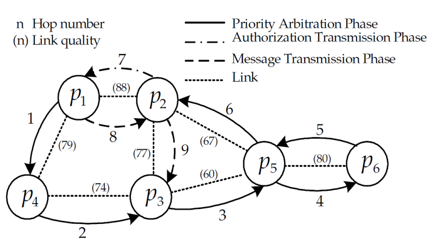
### *B. Authorisation Transmission Phase (ATP)*

With the conclusion of the PAP, the final node sends an authorisation by calculating a path (via the other nodes in the network) to the holder of the MPM. Once the authorisation arrives at the node with the MPM, the next phase begins.

### *C. Message Transmission Phase (MTP)*

Once the authorisation to transmit is received, the node then calculates a path to the destination and then sends the message accordingly. Other vital components of the protocol are further described below:

### *D. Link Quality Matrix (LQM):*




Link Quality Matrix

| | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ |
|---|---|---|---|---|---|---|
| $p_1$ | - | 88 | 0 | 79 | 0 | 0 |
| $p_2$ | 88 | - | 77 | 0 | 67 | 0 |
| $p_3$ | 0 | 77 | - | 74 | 60 | 0 |
| $p_4$ | 79 | 0 | 74 | - | 0 | 0 |
| $p_5$ | 0 | 67 | 60 | 0 | - | 80 |
| $p_6$ | 0 | 0 | 0 | 0 | 80 | - |

Figure 1. A hypothetical situation described by the network graph and the corresponding LQM. The hops sequence of the protocol is also shown.[9]

The topology of the network is described by the LQM. Each column of the $LQM_k$ describes the link quality of the $p_k$ node with its neighbours. The ATP and MTP (described in Fig 1) use the LQM to calculate the best path to send a message from source to destination. Every node is responsible for updating its column of the LQM (both the local copy and shared copy) to update all other nodes about topology changes. This is done by modifying and sending the shared copy of the LQM with the token during the PAP.

### *E. Error-Handling*

The RT-WMP protocol is able to handle node failures and frame duplications.

In the former case, when a node B (say) receives a frame from node A, node B processes a frame and sends another frame to a third node C (or back to Node A). Node A listens for this frame, which it considers as an acknowledgement when it detects it (implicit acknowledgement). If no frame is detected after a certain period of time, the node is assumed to be lost. Depending on which phase the protocol is in, separate recovery schemes are put into action to ensure transmission continues as soon as possible. If a node fails during the ATP or MTP, the transmission is halted and a PAP is reinitiated. If a node fails during the PAP, the shared LQM carried by the token is updated and the PAP continues, with the token bypassing the failed node. Should a failed node suddenly reappear, mechanisms are in place to allow it to be reinserted back into the network.

For the latter case, each frame (tokens, authorisation, actual messages are considered as frames) has a serial field that is incremented by each node. This value is also saved locally by the node before the frame is transmitted to the next node. When a node receives a frame with a serial field that is lower than or equal to the highest serial that has been transmitted, the node discards this frame. The node also sends the originator a drop frame to disregard this frame.

## III. Implementation of RT-WMP

### *A. Components Used and Respective Specifications*

The following section gives a brief description of the robotic agent and its corresponding components used to implement the protocol:

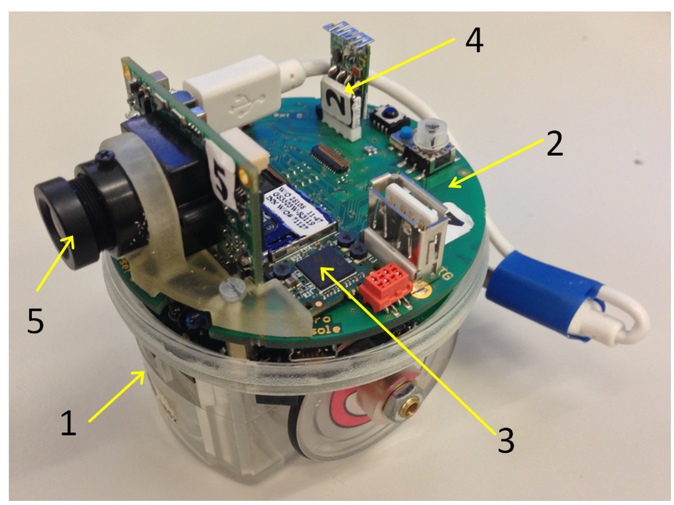



Figure 2. e-puck Robot. (Note that the camera USB is unplugged here to give a better view of the components.

