



HAL
open science

Multipath OLSR: Simulation and Testbed

Jiazi Yi, Sylvain David, Hassiba Asmaa Adnane, Benoît Parrein, Xavier Lecourtier

► **To cite this version:**

Jiazi Yi, Sylvain David, Hassiba Asmaa Adnane, Benoît Parrein, Xavier Lecourtier. Multipath OLSR: Simulation and Testbed. 5th OLSR Interop/Workshop, Oct 2009, Vienna, Austria. hal-00424891

HAL Id: hal-00424891

<https://hal.science/hal-00424891v1>

Submitted on 19 Oct 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multipath OLSR: Simulation and Testbed

Jiazi YI, Sylvain DAVID, Asmaa ADNANE, Benoît PARREIN
Université de Nantes, Nantes Atlantique Universités
IRCCyN, CNRS UMR 6597, Polytech'Nantes, rue Christian Pauc - BP50609
44306 Nantes cedex 3 France
Email: {firstname.lastname}@univ-nantes.fr

Xavier LECOURTIER
Kéosys, 1 impasse Augustin Fresnel
44815 Saint Herblain
France
Email: xavier.lecourtier@keosys.com

Abstract—MP-OLSR is a multipath routing protocol based on OLSR (Optimized Link State Routing). The multipath routing protocol is expected to provide more stable routes for the network. In this paper, several topics about MP-OLSR are discussed. We begin with introducing the functionalities of MP-OLSR, which includes topology sensing, routing computation, route recovery and loop detection. Then a testbed is implemented to verify the availability of MP-OLSR. Given OLSR one of the most populated proactive protocols for ad hoc networks, the compatibility between MP-OLSR and OLSR is also discussed.

The results based on simulator and our testbed show that MP-OLSR could offer more stable data transmission over the unstable wireless interface. And it could cooperate well with the established OLSR protocol.

I. INTRODUCTION

Ad hoc networks are multi-hop wireless networks without any pre-installed infrastructure. To overcome the topology changes and the instability of the wireless medium, a lot of multi-path routing protocols are proposed [5], [9]. The simulation results of these protocols offer generally better performance compared with the single path protocols.

OLSR is a proactive protocol adapted to the mobility of the network. To improve the stability of the route, specially in the scenarios of high mobility and traffic load, we propose the multipath extension of OLSR to proved high aggregated bandwidth and load balancing. The algorithm could generate link-disjoint or node-disjoint paths set based on a modified version of Dijkstra Algorithm. Our previous work has proved that MP-OLSR is especially suitable for large dense networks with high-load traffic.

Most of results discussed in those multipath protocols are based on simulator. Compared with the numerous multipath protocols proposed, there are very few protocol realized as testbed in the literature. [10], [11] are based on MSR (multipath source routing), which is the multipath version of DSR. [12] is based on AOMDV, which is an extension of AODV.

In this paper, we will introduce MP-OLSR, a multipath extension of OLSR. A testbed is set up to test the performance of the multi-path routing. Furthermore, the compatibility between OLSR and MP-OLSR protocol is discussed. The remainder of the paper is organized as follows: In section II, we present the functionality of MP-OLSR protocol. Then the testbed for MP-OLSR is set up to test the protocol. Next, in section IV

the compatibility between OLSR and MP-OLSR is discussed. At the end we conclude the paper.

II. MULTIPATH OLSR - FUNCTIONALITY

The MP-OLSR can be regarded as a hybrid multipath routing protocol. It sends out *HELLO* messages and *TC* messages periodically to be aware of the network topology, just like OLSR. However, MP-OLSR does not always keep a routing table. It only computes the multiple routes when there are data packets need to be sent out.

A. Topology Sensing

The nodes use *Topology sensing* to get the topology information of the network. It includes link sensing, neighbor detection and topology discovery, as OLSR [1].

B. Route Computation

The route computation is based on the *Multipath Dijkstra Algorithm* [2]. The idea of the algorithm is to use the cost function to get mainly disjoint paths. The details are not presented here because of limitation of the space. For further information, please refer to [2].

C. Route Recovery

By using the scheme of the *Topology Sensing*, we can get the topology information of the network with the exchange of *HELLO* and *TC* messages. And all these information are saved in the topology information base of the local node. In the ideal case, the topology information base can be consistent with the real topology of the network. However, in reality, it is hard to achieve, mainly because the mobility of the ad hoc network.

