International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

http://dx.doi.org/10.12785/ijcds/110104

Virtual Machine Migration in IoT Based Predicted Available Bandwidth and Lifetime of Links

Ali Ouacha[1]

[1] *Computer Science Department, Faculty of Sciences, Mohammed V University, Rabat, Morocco*

Received 26 Nov. 2020, Revised 02 Apr. 2021, Accepted 22 Jul. 2021, Published 9 Jan. 2022

Abstract: The Internet of Things environment is described by a large number of connected physical objects that have their own digital identity and capable of communicating with each other through wireless interfaces and through information and communication technologies. The networked objects (nodes) are generally characterized by limited resources (low capacity of calculation, storage, energy, etc.). In order to overcome this weakness, each of these connected nodes can delegate the execution of a set of tasks to another selected node in the network or to an Edge Server in the periphery of the network via a routing protocol. Thus, to process them, the destination node (with unlimited resources) must firstly have the appropriate environment represented by the virtual machine of the source node. In order to offer the possibility of migrating virtual machines in the network, we aim in the present study to modify the OLSR routing protocol in such a way to introduce the predictive dynamic available bandwidth as a criterion for the construction of routes. In addition, to reduce the mobility effects on the network topology, the predicted lifetime of links to be broken is taken into account. To validate our approach, intensive simulations have been made with NS3 and have shown the effectiveness of our proposal.

Keywords: Internet of Things (IoT), Virtual Machine Migration, MANET, OLSR Routing Protocol, Dynamic Avalable Bandwhidth Prediction, Rest Time To Quitte

---

1. INTRODUCTION

With new wireless communication technologies and the accelerated development of the Internet of Things (IoT), the world is moving towards the use of Smart Mobile Devices (SMD). However, some of these devices remain poor in resources, unstable in terms of connectivity, reduced response time and constrained by short battery life. In addition, the stability of different routes built by routing protocols is affected because of the mobility of these devices. To deal with all these constraints, a simple solution for all these nodes that are poor in computing and storage resources is to offload a set of heavy tasks to another node in the network or in the periphery of the network (e.g. an Edge Server) to process them. In general, Edge Servers (ES) offer powerful capabilities such as big storage space and efficiency of computing. They are usually installed with multiple hosts and use virtual interfaces to access storage and computing features.

The calculations offloading solution runs up against the problem of the execution environment incompatibility and divergence between different systems managing devices. Indeed, before each source node sends its tasks to be executed remotely, the other distant node must be equipped with the proper processing environment which is represented by the virtual machine of the source node. Furthermore, In a MANET environment, the second target and powerful node (i.e. ES) intended to perform a number of tasks of the first node with limited resources may be out of radio range of this latter. So the use of a routing protocol to determine a route to the ES seems inevitable. Indeed, there is a very large number of protocols oriented towards MANETs [1] or even more, oriented towards IoT. In our situation, we adopted the OLSR protocol. This choice is motivated by the proactive nature of this protocol. Indeed, we take advantage of the periodic exchange of control messages in order to be able to disseminate information on the processing capacity of nodes throughout the network. In addition, knowing the times when these control messages were sent gives us an almost exact idea of the available bandwidth.

Therefore, the aim of this research is to propose an approach that allows the migration of Virtual Machines (VM) between different nodes of the network. The

E-mail: a.ouacha@um5r.ac.ma

mechanism proposed here is based on the splitting of each VM into several pieces that will then be sent by small subsets without disrupting the network operation. to this end, the information exchanged by the routing protocol is used to build metrics that serve as the basis for finding the most stable routes with sufficient bandwidth and also to find the ideal time instants where the parities of VM are sent.

The main contributions of this article are summarized as follows:

(a) For an almost precise determination of the predictable available bandwidth at each node, we start with an analysis of the times when control messages are sent and received from neighboring nodes. This allows us to learn about times when there is no traffic.

(b) A virtual machine is then split to a number of equal sized pieces. An algorithm is used to know the number of pieces as well as the time instant when the sending will be started depending on the available bandwidth.

(c) to take into account the nodes mobility, another algorithm is used to determine the time remaining before the link between the two neighboring nodes is broken. This is to avoid paths to the Edge Server before sending a total of a subset of all pieces of the VM.

(d) We study the validity of our approach and the performance of our improved version of the OLSR protocol by performing intensive simulations by the NS-3 simulator.

The rest of this paper is organized as follows. The second section provides a brief overview of the basic operation of the OLSR protocol and the third section synthesized several research related to our objective. The fourth section offers a detailed explanation of the proposed approach of virtual machine migration whereas the fifth section presents the various simulations carried out to test and prove the validity of our approach as well as the different obtained results. Finally, the article’s conclusion is provided in the sixth section.