*1) e-puck Robot:* The e-puck robot was first developed by the Swiss Federal Institute of Technology in Lausanne (EPFL) for teaching purposes. It is now sold commercially by GCtronic*[12]*.

The e-puck is controlled by a dsPIC 30 microprocessor running at 30MHz, processing up to 15 million instructions per second (MIPS). It has 8KB Random Access Memory (RAM) and 144KB Flash memory.

To move around, a two wheel centred differential drive configuration is used, with each wheel controlled by a stepper motor. This allows for reliable control of the motor speed without having additional velocity sensors installed. The maximum drive speed of the e-puck in a straight direction is 0.129m/s [13].

*2) Overo Extension Console Board [14]:* The Overo Extension Console connects directly to the UART expansion bus slots of the e-puck, thereby allowing direct communication between an Overo Gumstix computer (described later) mounted on the Extension Console and the dsPIC microprocessor on the e-puck. It also contains a USB Host port and USB On-The-Go (OTG) port to connect external accessories to the console. Finally, a mini-USB serial port is provided to allow access to the Linux system found on the Gumstix through a terminal console via another laptop.

*3) Overo Gumstix Computer [15]:* This Gumstix computer utilises the Texas Instruments OMAP 3530 Applications Processor clocked at 720 MHz. The processor's architecture is the ARM-based Cortex-A8 core. A lean version of Ubuntu 10.04 LTS Lucid using Kernel Version 2.6.34 was installed on the Gumstix computer. Following that, a scaled down version of the open-source Robot Operating System (ROS) (Electric distribution) was also installed. All these were easily accessible via a terminal program from an external laptop connected via the mini-USB serial port of the Overo Console. A huge benefit of doing so is to allow the user to operate the e-pucks independently using similar programming environments and tools.

*4) WiFi Dongle [16]:* The WiFi dongle used is the Edimax EW-7811Un, which uses the Realtek RTL8192cu chipset and is isotropic (the power of the transmitted signal is the same in all directions) . It is 802.11b/g/n compatible , but is only able to run in the b mode, b/g mode or n mode separately. The Linux system on the e-puck uses the 802.11b/g mode only. For this mode, the output transmission power is 15 ± 1.5dBm. Range testing completed with two e-pucks yields a range of approximately 20m before the connection drops.

*5) Point Grey Firefly Camera [17]:* While the e-puck has an inbuilt pinhole camera, it produces a lot of noise and has a huge motion blur, due to its small sensor area.

The Point Grey Firefly camera (Part Number: FMVU-03MTM-CS) was chosen as it is widely used in the robotics community, has drivers available in the Linux kernel and is also compatible with the stable libdc1394 library. The camera itself produces 0.3 MegaPixel monochrome images at a specified rate of 60 frames per second (fps) for a 752x480 image, using a 1/3-Inch CMOS sensor. Images produced are accessible by running a node on ROS and subscribing to the advertised topic[13].

### B. Performance Impacts

*1) Laptop:* To display the image transmitted from the e-puck, a custom Graphical User Interface (GUI) was created. The user is able to choose which image topic to subscribe to via ROS and display the output on the display area. As the GUI was created using QtCreator, the image transmitted via ROS had to be converted to a QImage *[18]* object, before it could be displayed on the screen. Thus, to test if this conversion process would affect performance, a well-used ROS node that accesses the laptop's built-in camera was used to send images via ROS to the GUI.

On the GUI, a calculation of the number of Frames per Second (FPS) was executed to determine the instantaneous frequency at which images were being displayed on the screen. The FPS formula used is a simple moving average of the five most recent images. In other words:

$$FPS = \left[\frac{dur1 + dur2 + dur3 + dur4 + dur5}{5}\right]^{-1} \quad (1)$$

where *durx* represents the duration taken for the $x$th image to arrive at the laptop after the $(x-1)$th image. Each instantaneous FPS value is saved to a CSV file for the calculation of the average FPS over the particular session whenever a new image is received at the laptop (ie the receiver).

When executing the GUI with the laptop's built-in camera running, the frequency yielded is similar to that yielded by the GUI. There was also no visible lag present in the video stream. Thus, it may be considered that the GUI does not contribute any significant latency.

