Assessment of MANET Broadcast Schemes in the Application Context of Multiplayer Video Games

Ahmad Sardouk and Sidi Mohammed Senouci
Orange Labs – CORE/M2I
2, Avenue Pierre Marzin
22307 Lannion, France
ahmad.sardouk@orange-ftgroup.com,sidimohammed.senouci@orange-ftgroup.com

Nadjib Achir and Khaled Boussetta
L2TI – Institut Galilee – University Paris 13
99 Avenue J-B Clement
93430 Villetaneuse, France
nadjib.achir@l2ti.univ-paris13.fr,khaled.boussetta@l2ti.univ-paris13.fr

ABSTRACT

Taking into consideration the popularity of multiplayer games and the tremendous success of wireless networks, we strongly believe that future game engines will benefits from the integration of MANET (Mobile Ad hoc NETworks) technology [2]. Indeed, in addition to the social interests of playing with close located persons, MANET technology will allow players, like a group of kids in a playground, to easily improvise a LAN multiplayer party without any needs for an existing Wireless network infrastructure. In order to make such scenario a common based reality, several issues still have to be addressed. Among which is the MANET's broadcast problem. In fact, broadcast is a fundamental network service for multiplayer LAN mode. But up to now, the most used scheme for broadcasting in MANET relies on flooding. Unfortunately, this simple scheme leads to high redundancy and collisions. In the context of a multiplayer party supported by mobile terminals, this broadcast storm problem [10], could result into a dramatic reduction of the network energy-based lifetime and to a high variability of the transmission delay. The latter one is know to be a crucial QoS parameter which can significantly affect the game play. In this paper, we evaluate the performances of alternatives MANET broadcast schemes in the context of multiplayer video games. We analyze the capacity of *simple flooding*, *distance based*, *probability based* and *self pruning* broadcast schemes to satisfy the QoS constraints required by multiplayer games like FPS. Assessment simulations are conducted for both Client/Server and P2P network game architectures.

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Computer–Communication Networks – Network Architecture and Design, Network Protocols, Distributed Systems.; C.4.1 [**Computer Systems Organization**]: Performance of Systems - Performance attributes Reliability, availability, and serviceability.

General Terms

Algorithms, Performance, Reliability.

Keywords

Multiplayer video games, ad hoc networks, broadcast.

1. INTRODUCTION

Up to now, the multiplayer mode is generally supported by wired communication infrastructures. Wireless links, based on cellular technology (e.g. UMTS) or WiFi, could be used as well but only as a last link in the transmission path. This is typically the case with laptops, PDAs or recent consoles such as Nintendo Wii, Nintendo DS, Sony's Portable Playstation (PSP) and Microsoft Xbox, all whether offering a build-in WiFi interface or could be equipped by a USB WiFi connector. Thus, in this last generation of consoles LAN multiplayer mode based on WiFi is now available. However, only one single hop is supported. In other words, in order to play together, all players must be located within the same radio coverage. Multi-hop radio communications using a MANET (Mobile Ad-hoc NETworks) [2] technology is not supported yet by the actual generation of consoles.

MANETs are an ultimate generalization of the wireless networks. Indeed, ad-hoc networks technology allows spontaneous creation of data networks by exploiting the capacity of the mobile terminals to communicate directly between them without the need of a centralized infrastructure. Terminals forming the ad-hoc network should play the role of router in order to ensure the gradual transfer of data packets. Moreover, the network management tasks (auto-configuration, security, etc.) are carried out by the terminals themselves in an entirely distributed way.

Taking into consideration the popularity of LAN multiplayer games and the tremendous success of wireless networks, we claim that future game engines will benefit from supporting the MANET mode. In addition to the social and cultural interest of playing with close located persons, MANET technology could offers the comfort of mobility for a much higher number of potential players than actually supported by last generation of commercialized consoles.