2. OLSR OVERVIEW

Optimized Link State Routing (OLSR) [2, 3] is a proactive protocol that provides certain improvements on the basic principle of link state algorithms such as Open Shortest Path First (OSPF) [4] in order to reduce the amount of broadcast in an Ad Hoc context [5]. Indeed, the OLSR minimizes network flooding by reducing redundant retransmissions and reduces the size of the exchanged packets. In fact, OLSR is essentially based on the concept of Multi Point Relay (MPR), a subset of one-hop neighbors that allows all two-hop neighbors to be reached. Thus, during a broadcast, all the neighbors receive and process the message but only the nodes chosen as MPR retransmit it. Since the OLSR protocol is proactive, each node has permanently a view of the topology of the network. The consistency of this vision is ensured through periodic dissemination of information on the stats of links. Thus, a node receiving this information updates its view of the topology and applies the shortest path algorithm to choose the next hop to each destination. In what follows and as illustrated in figure 1, we present the steps allowing the construction of this topology.

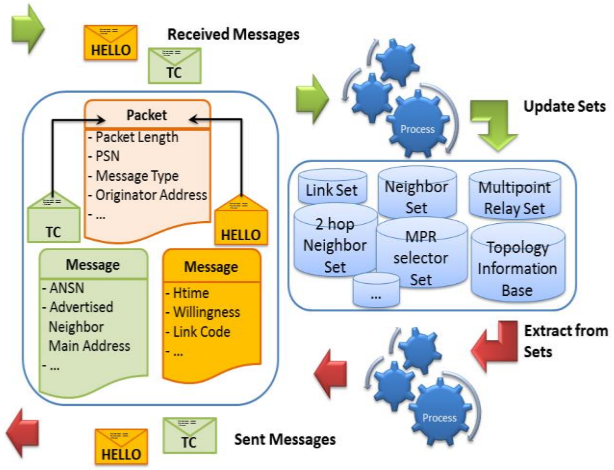

Figure 1. HELLO and TC Messages Transmission.

A. *Neighbor Sensing*

This is the direct and symmetrical neighborhood discovery process that is performed through the periodic broadcast of HELLO type messages containing neighborhood information as well as links state. This control message is intended exclusively for nodes in the first neighborhood and is therefore never retransmitted. In this way, a node constructs the list of one-hop neighbors N (Neighbor Set - red color nodes in figure 2) and also the list of two-hop neighbors N2 (2-Hops Neighbor Set - blue color nodes in figure 2)

B. *Selection of Multipoint Relays*

This selection is based on the choice of the subset of nodes from the set N allowing reaching all the nodes of N2. This subset is called Multipoint relay Set (nodes with green color in figure 2). The resulting subset is announced to all neighbors in subsequent HELLO messages.

C. *Declaration of Multipoint Relays*

MPR nodes broadcast Topology Control (TC) messages, allowing all nodes to build an information base on the topology of the network. TC messages are transmitted at regular intervals and declare the set MPRSS (Multi Point Relay Selector Set), that is to say the set containing the neighbors who have chosen the origin node of this message as MPR. The information on the topology of the network received in the TC messages is recorded in the topology table.

D. *Routing Table Calculation*

The neighbor table and the topology table are used for the calculation of the routing table which is based on the shortest path algorithm. Any modification of one of these tables causes the modification of the routing table.

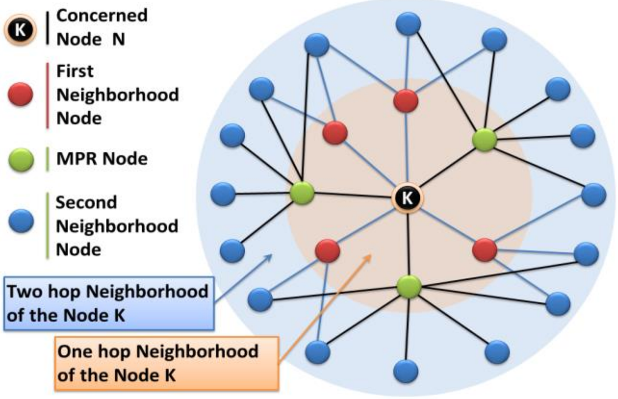

Figure 2. OLSR Nodes Type.