*2) e-puck:* The camera's connection over USB produces an additional CPU load of about 15% (maximum up to 30%) on the Gumstix when set to acquire images at a rate of 15Hz[13]. However, this does not affect performance significantly. A check using the e-puck's system monitor revealed that with all the required processes running, there was still more than 50% of the CPU available for other processes. Furthermore, a check on ROS further confirms that the images captured from the Firefly camera are constantly retrieved at a frequency of 15Hz, as set.

In short, all performance degradations are purely due to the impedences experienced in the wireless network.

## IV. Benchmarking

In this section, the author explains his motivation for conducting a series of static and mobile benchmarking tests, as well as describe the environment and methods used to carry out the tests.

### A. Motivation

The main motivation for conducting this series of static benchmarking tests is to determine the best possible performance obtainable by the RT-WMP protocol between a network of mobile agents, which may use either an Intel platform or an ARM platform, at various distances. Clearly, the best possible performance is attained by having a single-link network between two nodes. This is the basis of the benchmarking setup.

The results obtained here will be used to gauge the performance of a novel network resource allocation algorithm in a multi-link ad-hoc network. This algorithm is currently under development.

### B. Description of Environment and Setup

The routing protocol was evaluated along an office corridor. (Refer to Figure 3). Along this corridor, there are several WiFi routers running on the 2.412GHz frequency range. Fig 4 shows the amount of traffic on that channel. On the other hand, the next commonly used frequency band at 2.462GHz has much less traffic (Figure 5). Thus, the network was set to run at 2.462GHz.

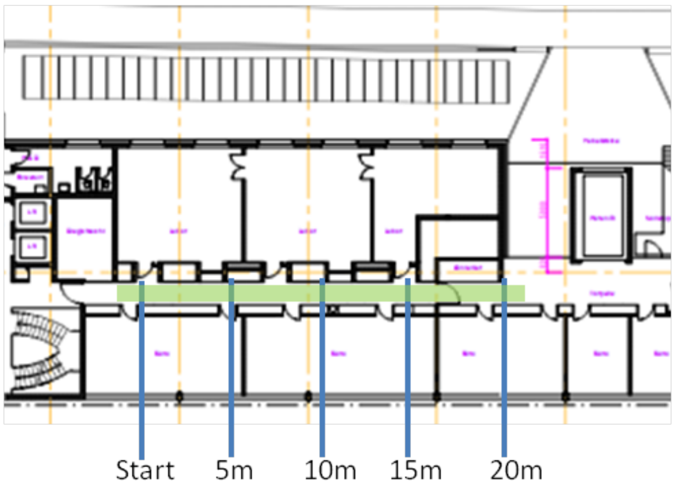



Figure 3. Floorplan of Office Corridor Used. The e-puck was placed at each interval.


Figure 4. WiFi Sweep at 2.412GHz

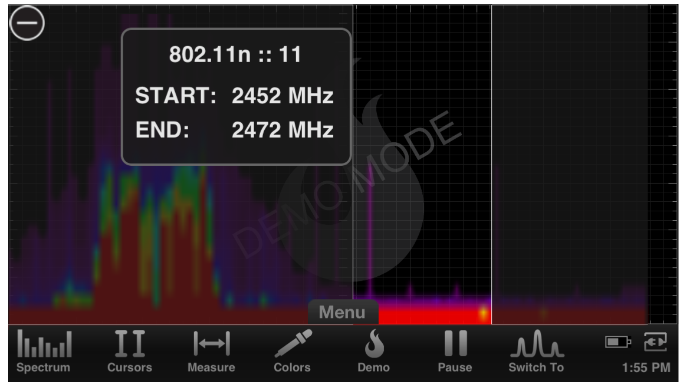



Figure 5. WiFi Sweep at 2.462GHz

The network consisted of two agents: an e-puck and a standard Linux laptop running Ubuntu 12.04 Precise Pangolin. The laptop utilises a built-in Ultimate N WiFi Link 5300 network card from Intel Corp, with a transmission power of 15dBm, the maximum possible value. The Linux 3.2.40-generic-pae kernel was used to compile and run the ROS-RT-WMP node (ref: `www.ros.org/wiki/ros_rt_wmp`) in the user space. In addition, the custom-made GUI was used to view images received by the laptop via ROS.