Other advantages of MANET technology for LAN multi-party gaming are the facility of deployment, the autonomic reconfiguration, the self-healing and the possibility of capacity addition without preliminary planning. All these elements make us strongly believe on the suitability of integration of MANET technology in future games terminals (e.g. portable consoles). A scenario, like a size varying group of kids equipped with portable consoles supporting MANET services and wishing to play a WLAN party in a playground (without necessitating any particular fixed network infrastructure) will be possible.

In order to make such scenario a common based reality in the near future, several issues still have to be addressed by the research community. Several difficulties come from the inherent constraints that characterize the MANET technology. In particular the mobility of the core network, the energy limitations and the variation in terminal resources sharing. All these problems lead to a variable connectivities and potentially high fluctuation in the provided Quality of Service. On the other hand, multiplarty games are very sensitives to resource availability. They are very demanding on CPU and thus high consuming on energy. In addition, the Game play could suffer form significant degradation in case of variables delays, which is typically the case in MANET (due the MAC layer nature and the mobility of the users). Hence, certain technological issues remains to be solved to hope a real deployment of such MANET in portable game terminals.

In this paper, we address the MANET broadcast problem in the application context of a LAN multiplayer games. Indeed, independently of the game architecture, in a LAN party, a large amount of traffic generated by the entities evolved in the game are exchanged through MAC broadcast messages. A simplest and commonly used scheme for broadcast in MANET is based on a simple flooding mechanisms. Unfortunately, when several nodes are involved in the broadcast process and when all of theme compete to transmit the simple flooding scheme could raise a well known problem, refereed as the broadcast storm problem [2].

Several mechanisms have been proposed in the literature in order to optimize the broadcast transmission in MANET, in term of reliability, efficiency, energy consumption and ability to satisfy the application QoS constraints. Assessment of the proposed schemes were conducted for several applications like video or audio streaming, file sharing, communications for military application or for civilians through wireless sensors networks. But in the best of our knowledge, none of the previous work have focused on the context of video games.

In this paper we evaluate the performances of four MANET broadcast schemes in the specific context of multiplayer video game support. Each scheme is representative of a particular approach. We analyze their ability to satisfy the requirement of multiplayers games in term of reliability, delays and energy consumption. We propose the assessment of different configuration cases depending on the game architecture, the network density and the parametrization of the broadcast schemes.

The remainder of this paper is organized as follow: in the next section, we detail the two common network game architectures, namely : C/S and P2P. Related works on MANET broadcasting schemes are reviewed in section 3. Section 4 gives a detailed description of our simulations scenarios. Obtained results of the assessments of four MANET broadcast schemes in the application context of multiparty video gaming are exposed and analyzed in section 5. Finally 6 summarizes the main contributions of this work and provide some future prospects.

2. GAME ENGINE ARCHITECTURES

The game engine is a software which provides all the functionalities necessarily to play video games using a hardware terminal (e.g. console or a PC). Typically, it includes several engines, where the most common ones are the 2D/3D graphic render, the physics engine, the sound module, the Artificial Intelligence engine, the scene graph, the animation module, the threading and the networking systems. The latter one is particulary important when the multiplayer mode is proposed by the game. Basically, the networking system is in charge of transmitting the game's data between all the entities involved in the game. The nature of these entities depends on the architecture of the game: Client/Server or Peer-to-Peer.

Actually, most of the multiplayer games are using a centralized Client-Server (C/S) architecture. In this case, one part of the game software/program is located in each client, and the second part in a central server, which acts as a master of the game. The clients are responsible of collecting and sending, to the server, all primitive commands related to the players actions within the game, such as avatar control commands (movements, shooting, etc.). On the other hand, the server is responsible for computing the game actions received from each client in order to compute the overall game state. This overall game state is then shared back (e.g. using broadcast scheme in LAN), by the server, to all the clients. The C/S architecture can provide good performances (consistent overall game state) only if we consider both stable and low latency connections between the clients and the server, which is the case in LAN environment. However, when we consider a MANET environment, the C/S architecture can raise many problems. In MANET clients' mobility and channel condition can strongly affect the connectivity among the mobile. Thus, the game playability (gameplay) can strongly be degraded as the latency between the client and the server is increased. Indeed, the gameplay is very sensitive to the round-trip-time (RTT) between the clients and the server. Smaller is the RTT, better is the gameplay.