3. RELATED WORKS

In this section, we will present some previous researches on the migration of Virtual Machines (VM) to another node in a IoT environment. Indeed, the authors of [6] propose a resource-aware virtual machine migration technique whose main objective is to provide uninterrupted service to the IoT environment. To select the suitable target server, the authors are based on the resource utilization and job arrival rate for each possible destination server. Moreover, they are based on clustering servers to detect any sudden change in the sensing environment, which enables Cloud to accommodate these changes by virtual machine migration. The authors of [7] suggest a framework for the allocation and migration of virtual machines that exploits performance-to-power ratios (PPRs) for different types of hosts. By achieving the optimum balance between host usage and power consumption, this framework is able to ensure that hosts are operating at the most efficient energy usage levels. Given that the performance of VM migration is affected by VM size, running application’s dirty rate and the number of migration iterations, the authors of [8] present a study where they evaluate the performance impact of these migration control parameters. Indeed, they explore and analyze the parameters that affect performance of the VM migration technique. In order to process IoT data in real-time, the authors of [9] exploit the transmission rush hour and off-peak time to find the computing power they need to efficiently place virtual machines. Thus, they propose to use the metaheuristic algorithm allowing finding the optimal position of VMs and the number of VMs in each physical machine (PM). The solution they come up with must also ensure that the cost of resources is minimal and can process the most data. The authors of [10] propose an improvement of a pre-copy algorithm for migration of virtual machines to calculate the optimal total migration time and downtime using the flowing three proposed models: compression model, prediction model and performance model. In [11], the authors propose a method of migration of VM, named Ada-Things, which is determined by its workload characteristics. Specifically, based on the variation in the current rate of thirty pages of memory in IoT applications, Ada-Things can adaptively select the most suitable migration method to copy the memory pages. The authors of [12] propose knowledge based reduction method for Virtual machine assignment and consolidation. In order to reduce dissipated energy and get better quality of service, the proposed method chooses the Virtual machines from the overloaded host and migrates the Virtual machines to other physical host. The authors of [13] propose a VM migration algorithm that takes into account the cost of communication between VMs in a normal operating situation. The algorithm proposed in this research consists of two steps: the first one is to select the VMs to migrate based on the communication and occupied resources factors of the VMs. The second step aims to determine the destination host of the VMs to migrate. From an energy point of view, several studies propose approaches that aim to improve the energy consumption of heterogeneous MANET networks [14, 15]. Indeed, in [14] the authors propose cross-layer designed Device-Energy-Load Aware Relaying framework, named DELAR, whose objective is to achieve energy savings from multiple facets, in particular at the level of sensitive routing, power supply, transmission planning and power control. On the other hand, to take into consideration the constraints dictated by the nature of DTN (Delay-Tolerant Network) nodes and which are in most cases limited energy devices that operate on battery power, the authors of [15] propose a message-based algorithm complement to existing routing protocols such as Two-Hop Routing (2HR) and Epidemic Routing (ER). The approach adopted in their research also aims to improve energy efficiency through intelligent transfers based on the lifespan of individual messages and delivery requirements.

Unlike these researches, our research seeks to take advantage, on the one hand, of the control information exchanged between the nodes of the network. It uses the available bandwidth that can be computed based on the OLSR control messages. On the other hand, it takes advantage of the knowledge of the remaining time before the links are interrupted to use only the most stable routes while nodes are mobile.

4. PROBLEME FORMULATION AND RESOLUTION

A. *Prolbem Context*

An IoT environment is made up of the interconnection of a very large number of devices, most of which have limited computational, communication and storage resources. To overcome those weaknesses, low-resource devices can seek the help of other devices with high performances. Once a powerful machine is chosen to be the target where tasks are sent to be processed, the source machine (node with low performance) must also send the appropriate environment that is represented by a virtual machine (VM). However, VMs are, more often, very large size. This requires finding a mechanism that allows sending very large amounts of data without paralyzing the network operating.

Indeed, the approach adopted in this paper is to cut a VM into several pieces of equal size and use the routing table built by OLSR to transport these different pieces in small groups to the destination node (Server Edge for example). In order to ensure the reception of the different groups of pieces, the sending of each group (with variable cardinal) depends essentially on the Predictive Available BandWidth (PABW) and the predictive lifetime (RTTQ: Rest Time To Quite) of each link constituting the route between the source and destination nodes. So, the determination of the number of pieces of the VM contained in each group as well as the time instant when the sending will begin is calculated depending on two metrics PABW and RTTQ of which the way they are calculated is illustrated in the following sections.

As an example, figure 3 shows a situation where we have an Edge Server near MANET networks. So, network devices can send their data to be executed on this server.