Various network monitoring programs were used to record measurements of the ad-hoc network, in addition to the FPS measurement tool on the custom GUI. The programs used were the wmpSniffer (described in [9]) and the free Linux console program BWM-NG.

For the e-puck, the same ROS-RT-WMP node was compiled on the ARM platform. Images transmitted by the e-puck were set to have as low a quality as possible, without losing too much resolution such that the image can still be interpreted by the user. The image stream took up an average bandwidth of around 160KB/s. In addition, the motors of the e-puck were activated to receive movement commands from the laptop.

In the ROS environment, there were a total of three topics transmitted over the network, ranked from highest priority to lowest priority:

- a "watchdog" topic that routinely transmits a "hello" message to the laptop at a frequency of 5Hz to ensure network connectivity between the laptop and the e-puck. If the laptop does not receive a "hello" message after 20s, a warning appears on the GUI to alert the user and the motor command topic (described next) is automatically cut-off. The bandwidth taken up is about 165B/s.
- a motor command topic that transmits movement commands from the laptop to the e-puck at a frequency of 20Hz when activated, taking up an average bandwidth of 960B/s.
- a compressed image topic that handles the image transmission from the e-puck to the laptop.

The priority of these topics was determined in such a way that all the nodes were able to transmit with the e-puck placed next to the laptop (ie the distance between the e-puck and laptop was approximately 0m) and all of the ROS topics could be transmitted through the network. Altering the above priority settings would cause one of the topics to be blocked out by the other topics.

### C. Procedure

In this subsection, the procedures of the static tests are mentioned. The main measurement used to determine performance shall be the average of the instantaneous FPS (as shown in equation (1)). Mathematically,

$$Avg\,FPS = \frac{\sum_1^n (\text{instantaneous frequencies})}{n} \qquad (2)$$

where $n$ is the number of images received over the transmission period.

The laptop used acted as a "base station" that would receive images from the e-puck's camera. Hence, it remained at the point labelled "Start" in Fig 3. On the other hand, the e-puck acted as a mobile agent that transmitted images back to the "base station". Measurements were taken with the e-puck placed at 5m intervals along the corridor, up to 20m away from the laptop. At each interval, measurements were taken for a duration of thirty seconds with the e-puck continuously transmitting images over the network, in addition to the other transmitted ROS topics mentioned in Section IV-B.

## V. Experiments

In this section, we briefly discuss the behaviour of wireless channels and present results.

### A. Discussion

In general, a wireless channel is exposed to various impediments, such as path loss and interference, that dissipates the transmission signal as it travels through the air medium. This, in turn, limits the range and data rate of the wireless transmission. In other words, the performance of a wireless communication channel is dependent directly on the power of the transmission signal received at the receiver.

For a direct line-of-sight setup, the free space propagation model is a sufficient model that calculates the amount of power received by the receiver at a distance $d$ away from the transmitter. In this model, the relationship between the received power $P_r$ and the transmitted power $P_t$ is given by:

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\Pi d}\right)^2 \qquad (3)$$

where $\lambda$ is the wavelength of the signal, $G_t$and $G_r$are the transmitter and receiver antenna gains, respectively[19]. From the above equation, one can expect the performance of the image transmission to vary inversely with the squared of the distance. In mathematical terms, the average FPS is expected to vary with the distance $d$ according to:

$$Avg\,FPS \propto \frac{1}{d^2} \qquad (4)$$

Note that the above equations do not take into account interference from other sources, particularly transmissions at neighbouring frequencies. With the interferences present in reality, the performance is expected to be worse than that calculated by equation (3). However, the trend formed is still expected to follow the relationship described by equation (4), since the frequency range used in the setup has significantly low interference, as depicted in Figure (5).

### B. Results

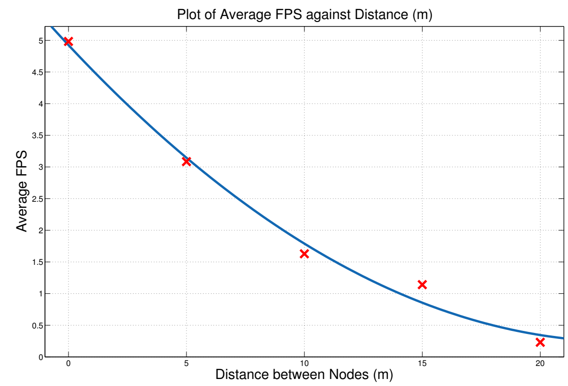



Figure 6. Plot of Average FPS against Distance (m)

Figure 6 shows the average FPS for a thirty second transmission period held at each interval. An inverse relationship can be observed between the distance and average FPS, as predicted by equation (4). Figure 7 shows the boxplots of the instantaneous frequencies at each distance interval. Observe that the median values also follow an inverse relationship.