One alternative to using C/S architecture for multiplayer games is Peer-to-Peer (P2P) architecture. In Peer-to-Peer architecture, we do not have the notion of clients or server, but only equal peer nodes that can simultaneously act as a client and a server. In this case, every peer sends its local player actions to all other peers in a completely distributed manner, without the need of any central server. Thus, each peer is responsible of computing the overall game state from the received data. The main advantage of using this architecture resides in the inexistence of a central coordination/administration point. Thus, P2P architecture seems more appropriated in MANET environment since it is not necessary to maintain persistent connectivity with a server. However, it implies efficient synchronization process between the peers.

In both C/S and P2P architectures, broadcast is a fundamental communication service for LAN multiplayer games. Indeed, in C/S architecture, the server must regularly trans-

mit, using broadcast messages, the overall game state (e.g. object positions and speed, hits, score etc.) to all the clients. Similarly, in P2P architectures, each peer can broadcast its game state view to all other participant peers, so that all nodes can converge toward a synchronized game state.

In the next section, we give a short overview on main broadcast schemes in MANET environments.

3. BROADCAST SCHEMES FOR MANET

Different broadcast schemes for the MANET environment have been proposed in the literature [10] [6] [8] [9] [7]. Williams and Camp [11] categorize the broadcasting schemes into four families: *simple flooding*, *probability based methods*, *area based methods* and *neighborhood based methods*.

The first family, *simple flooding* (figure **??**), is considered as the default solution for packet broadcasting in ad hoc networks. In this scheme, each node will be required to rebroadcast each received packet, once time, to its neighbors. The main advantage of using this schemes is that simple flooding is the only way ensuring that each broadcasted packet is guaranteed to be received by every node in the network. However, when considering high density network, Ni and al. [10] shows that *simple flooding* may lead to serious message redundancy, high contention and collisions. This problem is referred as the broadcast storm problem.

The second family, *probability based methods*, includes two broadcasting schemes [10]: *probability based* and *counter based*. In the first scheme, *probability based*, each node is given a predefined packet re-broadcasting probability (p). Precisely, whenever a node receives a broadcast packet, it will rebroadcast it with a probability (p), or drop it with a probability ($1 - p$). Note, that if *p=1* then this scheme leads to *simple flooding*. In the second scheme, *counter based*, each node rebroadcast a packet if same copies of the packet have already been received for less than a given number of times. Thus, compared to previous broadcast schemes, the reduction on the number of redundant broadcasted packets that could be obtained by the *counter based* scheme will relies on the network density. Indeed, in a high dense MANET, few nodes will rebroadcast packets, while in a sparse topology a majority of nodes will have to rebroadcast the packets.

In the third family, which includes *area based methods*, packet's re-broadcasting decision will be made upon the estimation of the relative physical distance between nodes. Within this family, we can cite two schemes [10]: *distance based* and *location based*. In the first scheme, each node is given a predefined distance threshold (d). When a node receives a broadcast packet it estimates the physical distance separating him from the last relaying node, which sent that packet. The packet is dropped if the evaluated distance is less than d. Otherwise, the packet is rebroadcasted. The main problem of this scheme is due to the accuracy on distance assessment. One possible solution is to derive the distance from the measured signal strength of the received packet. Unfortunately, this assumption does not holds in a realistic ad hoc network. In *location based* scheme, each node adds its location information in the header of each sent packet. Thus, packet re-broadcasting is done only if the additional coverage area added by the receiver is greater than a given threshold. Unfortunately, this scheme makes the strong assumption that each node know its location (e.g. using the global positioning system – GPS). For costs reason, it is clear that such requirement for a geolocalization

system could not be reasonably fulfilled in the context of video gaming upon cheap portable consoles.