To overcome the disadvantage of the source routing, we proposed *Route Recovery* for MP-OLSR. The policy is very simple: before an intermediate node trying to forward a packet to the next hop according to the source route, the node first check if the next hop in the source route is one of its neighbors (by checking the neighbor set). If yes, the packet is forwarded normally. If no, then it is possible that the “next hop” is not available anymore. Then the node will recompute the route and forward the packet by using the new route.

D. Loop Detection

In theory, the paths generated by the Dijkstra algorithm in MP-OLSR is loop-free. However, in reality, the LLN (Link Layer Notification) and *Route Recovery* which are employed to adapt to the topology changes make the loops possible in the network. This kind of abrupt interruption will result in additional operation on the topology information base rather than just regular HELLO and TC messages. It means that other nodes are not able to aware of these changes immediately. In [4], the author also addressed the looping issues in OLSRv2, and LLN will significantly increase the number of loops. So the author introduced two types of loop detection techniques: LD-Mid (Mid-Loop Detection) and LD-Post (Post-Loop Detection).

For MP-OLSR, we proposed a novel and simple method that can effectively detect loops without causing extra cost of memory: after *Route Recovery* was performed, we can get a new set of multiple paths from the current node to the destination. If the new path includes the node that the packet has passed before (by comparing with the packet source route), there is great possibility that a loop will happen. We will switch to the next path of the multiple paths set, until all the paths have been verified. If there is no suitable path, the packet will be discarded.

Our loop detection mechanism could effectively detect the possible loops in the network without consuming extra memory space. By reducing the loops in the network, the network congestion can be reduced. So the performance of the network can be improved, especially the end-to-end delay. In Section II-E, the simulation results will be presented to show the effect of the loop detection mechanism.

E. Simulation of Route Recovery and Loop Detection

In this part, the auxiliary functionalities are simulated to see the effects of the *Route Recovery* and *Loop Detection* on the performance of the network. We simulated an ad hoc network in Qualnet Simulator, with 81 nodes, moving in an area of 1500×1500 meters with different speed. The radio propagation range is about 270 meters.

Here, three different MP-OLSR protocols are compared:

- MP-OLSR without Route Recovery and Loop Detection,
- MP-OLSR with Route Recovery but without Loop Detection,
- MP-OLSR with Route Recovery and Loop Detection.

Figure 1 shows the comparison of the delivery ratio. The delivery ratio of the protocol without route recovery is very bad and very sensitive to the mobility of the nodes. The reason is that because of the delay of the transmission of the routing control packets (especially the TC message), the source node is very hard to get the most updated topology information when it is constructing the source route. This will result in that a node transmits a packet through a link that does not exist anymore. This phenomenon is more serious when the speed of the nodes increases. In fact, the SR-MPOLSR [5] also has very low delivery ratio as MP-OLSR without route recovery in our settings.

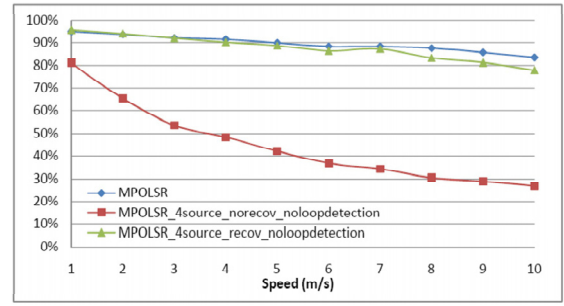


Fig. 1. Delivery Ratio with or without route recovery and loop detection

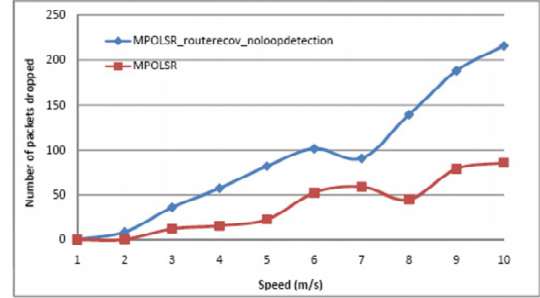


Fig. 2. Number of packets dropped because of the TTL comes to 0

From Figure 1 we can also find that the MP-OLSR protocol with loop detection has slightly better delivery ratio than the one without loop detection. This is because with the mechanism of the loop detection, there is more possibility to avoid the loop by switching to another path. When the loop detection is not applied, more following packets will be involved in the loop and be dropped because of the TTL (time-to-live) count to zero. Figure 2 presents the number of packets dropped because of the TTL comes to 0 (TTL is set to 64 in our case). Given the size of the simulation area (1500×1500) and the transmission range of the node (271m), the number of hops is less than 10 in most case. So when a packet is dropped because of TTL counts to 0, it can be regarded that a loop exists in our simulation.