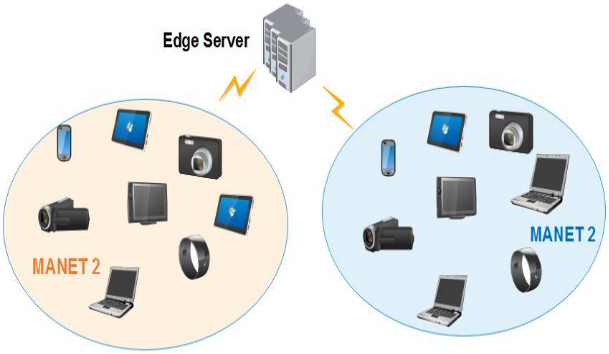

Figure 3. MANETs and Edge Server.

As the medium used is wireless transmission, the virtual machine sending may experience the problem of losing parts due to the disruption that this type of transmission may experience. Since the operation of the virtual machine essentially depends on receiving all pieces of the machine, the use of a lost packet detection and recovery technique is inevitable. Indeed, the use of an error correction code or of a random network coding [16] presents a quick solution. For example, we can transmit the acknowledgment mechanism to signal to the node that is transmitting its virtual machine the numbers of packets that are correctly received.

B. Predicted Dynamic Available Bandwidth

In a network where the routing protocol is the OLSR, the HELLO and TC messages are sent during regular time intervals. Knowledge of the time instants of sending the first HELLO and TC messages and the time instants of receiving the first HELLO message from each neighbor and the first TC message from each node of the network as well as the knowledge of the size of these messages, combined with knowledge of the physical characteristics of the node (CPU, RAM, WiFi type, etc.) gives us an idea of the calculation of available bandwidth.

Indeed, in this subsection, we will illustrate how the maximum available bandwidth ($MABW$) is calculated. Thus, in a MANET environment where the routing protocol is the OLSR, and as illustrated by figure 4, the basic idea is to extract the largest sub-time interval from an interval $[T_0, T_0 + 2[$, with duration of two (2) seconds, where there is no control traffic. T_0 represents the time when the first control message was received or sent by the node k

Let's denote by t_i^k the time instant when the packet i containing a HELLO or TC control message was received or sent by the node k, and by t_{i+1}^k the time instant when a packet $i+1$ containing other control message received by the same node k. Furthermore, if TR_i^t is considered to be the time it takes to process the packet i, then the available bandwidth in the time interval $[t_i^k, t_{i+1}^k[$ will be represented by $ABW_{i,i+1}$ calculated by:

$$ABW_{i,i+1} = t_{i+1}^k - (t_i^k + TR_i^k) \tag{1}$$

It should be noted that in the practical case, we always has $t_{i+1}^k \geq t_i^k + Tr_i^k$ because otherwise cannot take place due to a collision between the i and $i+1$ packets. In addition, the sending of a HELLO message occurs only once in the interval $[T_0, T_0 + 2[$ and sending a TC message occurs only once in the interval $[T_0, T_0 + 5s[$. In other words, sending a HELLO message happens only once in a two-second time and sending a TC message happens only once in a five-second period.

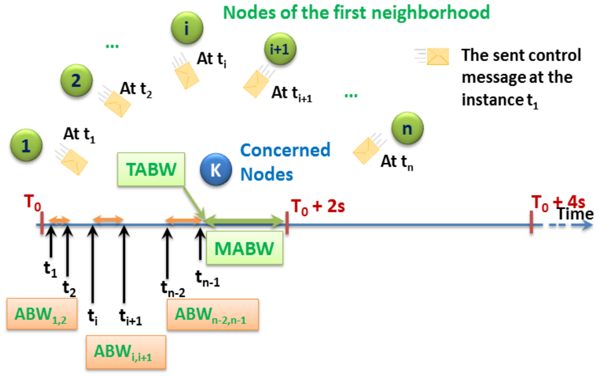

Figure 4. Avalable Bandwidth Computation.

If Np is the number of packets received and sent during the 2 second time interval $[T_0, T_0 + 2[$, the Maximum Available Bandwidth ($MABW$) for the node k will be represented by the largest time interval $ABW_{i,i+1}$ defined by:

$$MABW_k \ = \max_i(\,ABW_{i,i+1}) \qquad (2)$$

With $i \ = T_0, t_0, t_1, \dots, t_{n-1}, T_0 + 2s$

Since HELLO messages are sent every two seconds, the Predictable Available Bandwidth ($PABW_j^k$) in the j^{th} interval for the node k will be defined by:

$$PABW_k = \ MABW_k + (2 \times j) \qquad (3)$$

C. Predictable Rest Time to Quit (RTTQ)

The method used to calculate the RTTQ is the one used in [17]. Its estimation is based mainly on cartesian coordinates of the nodes exchanged through HELLO control messages. It is calculated according to algorithm 1 below translating figure 5:

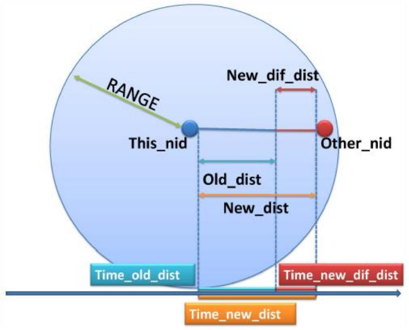

Figure 5. Predict Remaining Time Estimation.

```
Algorithm 1 :          RTTQ Computation
Initialisation
  Define the constant MaxValue

Begin
  Calculate the new distance NewDist
  Get the instance TimeNewDist
  DeltaD ← NewDist - OldDist
  DeltaT ← TimeNewDist - TimeOldDist

  If (DeltaD < 0) then
    RTTQ ← MaxValue
  Else
    RTTQ ← (RANGE - NewDist)/DeltaD × DeltaT
  End If

  OldDist ← NewDist
  TimeOldDist ← TimeNewDist

  Return RTTQ
End
```

The $RTTQ$ Metric is used to avoid links that risk to be interrupted and disturbing the stability of the routes while the node k is sending a set of pieces VM_i into the target node.

D. Routing Based on MABW and RTTQ

Below, a modified version of the $OLSR$ protocol is designed so that the delivery of packets representing the VM_i to the target node takes into account the $MABW$ and the$RTTQ$. Thus, the routing of these large amounts of data must be carried out without disrupting the operation of the network.

1) Virtual machin splitting

VM of size x is split into several pieces VM_i of equal size x_i whose total number P is calculated by:

$$P = \begin{cases} \dfrac{x}{x_i} & if\ x\,\%\,x_i = 0 \\ \left[\dfrac{x}{x_i}\right] + 1 & otherwise \end{cases} \qquad (4)$$

With $[\dots]$ represents the entire part and $\%$ represents the rest of the entire division. If $x \ \% \ x_i \neq 0$, the last piece x_p may be smaller in the size than the other pieces x_i

2) Number of parts MV_i and the time of sending

The number of parts to be sent by the node k is calculated based on the maximal available bandwidth ($MABW_k$), the Time when this $MABW_k$ ($TABW_k$) begins, and also the time instant (At_k) when the node k wants to send pieces of the VM. Thus, the Number of Packets To Send ($NPTS_k$) is calculated according to several cases illustrated by figure 6.

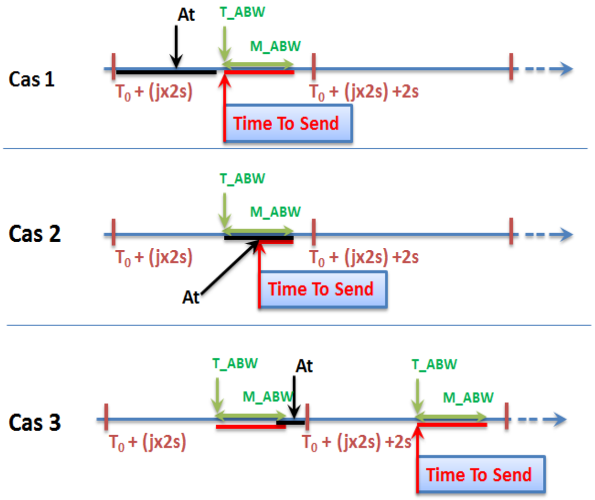

Figure 6. Time and Numbre of Packets to send estimation.

- Case 1 $At_k < TABW_k$

  The full maximal available bandwidth will be used and the number of VM_i to be sent will be calculated by:

$$NPTS_k = \frac{MABW_k}{(Time_{ToSendOnePacket\ i} + Tr_i^k)} \qquad (5)$$

In this case, the time instant when the node k wants to send the pieces of the VM coincides with a traffic exchanges. So the start of the sending will be shifted to $TABW_k$

- Case 2 $At_k > TABW_k$ And $At_k + Tr_i^k < TABW_k + MABW_k$

  Just a part of the maximal available bandwidth will be used and the number of VM_i to send will be calculated by:

$$NPTS_k = \frac{MABW_k - (At_k - TABW_k)}{(Time_{ToSendOnePacket\ i} + Tr_i^k)} \quad (6)$$

  In this case, the node k that wishes to send the pieces of the VM starts sending it immediately.

- Case 3 $At_k > TABW_k$ and $At_k + Tr_i^k > TABW_k + MABW_k$

  The full available bandwidth will be used and the number of VM_i to be sent will be calculated by the equation 5.