Finally, *neighborhood based methods* are based on the exploitation of the neighborhood nodes' informations. The mains schemes belonging to this family are: *sefl pruning* [6], *dominant pruning* [6], *scalable broadcasting* [8], *multipoint relay* [9], *Ad Hoc broadcast protocol* [7] and *Simplified Multicast Forwarding for MANET* [1]. In *sefl pruning*, each node piggyback, into each re-broadcasted packets, its adjacent nodes. Thus, nodes that received this packet can checks if their own adjacent nodes are the same than those piggybacked in the received packet. If yes, the packet is dropped. Otherwise, it is re-broadcasted. While *sefl pruning* exploit only one hop node information, the remaining schemes *dominant pruning*, *scalable broadcasting*, *multipoint relay*, and *Ad Hoc broadcast protocol Simplified Multicast Forwarding for MANET* requires that all nodes have knowledge of their neighbors with two-hop radius. Unfortunately, all of *neighborhood based methods* requires extra transmission overhead, especially in dense MANET.

In this paper, we evaluate the performances of some of the mentioned ad hoc broadcast schemes in the application context of multiplayer video games. From each family, we chose the simplest mechanisms, since we believe that reduced complexity is a fundamental criteria when considering implementation in portable consoles. Following these considerations, we chose to assess four broadcast mechanisms. Namely: *simple flooding*, *distant based*, *probability based* and *self pruning*.

Note that these mechanisms have already been assessed. Mainly, through simulations. However, in the best of our knowledge, all the evaluation studies have been conducted out of the particular context of video gaming. Traffic models, mobility models and performance metrics were specific to other applications, such as video or audio streaming, file sharing, VoIP, communications for military application or for civilians through wireless sensors networks. The specific traffic models characterizing multiplarty video games (like those of FPS [3]) the relationship between interacting nodes (which will depends on the architecture nature: Client/Server or P2P), the node mobility and the network density (which characterize playing places, like playground) are not considered in any other previous studies.

We integrate all these considerations in our work. Details on our experimentation scenarios are given in the next section.

4. SIMULATIONS SCENARIOS

In order to compare and evaluate the impact of simple flooding, probability based, distance based and self pruning mechanisms on the performances of video games over mobile ad hoc networks, we carry out a set of simulation tests. We have implemented and integrated the four mentioned broadcast schemes in the Glomosim 2.0 [5] simulator.

In our simulations setup, we have considered that each node has a radio propagation range of 120 meters and a channel capacity set to 2 Mb/s. To model a playing place, like a school playground, we have assumed a limited area characterized by a square with a side size equal to 300 meters. We have supposed four possible configuration scenarios, depending on the total number of nodes (which correspond to the players' consoles) in the area. Precisely, we have run simulations assuming successively 4, 8, 12 and 16

participant nodes. We believe that these values are representative of typical group sizes that should be observed for MANET's LAN parties.

In regards with radio coverage (120 m) and the area surface (300 × 300 m^2) we were faced to the initial node distribution problem. Indeed, when considering the small number of nodes scenario (e.g. 4 or 8) and assuming a uniform random distribution for initial nodes placement, sampling may leads to a disconnected graph topology (all nodes are not reachable). This scenario is not realistic in the general context of MANETs. And particularly, in the context of multiplayers games application. A realistic scenario would be to assume that before starting a party, gamers are distributed so that they form a connected graph. If their initial position when switching on the radio interface indicates that all the players are not reachable, we think that a realistic behavior would be that some players will move so that the connectivity is obtained for all the group.

Following this assumption, all initial nodes placements generated in our simulation executions are obtained with a connected graph. In other words, in the coverage area of each node, there is at least another node. Sampling that led to a disconnected graph are simply discarded.