In fact, compared with the effect on the delivery ratio, the loop detection has more influence on the average end-to-end delay. Figure 3 shows the delay of the different protocols. It is worth to mention that only the packets that successfully reached the destination are considered. So with or without route recovery does not affect the delay very much and gives similar results. However, the protocol without loop detection has much longer delay than the others because loops can easily congest the network.

III. TESTBED FOR MP-OLSR

This section presents our testbed for MP-OLSR. Different scenarios are proposed in order to verify the MP-OLSR protocol and compare with OLSR. The following test was realized in the *Ecole Polytechnique of University of Nantes*.

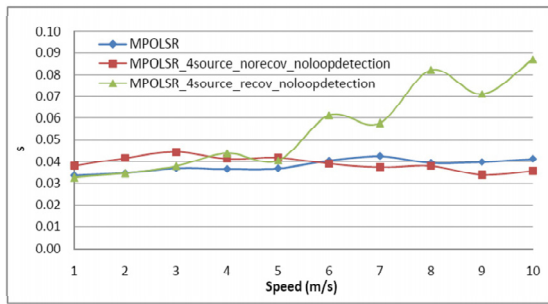


Fig. 3. Average end-to-end delay with or without route recovery and loop detection

Parameter	Value
CPU	Intel Atom N270
CPU Frequency	1600 MHz
Memory	DDR2 1024 MB
Radio Frequency	2.487GHz
Rate	54Mb/s with automatic bit-rate mode
Physical Layer	802.11g
Operating system	Linux (backtrack 3 live)
Kernel	Linux 2.6.21.5
OLSR version	Olsrd 0.5.6r2
Network protocol analyzer	Wireshark v0.91.6

TABLE I
EEEEPC CHARACTERISTICS

A. Hardware and Configuration

In our testbed, ASUS 901 EeePcs with the characteristics shown in table I are employed.

On the other hand, we use UFTP (UDP-based file transfer protocol [7]) as application layer protocol to test bi-directional exchanges.

B. Implementation of Multipath Routing

MP-OLSR is implemented based on an existing implementation of OLSR protocol (*Olsrd* [8]). Compared to the standard operations of OLSR protocol, the following modifications are made on the implementation provided by *Olsrd*:

- OLSR protocol calculates only one path between two nodes. Routes are updated when the network topology changes. Thus, it is necessary to change route calculation algorithm to take into account the specificities of MP-OLSR : multi-path and on-demand route calculation.
- Integration of a new function to receive paths requests sent by TCP/IP stack for the user data transmission.

As OLSR, after the reception of HELLO and TC messages, MP-OLSR updates the network topology information base. But no route is calculated. To send a user data, the IP layer sends a request to MP-OLSR to compute multiple routes to the destination (for the first request or when the topology changes) or to return existed routes in the multipath routing table(for the following requests).

It worth to mention that for the convenience of development and debug, the MP-OLSR module exists as an application layer program in our setting. This will result in frequent data

Protocol	MP-OLSR	OLSR
Duration of the transmission	6m 12s	5m17s (Connection lost)
Test duration	7min50s	6m26s
Packets sent by 10.0.0.100	15002	9784
Packets sent by 10.0.0.90	4503	5303
Packets sent by 10.0.0.99	3715	0 (route not used)
Packets sent by 10.0.0.95	2084	2909
Packets sent by 10.0.0.105	3726	0 (route not used)
Packets received by 10.0.0.98	13516	6112
Rate of lost packets	9.90%	37.53%

TABLE II

RESULTS OF SCENARIO 1, TEST1 : RATE = 62 KBYTES/S

exchanges between the kernel space and user space. For the purpose of efficiency in the future, it is better to put the multipath routing table in the Linux kernel as a kernel module after the test for the protocol is completed.

C. Results

In this subsection, two different scenarios are presented to test the functionality of MP-OLSR, and to compare the performance between multipath and single path protocols.