  In this case, the instant when node k wants to send the parts of the VM coincides with a traffic exchange. So the start time of sending will be shifted to the next interval and will be initialized by $TABW_k + 2s$

Here $T_{ToSendOnePacket\ i}$ is the transmission time of a single piece VM_i by a node k and Tr_i^k is the processing time of the packet containing the piece VM_i which will be estimated practically.

3) *Pieces routing*

The routing of the different pieces will be mainly done in two steps:

- The first step is to determine the next intermediate node (next hop) to which packets will be delivered. For this reason, the routing table constructed by the OLSR protocol is consulted. Note that during the construction of the entries routing table, the RTTQ is taken into account to ensure the stability of the links and to minimize the consequences of the nodes mobility. So, the routes are constructed in such a way as to avoid using as the next hop each node that risks leaving the neighborhood in the packets sending process.
- The second step is to determine the $NPTS_k$ when pieces will be sent. For this purpose, the algorithm illustrated by the previous subsection will be used.

5. SIMULATION AND RESULTS

After the problem has been formulated, several simulations are done to test the formulated approach. Thus, the simulations are carried out by NS3 [18]. It is a simulator that has proven its efficiency in wireless network environments. It is also loosely used by the scientific research community to test and validate several approaches. In this section we will present the environment of our simulations and will discuss the obtained results.

A. *Simulations Parametres*

In order to test our approach in mobile environment, the simulation environment consists of a network of 50 mobile nodes according to the Random Way Point (RWP) mobility model [19] with a maximum movement speed equal to 20 m/s and the pause time varied from 0 second to 20 seconds with a step of 5. Each node is equipped with a single wireless communication interface whose link layer protocol is 802.11b. The routing protocol adopted in our network is OLSR. In order to simulate sending VMs, a node is randomly chosen to be the target of a set of 250 packets. Each packet, which is 1000 Bytes in size, simulates the sending VM_i parts from the same VM. To give more credibility to our results, each simulation is repeated several times with a change of the source. For more information on the simulation environment, Table I presents more parameters.

TABLE I. SIMULATIONS PARAMETERS

Parameter	Value
Network Simulator	NS -3.29
Protocol	OLSR
Simulation Time	100 Second
Simulation Area	1500 ×1500 m^2
Number of Nodes	50
Transmission Range	500 m
Mobility Model	Random Waypoint
Max Speed	20 m/s
Pause Time	0, 5, 10, 15, 20 Second
Data Packet Size	250, 500, 750, 1000, 1250, 1500, 1750, 2000 Byte
WiFi Mac Protocol	802.11b

B. *Avalable Bandwidth*

A first series of simulations is made by NS3 where the instant t_i^k of sending or receiving of each control message is recorded. In this case k represents a node of the network chosen randomly. Figure 7 shows the time when a packet was received or sent by the node k depending of the order number where this packet is received or sent. The obtained results confirm that during each 2 second span interval, there is a sub-interval (marked by green color) where traffic is absent. It is this sub-interval that will represent our Maximal Available Bandwidth ($MABW_k$). So the sending of each set of VM_i packets must be done within this interval.

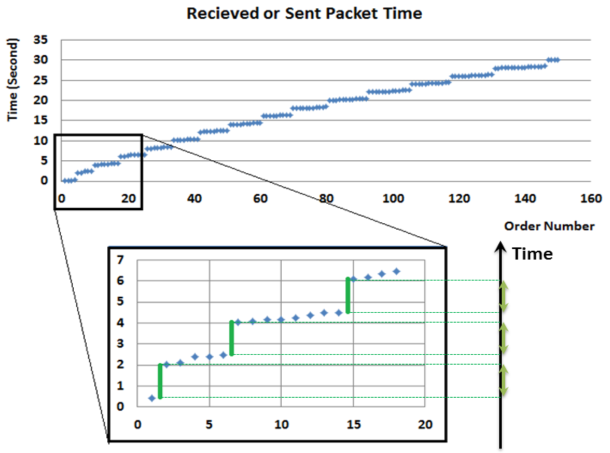

Figure 7. Example of packets sent and recieved time for one node..

C. *Transmition Time Computation*

To be able to calculate the time instant when the sending will begin and also the number of packets to be transmitted when the maximum available bandwidth ($MABW_k$) and the time instant when this available bandwidth begins ($TABW_k$), a second series of simulations are made to give an exact value of time required to process each piece VM_i. Indeed, Table II shows the time values corresponding to each size of pieces VM_i.