For the same reasons explained before, the node mobility model that we implemented for our experiments relies on the hypothesis that all nodes movement is conditioned by the maintenance of the graph connectivity. Of course, in practise, such assumption may not holds. However, in this work we kept this assumption for simplicity reasons and clarity of our results and analysis. Reliability issues resulting from disconnections due to MANET's nodes movements are kept for our future work.

Following this analysis, we have discarded some classical node mobility models (e.g. random way point). Rather, we have propose an adapted gamers behavior mobility model. Basic assumptions relies on the fact that players will be concentrated on the game and thus should not move frequently. Since we consider pedestrians, then a moving node should correspond to a slowly walking person. Typically, we took an average speed of a moving node which varies between 1 and 2 meters per second. In our model, the trajectory of a moving node is linear with an average distance set to 10 meters. Whence this distance reached, the player will remain on place for a randomly chosen time. The average value of this pause time is about 10 seconds.

Two traffic models were implemented, each one is supposed to reflect the realistic behavior of transmitted data during a LAN multiplarty game. Each model is associated to the game architecture: C/S and P2P. In the first case, we set one node (node 0) as the hosting server. Broadcast messages are only generated by this node. We supposed an FPS (First Person Shooter) game gender (e.g. Counter Strike or Quake 3). Following approximatively the traffic model described in [3] for such games, we assumed that the server generate a Constant Bit Rate (CBR) traffic, where a packet size of 80 bytes is generated every 40 ms. The remaining nodes acts as clients, which send to the server unicast messages according to a CBR model. In each node, a unicast packet of size equal to 127 bytes is generated every 60 ms. We assumed that unicast packets are relayed using the AODV protocol. *multipoint Hello* messages being sent every 10 seconds. Note that we have chosen CBR model, rather than the Lognormal or the shifted Weibull models described in [3] to speed up the simulation time and to simplify the analysis.

In P2P case simulations, all nodes broadcast their local game information states. Since we did not found in the literature any traffic model for FPS LAN multiplayer games based on a P2P game engine, we have supposed that the generated traffic per node, should follow the client model in the C/S archiecture (since in a well designed P2P system, only local game state informations needs to be sent). That is, each peer node generates a broadcast message of size equal to 127 every 60 ms.

In order to evaluate the ability of the probability-based scheme to adapt to the network density, we have considered two probabilities values: $p = 0.8$ and $p = 0.4$. The first value is expected to provide good results in a sparse network, while the second one should be more adapted for a dense network.

Similarly, we have considered three configurations distance values for the distance-based version. Precisely, we considered the following cases: $d = 40$, $d = 70$ and $d = 100$ meters. Small value of d is expected to provide better results in the case of dense network.

Following all these assumptions, we have run extensive simulations for all possible scenarios described in this section. For each experiment, simulation duration times was set to 5 minutes. Results are exposed and analyzed in the next section.

5. RESULTS AND ANALYSIS

In figures 1 and 2, we denote the average packet delivery ratio for, respectively, client-server and peer-to-peer architectures. We plot the packet delivery by considering different number of players (gaming consoles) (4, 8, 12 and 16 players). For C/S architecture, the obtained results shows that, for all broadcasting shcemes, the packet delivery ratio is up to 75%. The better performances are obtained using simple flooding. We note also that, performances are affected by the given value of the re-broadcasting distance and the re-broadcasting probability. When we encrease the re-broadcasting distance, we observe a decrease in the packet delivery ratio, and when we increase the re-broadcasting probability, we observe an encrease in the packet delivery ratio. In P2P architecture, the observation are differents from C/S architecture. Here, the better performances are obtained using self pruning scheme, for 4 and 8 nodes. However, for 12 and 16 nodes the better performances are obrained when using distance based scheme (for $d = 100\ m$). Furthermore, we notice that enhancing the number of players affect strongly the performances of the P2P architecture. This is manly due to the huge number of packet transmitted within the network.