1) *Scenario 1, OLSR and MP-OLSR on 4 paths*: The first scenario presented includes 6 nodes. The location of the nodes is shown in figure 4. The number of paths for MP-OLSR is set to 4. Node with IP address 10.0.0.100 is chosen as source and 10.0.0.98 as destination. The distance from the source node to the destination node is about 60 meters. There are different kinds of obstacles in this scenario: trees, buildings and cars moving between the nodes. Iptable rules are employed to block the direct transmission between source and destination.



Fig. 4. Location of the nodes in scenario 1, in campus of Polytech'Nantes

In the following tests data rate is set to 62KBytes/s to transfer a file with 17.8 MBytes. Results are presented in table II. For MP-OLSR, the transmission finished in 6 minutes 12 seconds. Nodes with IP address 10.0.0.90, 10.0.0.95, 10.0.0.99 and 10.0.0.105 are chosen as intermediate nodes to relay the packets. During the transmission, 9.9% of packets are lost. For OLSR, only 10.0.0.90 and 10.0.0.95 participated in forwarding the data packets. The connection got lost after 5 minutes 17 secondes of transmission. For the packets sent out, 37.53% are lost.

To compare the network performance of these two protocols, we analysed the log file from *Wireshark*. Figure 5 and

figure 6 show the number of packets sent out to different nodes from the source (10.0.0.100) in each tick (1 second per tick).

As we can see from figure 5, for a fixed source rate, the traffic load is distributed in 4 paths. The transmission is almost continuous even some of the links are unavailable (the traffic will be assigned to other nodes in this case).

Compared with MP-OLSR, the transmission with OLSR (figure 6) was interrupted for a certain period because of a link is unavailable. In this case, the node will try to find another route to the destination. But the data transmission will be stopped during this period (for example, 80s-90s, 115s-130s in figure 6). And if the route switch takes too much time, the connection will be lost.

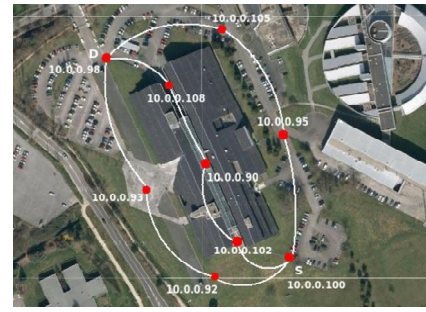


Fig. 7. Scenario 2: MP-OLSR routing with 3 paths

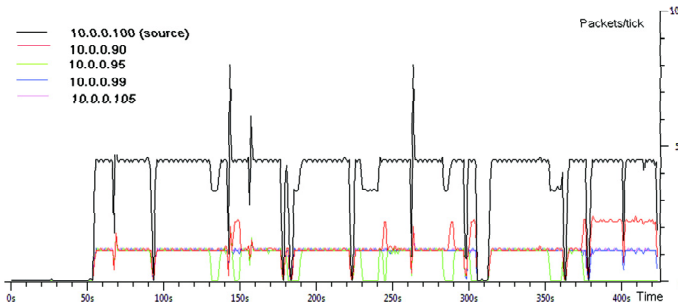


Fig. 5. Scenario 1 with MP-OLSR and rate=62KBytes/s

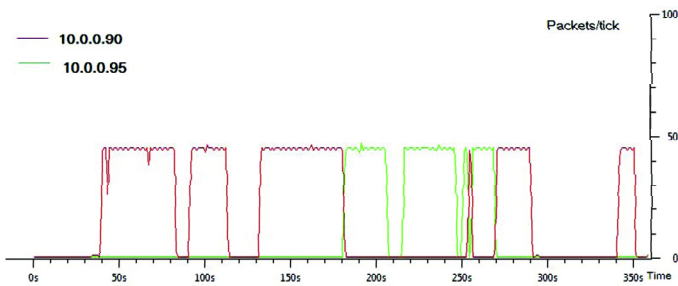


Fig. 6. Scenario 1 with OLSR and rate=62KBytes/s (Connection lost)

2) *Scenario 2, OLSR and MP-OLSR with 3 paths:* In the second scenario, we compared OLSR and MP-OLSR with 3 paths with are more than two hops away. The allocation of the nodes is shown in figure 7. The distance from the source to the destination is about 200m. There is a large building between them. To reach the destination, the packets have to travel around the building or go through the hall inside the building.