TABLE II. PROCESSING TIME

Pocket Size	Processing Time
250	0,0157
500	0,0250,
750	0,0348
1000	0,0437
1250	0,0534
1500	0,0626
1750	0,0815
2000	0,0817

Figure 8 shows a linear variation in transmission and processing time as the packet size increases. Therefore, the value of TR_i^k can be deduced from the forecast trendline whose equation is defined by equation 7. The factors α and β can be defined from two graph points.

$$y = \alpha x + \beta \tag{7}$$

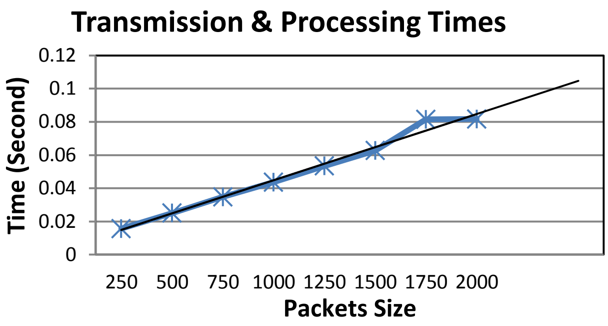

Figure 8. Transmission and processing time depending on packets size.

D. *Results and Discussion*

The results in figure 9 show a comparison between the numbers of packets generated, sent and successfully received for two routing protocol version as a function of pause time. Both versions are based on the OLSR. The first protocol version that implements our approach and which is based on the Maximum Available Bandwidth ($MABW$) and the lifetime of the links ($RTTQ$) shows a relevant improvement in the number of successfully received packets compared to the standard version of the OLSR protocol. Our version of the protocol also shows a successfully received packet rate which remains almost constant even if the degree of mobility changes. In other words this version remains immune to mobility unlike the standard version which shows a decrease in the number of successfully received packets as mobility increases.

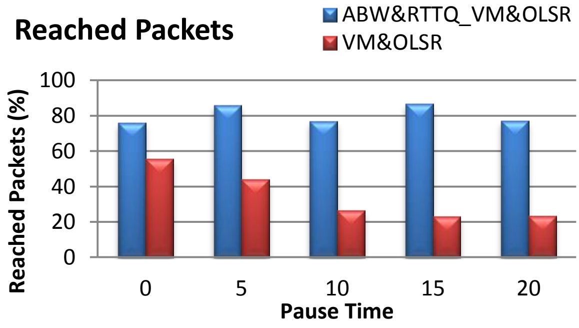

Figure 9. The percentage of Reached Packets for the two versions of the OLSR protocol .

Figure 10 shows the average packet delay for the two versions of the protocols. Our version presents a very encouraging average delay compared to the standard version of the protocol. Indeed, since the ABW&RTTQ_VM&OLSR version only starts sending packets when the bandwidth is available, this avoids collusion with the control traffic. Thus, our version of the protocol avoids the retransmission of packets, which considerably reduces sending times, unlike the standard version of the VM&OLSR protocol.

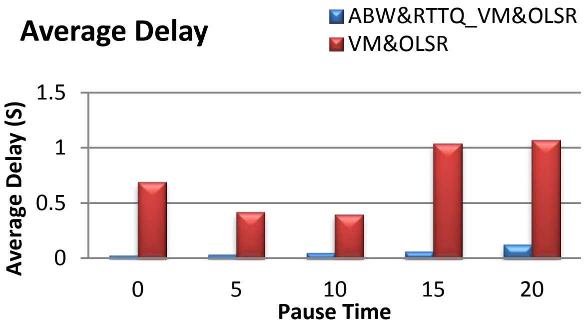

Figure 10. Average Delay for the two versions of the OLSR protocol.

6. CONCLUSION

This study focuses on the case where we have an intelligent mobile device in MANET network that has a processed data set and whose compute and storage resources are limited. These data can be offloaded to be processed on an Edge Server near the network. In order to migrate the virtual machine containing the environment for processing the data, we have proposed in this paper a solution based jointly on the prediction of the maximum available bandwidth and the remaining time for the link between different pair of nodes in the route constructed by the OLSR protocol in an IoT environment will be maintained. After giving approaches used to calculate the available bandwidth ($MABW$) and time remaining to leave the neighborhood ($RTTQ$) and how they are used to optimize the function of the OLSR protocol, several simulations are carried out by NS3 to. At results, we begin by the calculation of the correct values used for the calculation of the $MABW$ metric. After that, we test and validate the effectiveness of our approach. The results obtained show that the modified version of the OLSR routing protocol (ABW&RTTQ_VM&OLSR) presents considerable improvements with regard to the rate of packets successfully delivered and the routing details between the node pairs of each route compared to the standard version of the OLSR protocol. This prompts us to say that our approach is more efficient when it comes to transporting large amount of data and more specifically large virtual machine.