As packet delivery ratio, the gameplay is strongly affected by the end-to-end delay. Lower is the end-to-end delay, better is the gameplay. In order to evaluate the impact of the broadcast algorithms on the end-to-end delay, we plot, in figures 3 and 4, the average of maximum delay taken by each packet to reach the last destination node, for respectively C/S and P2P architectures. We mean by the last destination node, the last node in the network that received the message. In the case of client-server architecture we consider only the packets sent by the server, since it is the only one who send packets using broadcast. However, in the case of peer-to-peer architecture, we consider all the packets sent by all the nodes within the network. The obtained re-
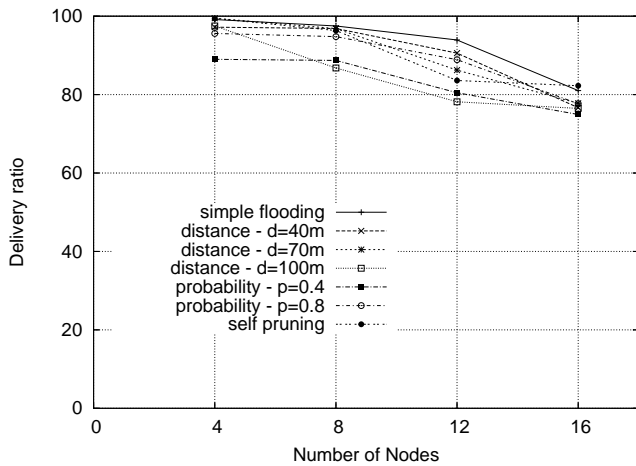
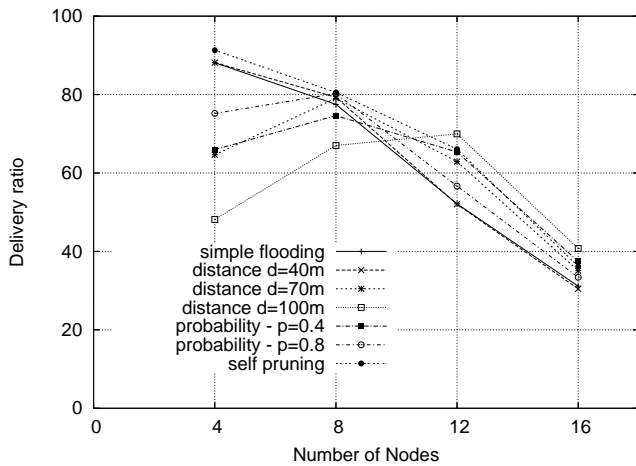Figure 1: Packet delivery ratio for client-server architecture

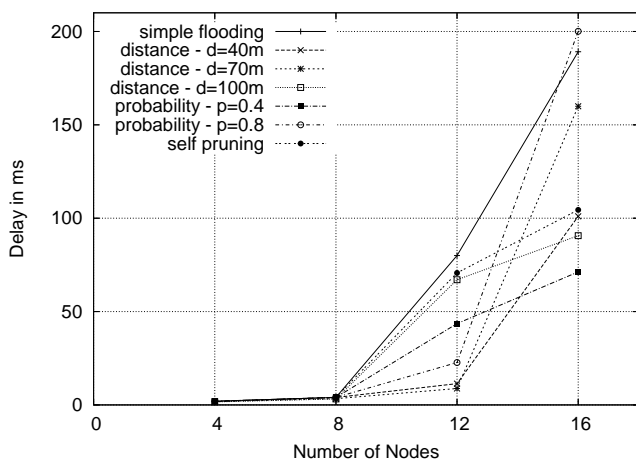Figure 2: Packet delivery ratio for peer-to-peer architecture