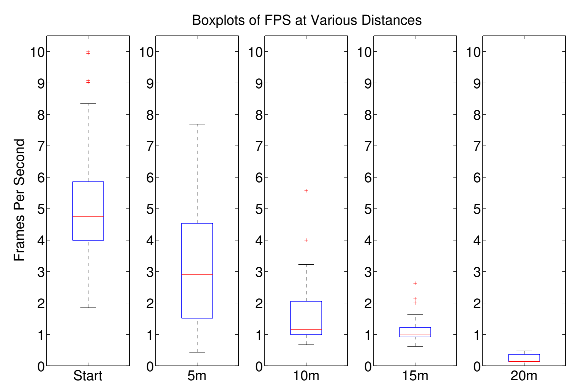



Figure 7. Boxplot of Instantaneous FPS at Various Intervals

It is worth noting the large range of values of instantaneous frequencies observed from the boxplots corresponding to the "Start" and "5m" intervals. This is due to the RT-WMP protocol delaying the transmission of the image topic, since it has the lowest priority. Therefore, there exists certain time intervals where no image is received by the receiver. At other intervals, the images are streamed in at a high rate, thereby causing the instantaneous FPS to increase. This could be due to the other two topics momentarily not transmitting any data, thus the protocol allowed the image topic to use the bandwidth.

As the distance between the two nodes increases, the median and average FPS show a general decrease, together with a decrease in the range of instantaneous FPS values observed. In general, the performance of the image transmission degrades significantly at higher distances. From Table I, a plausible reason is very likely due to the increasing proportion of messages that are lost in transmission, possibly due to path losses, interference or fading. This accounts for the substandard performance observed when the nodes are 20m apart, with the image topic experiencing an average delay of 94s.

Table I
MEASUREMENTS OF DATA TRAFFIC TAKEN USING WMPSNIFFER

| Dist (m) | Avg Delay (ms) | Avg Bandwidth (Kbps) | % of Messages Lost |
|---|---|---|---|
| 0 | 8882.86 | 409.25 | 31.208 |
| 5 | 26893.35 | 165.35 | 44.759 |
| 10 | 38982.25 | 113.14 | 49.412 |
| 15 | 70011.95 | 66.38 | 64.409 |
| 20 | 94366.02 | 46.99 | 77.087 |

As the transmission was ongoing with the e-puck at a distance of 15m and 20m away from the laptop, it was observed that the transmission of the "hello" messages from the "watchdog" topic became intermittent, just like the image topic. This would result in the 20s threshold being exceeded, cause an alert to be displayed on the laptop and cut off the motor command topic automatically. However, once the motor command topic was stopped, some images then began to start streaming in and the "hello" messages were once again received every 8 to 15s. Turning the motor command on again would cause the transmission of the "watchdog" and images to slow down instantly, mostly to a halt, until the 20s threshold was elapsed again. Clearly, this was due to the lack of bandwidth available to accommodate all the data streams present when the e-puck is placed a long distance away from the laptop. With all the three topics running, the motor command topic was able to grab onto most of the bandwidth, even though it does not have the highest priority, because it runs at a higher frequency than the "watchdog" topic, and it has a higher priority than the image transmission topic. Consequently, all the other topics were unable to transmit data.

## VI. CONCLUSION AND FURTHER WORK

In conclusion, this paper provides a benchmark for the use of the RT-WMP protocol implemented on a simple mobile ad-hoc network in a typical office environment, scaled across different device architectures (ARM and Intel). When nodes are placed near to each other, it is clear that the RT-WMP protocol is capable of prioritising data traffic. When the bandwidth is reduced, the RT-WMP protocol minimally still allows at least one topic to continue transmitting its data, unlike other routing protocols which would simply cause the network to get jammed under similar conditions. Naturally, even under harsh conditions, having only one topic transmitting over a long period of time while neglecting the other data streams is not a favourable situation either.