Table III presents the results obtained from this scenario. MP-OLSR finished the data transmission in 9 minutes 40 seconds, with a relatively high loss rate (37.05%). However, when using OLSR, the connection was lost after 8 minutes 43 seconds.

The tests with UFTP application presented in the section can prove that compared with single path protocol, the multipath protocol could provide more reliable routes because there is less possibility that the all the paths break in the same time.

Protocol	MP-OLSR	OLSR
Duration of the transmission	9m40s	8m43s(Connection lost)
Test duration	14m6s	9m6s
Packets sent by 10.0.0.100	14528	12548
Packets received by 10.0.0.98	9145	8544
Rate of lost packets	37,05%	31,90%(Connection lost)

TABLE III

RESULTS OF SCENARIO 2, OLSR AND MP-OLSR ROUTING WITH 3 PATHS

However, for OLSR, which just use single path, the data transmission will stop because of the path failure. If the route switch takes too much time, the connection will be lost.

IV. COMPATIBILITY BETWEEN OLSR AND MP-OLSR

As presented in the previous section, MP-OLSR is based on the OLSR protocol. Even if the multipath routing provides better performances than the single-path routing in most of the cases, studying the mutual compatibility between these two protocols is important. The backward compatibility makes it possible for MP-OLSR nodes work in an established OLSR network. On the other hand, it is important to keep a single-path routing because it could be more efficient in some scenarios (low mobility speed and sparse network).

To ensure the compatibility between the OLSR and MP-OLSR we propose an implementation of MP-OLSR protocol based on *IP-source routing*. By this way, OLSR nodes can forward data packets, generated by MP-OLSR nodes, according to the IP source routing. The intermediate nodes which run OLSR protocol will read the source routing option in the IP header, instead of consulting its routing table.

On the other hand, when MP-OLSR nodes receive data packets generated by OLSR nodes, they will behave as OLSR nodes by calculating the shortest path to the destination and forwarding the packets. The MP-OLSR nodes will recalculate the shortest path to the destination, and include the calculated path in the IP source routing option of the packet before forwarding. Thus, the packets will take advantage of the loop detection functionality implemented in MP-OLSR intermediate nodes. However, IP source routing option accepts only maximum of 9 addresses due to the limitation of the length of the IP header (20 bytes of fixed field and 40 bytes of optional field). Therefore, when the route contains more than 9 hops (large network), other solutions must be proposed.

The simplest solution is, if the route get by MP-OLSR is more than 9 hops away, it will just forward packet to the next hop, instead of using the source routing. The next hop will decide the rest of the route, no matter it is an MP-OLSR node or OLSR node. This solution is easy to implement but does not guarantee the multiple paths as defined by the source node. In the following simulation, results are based on this solution.

Another possible solution is using the *loose source routing*: source node just specify 9 “key” hops that the packet need to travel to get to the destination and allow each intermediate node to choose a route to the next hop. This solution maybe the optimal, because it guarantees the source routing as defined by the source node and does not require frequent updates. But it requires the MP-OLSR protocol to maintain an routing table just like OLSR.

The simulation based on Qualnet simulator is performed to study the compatibility between MP-OLSR protocol and OLSR protocol. The setting of the simulation is the same as section II-E.

We studied OLSR nodes sending packets to MP-OLSR nodes in the first place. In this scenario, there are 4 OLSR data sources in the network. We compared two scenarios in which the other 77 nodes are OLSR nodes, or MP-OLSR nodes. Figure 8 and figure 9 show the delivery ratio and the end-to-end delay of the simulation. So OLSR nodes have no problem in sending packets to MP-OLSR nodes. Furthermore, with the help of the *loop detection* of MP-OLSR, we can increase the packet delivery ratio and reduce the end-to-end delay of the network.