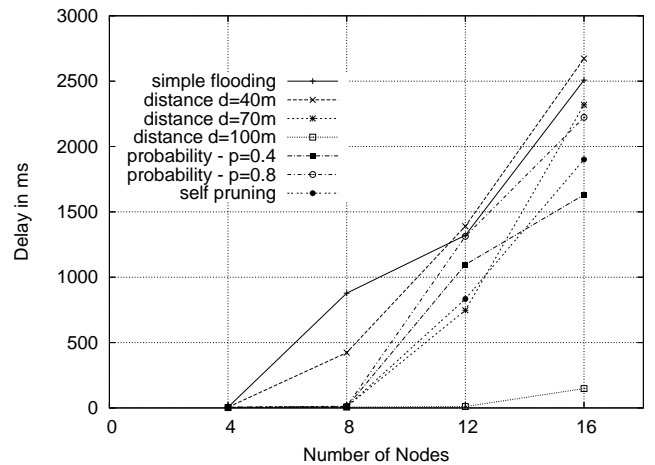Figure 3: Packet delay for client-server architecture

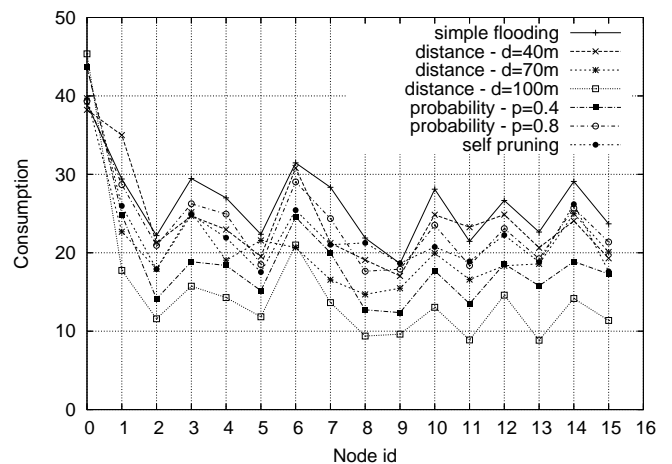Figure 4: Packet delay for peer-to-peer architecture

Figure 5: Energie consomption for client-server architecture

sults show that, in P2P architecture, except distance based scheme ($ford = 100\ m$) the average end-to-end delay is very high (greater than $1.5\ s$ for 16 players). However, the average end-to-end delay obtained for the C/S architecture (less than $200\ ms$ for 16 players). The main problem of P2P architecture is due to the huge number of re-broadcasted messages in the network, which encrease highly the end-to-end delay.

The battery life is one of the most serious issues for portable gaming consoles. Generally, default battery life varies widely depending on application. For example, the PlayStation Portable's lifetime varies from less than 3 hours while accessing a wireless network and having screen brightness on its highest setting to more than 10 hours during MP3 playback with the screen turned off. In order to evaluate the impact of broadcast algorithm on battery lifetime we plot, in figures 5 and 6, the consumed energy for each node, for respectively C/S and P2P architectures. We consider here a network game of 16 players. In our simulations we measured the energy consumption at the radio layer. According to the specification of IEEE 802.11-compliant WaveLAN-II [4] from Lu-
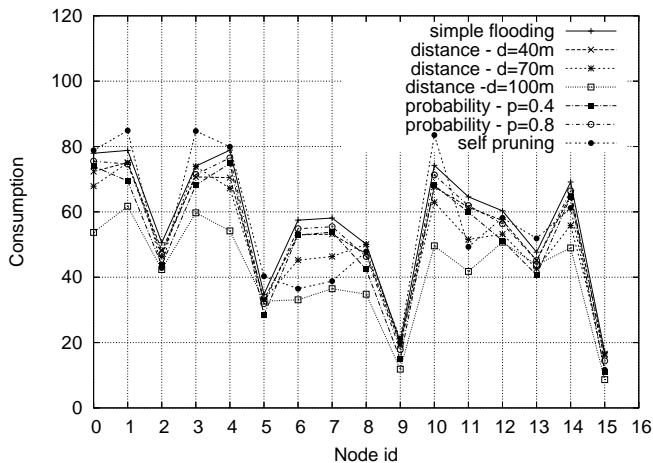
Figure 6: Energie consomption for peer-to-peer architecture