The next step forward, then, is to allow the RT-WMP protocol to dynamically change the priorities of the data streams so that each stream gets a chance to transmit over a narrow bandwidth. A preliminary concept is to use game-theory or auction-based methods to determine which data stream should be given the higher priority for using the scarce bandwidth for transmission. This shall be the scope of the author's future work towards developing a novel resource allocation strategy in a multi-link mobile ad-hoc network, and the results published in this paper will be used as a gauge to evaluate the performance of the author's contribution.

## REFERENCES


[1] P. Goyal, V. Parmar, and R. Rishi, "Manet: Vulnerabilities, challenges, attacks, application," *International Journal of Computer Engineering & Management*, vol. 11, pp. 32–37, January 2011.

[2] K. L. B. Ng, C. K. K. Lim, Z. Ye, J. Q. Yeow, L. Huang, and M. F.-H. Tan, "Commercial report (silverking team)." Imperial College London, Department of Electrical and Electronic Engineering, Third Year Design and Build Project., June 2012.

[3] M. Liu, F. Colas, F. Pomerleau, and R. Siegwart, "A Markov semi-supervised clustering approach and its application in topological map extraction," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.(IROS 2012)*, 2012.

[4] G. Kruijff, M. Janıcek, S. Keshavdas, B. Larochelle, H. Zender, N. Smets, T. Mioch, M. Neerincx *et al.*, "Experience in system design for human-robot teaming in urban search & rescue," in *Proc. of 8th Int. Conf. on Field and Service Robotics (FSR 2012), Matsushima, Japan, Jul*, 2012.

[5] L. Wang, M. Liu, M. Q.-H. Meng, and R. Siegwart, "Towards Real-Time Multi-Sensor Information Retrieval in Cloud Robotic System," in *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2012.

[6] L. Wang, M. Liu, and M. Q.-H. Meng, "Towards Cloud Robotic System: A Case Study of Online Co-localization for Fair Resource Competence," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2012.

[7] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, "The role of homing in visual topological navigation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.(IROS 2012)*, 2012.

[8] S. Xu and T. Saadawi, "Does the ieee 802.11 mac protocol work well in multihop wireless ad hoc networks?" *Communications Magazine, IEEE*, vol. 39, no. 6, pp. 130–137, 2001.

[9] D. Tardioli and J. Villarroel, "Real time communications over 802.11: Rt-wmp," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, 2007, pp. 1–11.

[10] M. Cornils, M. Bahr, and T. Gamer, "Simulative analysis of the hybrid wireless mesh protocol (hwmp)," in *Wireless Conference (EW), 2010 European*, 2010, pp. 536–543.

[11] B. Staehle, M. Bahr, D. Fu, and D. Staehle, "Intra-mesh congestion control for ieee 802.11s wireless mesh networks," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, 2012, pp. 1–7.

[12] GCtronic, "E-puck mini doc (ver 1.0)," August 2005, accessed on 8 June 2013. [Online]. Available: http://www.gctronic.com/files/miniDocWeb.pdf

[13] P. Gohl, "Localisation of a swarm of mini-robots," Master's thesis, ETH Zurich, Spring 2012.

[14] GCtronic, "Overo extension," May 2013, accessed 8 June 2013. [Online]. Available: http://www.gctronic.com/doc/index.php/Overo_Extension

[15] Gumstix, "Overo water com," 2012, accessed 19 June 2013. [Online]. Available: https://www.gumstix.com/store/product_info.php?products_id=228

[16] Edimax, "Edimax ew-77811un wifi dongle datasheet," accessed on 22 June 2013. [Online]. Available: www.edimax.com/images/Image/datasheet/Wireless/EW-7811Un/EW-7811Un_Datasheet_1103.zip

[17] PointGrey, "Point grey firefly camera specifications," 2008, accessed on 22 June 2013. [Online]. Available: http://www.ptgrey.com/products/fireflymv/models/fmvu-03mtm-cs.asp

[18] Q. Developers, "Qimage class reference," accessed 14 May 2013. [Online]. Available: http://harmattan-dev.nokia.com/docs/library/html/qt4/qimage.html

[19] C. S. R. Murthy and B. Manoj, *Ad Hoc Wireless Networks Architectures and Protocols*, 3rd ed., ser. Prentice Hall Communications Engineering and Emerging Technologies Series, T. S. Rappaport, Ed. Prentice Hall, 2004.