However, several other improvements are possible. We plan to generalize our study to the multi-server case while introducing new relevant parameters such as network condition, device mobility and wireless communication interference. By this, we plan to take into consideration the case of an adaptive offload policy, in which multiple tasks can be performed in parallel by the mobile nodes themselves as well as multiple Edge Servers.

REFERENCES

[1] H. Ashtiani, H. M. Pour, and M. Nikpour, "A Survey of MANET Routing Protocols in Large-Scale and Ordinary Networks," *Global Journal of Computer Science and Technology*, 2010.

[2] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, 2001, pp. 62-68.

[3] T. Clausen and P. Jacquet, *RFC3626: Optimized Link State Routing Protocol (OLSR)* vol. RFC3626: RFC Editor, 2003.

[4] J. Moy, "OSPF Version 2," *IETF Tools*, p. April 1998, 1998.

[5] S. Corson and J. Macker, *RFC2501: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*: RFC Editor, 1999.

[6] G. J. L. Paulraj, S. A. J. Francis, J. D. Peter, and I. J. Jebadurai, "Resource-aware virtual machine migration in IoT cloud," *Future Generation Computer Systems*, vol. 85, pp. 173-183, 2018/08/01/ 2018.

[7] X. Ruan, H. Chen, Y. Tian, and S. Yin, "Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds," *Future Generation Computer Systems*, vol. 100, pp. 380-394, 2019/11/01/ 2019.

[8] A. Bhardwaj and C. R. Krishna, "Impact of Factors Affecting Pre-copy Virtual Machine Migration Technique for Cloud Computing," *Materials Today: Proceedings*, vol. 18, pp. 1138-1145, 2019/01/01/ 2019.

[9] S.-Y. Huang, C.-C. Liao, Y.-C. Chang, and H.-C. Chao, "Virtual Machine Placement Based on Metaheuristic for IoT Cloud," Cham, 2017, pp. 472-482.

[10] M. Patel, S. Chaudhary, and S. Garg, "Performance Modeling and Optimization of Live Migration of Virtual Machines in Cloud Infrastructure," in *Research Advances in Cloud Computing*, S. Chaudhary, G. Somani, and R. Buyya, Eds., ed Singapore: Springer Singapore, 2017, pp. 327-350.

[11] Z. Wang, D. Sun, G. Xue, S. Qian, G. Li, and M. Li, "Ada-Things: An adaptive virtual machine monitoring and migration strategy for internet of things applications," *Journal of Parallel and Distributed Computing*, vol. 132, pp. 164-176, 2019/10/01/ 2019.

[12] R. Bhaskar and B. S. Shylaja, "KBR: Knowledge Based Reduction Method for Virtual Machine Migration in Cloud Computing," *Procedia Computer Science*, vol. 165, pp. 708-716, 2019/01/01/ 2019.

[13] J. Zhu, J. Wang, Y. Zhang, and Y. Jiang, "Virtual machine migration method based on load cognition," *Soft Computing*, vol. 23, pp. 9439-9448, 2019/10/01 2019.

[14] W. Liu, C. Zhang, G. Yao, and Y. Fang, "DELAR: A Device-Energy-Load Aware Relaying Framework for Heterogeneous Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 1572-1584, 2011.

[15] V. K. C. Manam, V. Mahendran, and C. S. R. Murthy, "Message-driven based energy-efficient routing in heterogeneous delay-tolerant networks," presented at the Proceedings of the 1st ACM workshop on High performance mobile opportunistic systems, Paphos, Cyprus, 2012.

[16] C. Servin, *Réseaux et télécoms*, 4 ed., 2013.

[17] A. Ouacha, N. Lakki, J. E. Abbadi, A. Habbani, B. Bouamoud, and M. Elkoutbi, "Reliable MPR selection based on link lifetime-prediction method," in *2013 10th IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC)*, 2013, pp. 11-16.

[18] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15-34.

[19] R. R. Roy, *Handbook of Mobile Ad Hoc Networks for Mobility Models*, 1 ed.: Springer US, 2011.

Ali Ouacha is professor of computer science at the Faculty of Science of Rabat (FSR) Universities Mohamed V (UM5) since 2018. He received his doctorate degree from the Mohammadia School of Engineering (EMI). Ouacha is a permanent member of the Intelligent Processing and Security of Systems (IPSS) team of Computer Science Department at the FSR. His research is currently focused on optimizing the performance of routing protocols in an Internet of Things (IoT), Mobile Edge Computing (MEC) and Mobile Ad-hoc Networks environment. He is the author and co-author of several publications in international journals and conferences. Ouacha is a member of the organizing committees of several scientific events (Conferences and Congresses). He is also a member of the Technical Program Committee (TPC) of several international conferences and journals.