cent, the power consumption varies from 0.045 $Watts$ in sleep mode to 1.25 and 1.50 $Watts$ in receiving and transmitting modes, respectively. The instantaneous power is multiplied by time delay to obtain the energy consumed. For example, data transmission of a 127–byte packet consumes $0.762 \times 10-3 Joules$ ($1.50\ Watts \times 127 \times 8\ bits\ /\ 2000000 mps$). It is important to notice that in our simulation of the client-server architecture we add the impact of the packets transmission from the clients to the server. We can see clearly, from the two figures, that the average energy consumed in P2P architecture is greater than the average energy consumed in the C/S architecture. As montioned above, this is due mainly to the number of packet re-broadcated in P2P architecture.

6. CONCLUSION AND FUTURE WORKS

Multiplayer games will certainly support MANET mode as mobile devices and wireless communication technologies are becoming ubiquitous. However, this is a challenging task and several issues still have to be addressed to make such scenario a reality in the near future. In this paper we are dealing with MANET broadcast optimizations for multiplayer games since broadcast is a fundamental network service for such applications. We evaluate the performances of different broadcast schemes (distance-based, probability-based and self-pruning) for both Client/Server and P2P network game architectures. Results show that broadcasting under the assumption of a C/S architecture are globally satisfying. In particular for low distance-based and probability based parameters value. The drawback relies on the energy consumption that by the server node. Taking into consideration that this node will consume further mode energy since have also to compute the global information state of the game, one could reasonably expect that a better solution for load balancing the energy consumption relies on P2P architecture; However the results shows a high degradation of delays and energy consumption even for P2P (due the fact that more nodes are broadcasting). Exception made for distance based with a large d value (100). But, at the cost for less reliability. Of course, these "bad" results should be counter balanced by the fact that the CPU for comput-

ing the game state is not balanced over all nodes (contrarily to the C/S case, where it was concentrated on the server). Our experiments, clearly shows the need for designing an efficient and adapted broadcast MANET protocol that can scale with the constraints and specificities of the video game support. Our future work will concern effects of disconnections and the analysis of the network parameters that affect the quality of the game play. To that end, we will work on objective assessment of the gameplay

7. ACKNOWLEDMENTS

8. REFERENCES

[1] Simplified multicast forwarding for manet. IETF Internet-Draft. March 2007.

[2] Ietf manet wg (mobile ad hoc network). IETF, www.ietf.ora/html.charters/manet-charter.html.

[3] J. Färber. Network game traffic modelling. In *NetGames '02: Proceedings of the 1st workshop on Network and system support for games*, pages 53–57, New York, NY, USA, 2002. ACM Press.

[4] A. Kamerman and L. Monteban. Wavelan-ıı: A high-performance wireless lan for the unlicensed band. In *Bell Labs Technical Journal*, pages 118–133, Summer 1997.

[5] U. P. C. Laboratory and W. A. M. Laboratory. glomosim: A scalable simulation environment for wireless and wired network systems. In *Proceedings of the 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (Het-Net'05)*, 2002.

[6] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *MSWIM '00: Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 61–68, New York, NY, USA, 2000. ACM Press.

[7] W. Peng and X. Lu. Ahbp: An efficient broadcast protocol for mobile ad hoc networks. *J. Comput. Sci. Technol.*, 16(2):153–167, 114-125.

[8] W. Peng and X.-C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 129–130, Piscataway, NJ, USA, 2000. IEEE Press.

[9] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report Research Report RR-3898, INRIA, february 2000.

[10] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wirel. Netw.*, 8(2/3):153–167, 2002.

[11] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 194–205, New York, NY, USA, 2002. ACM Press.