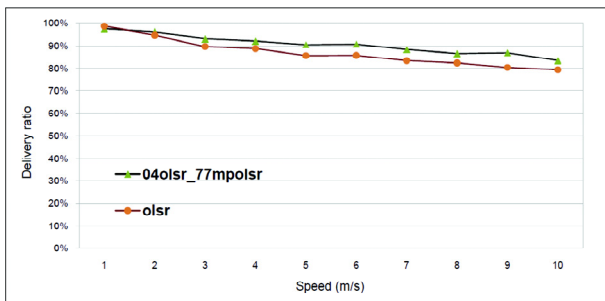


Fig. 8. Delivery ratio of the compatibility test, OLSR sources

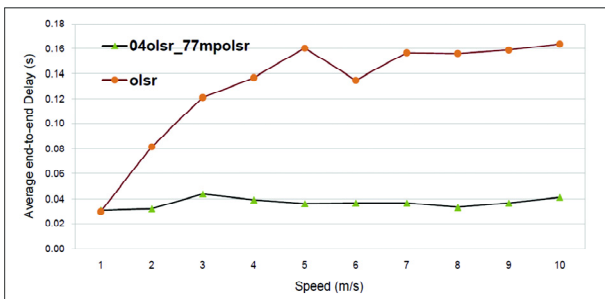


Fig. 9. Average end-to-end delay of the compatibility test, OLSR sources

In the rest of the simulation, there are 4 MP-OLSR sources

in the network. In these scenarios, we have different proportion of the OLSR nodes (marked as *04mpolsr_77olsr*, *20mpolsr_61olsr*, etc.). As we can see from the simulation results (figure 10), the existence of OLSR nodes has a negative affect on the delivery ratio. This is mainly because OLSR could not perform *route recovery* for the packets.

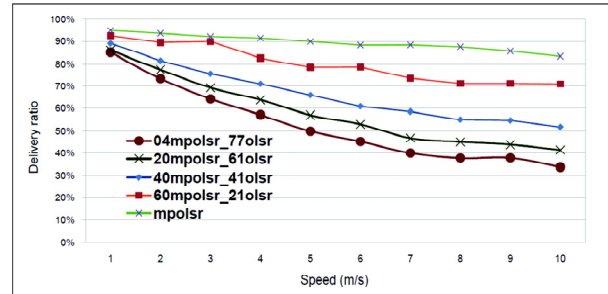


Fig. 10. Delivery ratio of the compatibility test, MP-OLSR sources

V. CONCLUSION

In this paper, we introduced the MP-OLSR with additional functionality such as *route recovery* and *loop detection*. The protocol is implemented as a multipath testbed. Our experiments show that multipath routing could offer more stable data transmission compared with the single path protocol. Given the distributed feature of the ad hoc networks and the popularity of OLSR, the compatibility between OLSR and its multipath version is also discussed.

Our future work includes improving the security of the protocol and exploiting the application of critical services video transmission based on MP-OLSR.

REFERENCES

- [1] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol OLSR, Request for Comments: 3626, IETF, Oct. 2003
- [2] J. Yi, E. Cizeron, S. Hamma, B. Parrein and P. Lesage. Implementation of Multipath and Multiple Description Coding in OLSR. Proceeding of the 4th OLSR Workshop, Ottawa, 2008.
- [3] nOLSRv2, Niigata OLSRv2 Implementation.
- [4] L. Speakman, Y. Owada, K. Mase, An Analysis of Loop Formation in OLSRv2 in Ad-hoc Networks and Limiting its Negative Impact, IEEE Communications Quality and Reliability Workshop, 2008.
- [5] X. Zhou, Y. Lu and B. Xi, A novel routing protocol for ad hoc sensor networks using multiple disjoint paths, 2nd International Conference on Broadband Networks, 2005.
- [6] Yasunori Owada, Taka Maeno, Hiroei Imai and Kenichi Mase. OLSRv2 implementation and performance evaluation with link layer feedback. Proceedings of the 2007 ACM international conference on Wireless communications and mobile computing, 2007.
- [7] UDP Based FTP with multicast, <http://www.tcnj.edu/~bush/uftp.html>
- [8] OLSRd, an adhoc wireless mesh routing daemon, <http://www.olsr.org>
- [9] Lee, S. and Gerla, M., Split multipath routing with maximally disjoint paths in ad hoc networks, Proceedings of the IEEE ICC, 2001.
- [10] Shiwen Mao, Shunan Lin, Panwar, S.S., Yao Wang. A multipath video streaming testbed for ad hoc networks. Proceeding of IEEE VTC, 2003.
- [11] Yihua Zhai, Yang, O.W.W., Wenlan Wang and Yantai Shu. Implementing multipath source routing in a wireless ad hoc network testbed. Proceeding of IEEE PACRIM, 2005.
- [12] K. TANIYAMA, T. MORII, S. KOIZUMI et al. Experimental evaluation of an on-demand multipath routing protocol for video transmission in mobile ad hoc networks. Zhejiang Univ SCIENCE A 2006 7(Suppl. D):145-150