Journal of Mobile Multimedia, Vol. 8, No. 3 (2012) 204–224

END-TO-END SINGLE AND MULTIPLE FLOWS FAIRNESS IN MOBILE AD-HOC NETWORKS

MAKOTO IKEDA
Department of Information and Communication Engineering
Fukuoka Institute of Technology (FIT)
3-30-1, Wajiro-higashi, Higashi-ku, Fukuoka 811-0295, Japan
m-ikeda@fit.ac.jp

Received November 1, 2011
Revised May 23, 2012

Wireless networks have become increasingly popular in recent years. They can provide mobile users with ubiquitous communication capability and information access regardless of locations. In this paper, we evaluate the heterogeneous environments of wireless mobile ad-hoc networks (MANETs) by simulations considering simulation runtime, memory usage, (average) throughput, percentage of received constant bit rate packets and congestion window. For simulations, we used ns-2 and ns-3 network simulators considering Ad hoc On-demand Distance Vector (AODV) and Optimized Link State Routing (OLSR) routing protocols. We evaluate the MANET performance considering random waypoint mobility models for different number of nodes, different area sizes and different maximum speed of mobility, by sending single and multiple flows in the network. The evaluation results show that OLSR had more disconnections than AODV and for large-size area, there is no fairness between flows. We found that TCP flow can have better connectivity than UDP flow in case of single flow data. While for dual flow, TCP performance is better in the beginning phases of transmission, because of the slow start mechanism.

Keywords: MANET, Multiple Flows, AODV, OLSR, ns-2, ns-3.
Communicated by: D. Taniar

1 Introduction

Mobility and the absence of any fixed infrastructure make Mobile Ad-hoc Networks (MANETs) very attractive for time-critical applications. There are a lot of issues and challenges in designing a MANET network. At transport layer, end-systems can gather information about each used path: congestion state, capacity and latency. These information can then be used to react to congestion events in the network by moving the traffic away from congested paths.

Most of the work for MANET has been done in simulation, as in general, a simulator can give a quick and inexpensive understanding of protocols and algorithms [1]. So far, there are many simulation results on the performance of MANET, e.g. in terms of end-to-end throughput, round trip time and packet loss.

In our previous work [2, 3, 4], we carried out experiments and simulations using network simulator version 2 (ns-2). We proved that while some of the Optimized Link State Routing (OLSR)'s problems can be solved, for instance the routing loop and self-interference problem.

There is an intricate inter-dependence between MAC layer and routing layer, which can lead the experimenter to misunderstand the results of the experiments. For example, the horizon is not caused only by IEEE 802.11 Distributed Coordination Function (DCF), but also by the routing protocol.

In this work, we investigate by simulations the performance of a MANET for different size environment (100m×100m and 200m×200m) considering Ad hoc On-demand Distance Vector (AODV) and OLSR routing protocols. First, we compare the simulation runtime and memory usage of ns-2 and network simulator version 3 (ns-3) simulators. Then, we compare the (average) throughput, percentage of received Constant Bit Rate (CBR) packets and congestion window (CWND) for different area sizes, different number of nodes and different Maximum Speed (MS) of mobility. In these evaluation, we deal with congestion control for multiple flows traffic in MANET.

The structure of the paper is as follows. In Section 2, we discuss the related work. In Section 3, we describe the ns-2 and ns-3 simulators and some radio propagation models. In Section 4, we make an overview of MANET routing protocols. In Section 5, we describe the simulation system design and description. In Section 6, we show the simulation results. Finally, conclusions and future work are given in Section 7.

2 Related Work

Until now, many researchers performed valuable research in the area of multi-hop wireless networks by computer simulations and experiments [5]. Most of them are focused on throughput improvement, but they do not consider mobility [6].

In [7, 8], the authors propose a dynamic probabilistic broadcasting scheme for MANETs. In this scheme, the nodes move according to different mobility models. Simulation results show that their approach outperforms the FP-AODV and simple AODV in terms of saved rebroadcast under different mobility models. It also achieves higher saved rebroadcast and low collision as well as low number of relays than the fixed probabilistic scheme and simple AODV.

In [9], the disadvantage of using hysteresis routing metric is presented through simulation and indoor measurements.

In [10], the authors compare the performance of a wired network by implementing an identical simulation set-up in five simulators, ns-2, OMNet++ [11], ns-3 [12], SimPy [13] and JiST/SWANS. The authors focused on simulation runtime and memory usage, but they only considered wired networks.

In our previous work [4], we carried out experiments and simulations in indoor environment using our implemented testbed, ns-2 and ns-3. We used OLSR protocol for performance evaluation. For comparison, we considered two models: linear and grid topologies and as metrics the throughput. From our evaluation, we found that the throughput simulation results with ns-3 are similar to results from testbed experiments and the network has a good performance.

In addition, we evaluated average throughput, average packet received rate, percentage of received CBR packets and received routing packets metrics of a MANET, comparing AODV, Destination-Sequenced Distance-Vector (DSDV) and OLSR protocols, using ns-3 simulator [14]. We made simulations for different number of nodes in the network and for different Area

Sizes (AS=100m×100m and 250m×250m).

3 Network Simulators

3.1 NS-2

The ns-2 is a object oriented simulator, written in *C++*, with an OTcl interpreter as a front-end. It uses two languages because simulator has two different kind of things it needs to do. On one hand, detailed simulations of protocols require a system programming language, which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. In ns-2, the front-end of the program is written in tool command language. The back-end of ns-2 simulator is written in *C++* and when the tool command language program is compiled, a trace file and *nam*[a] file are created which define the movement pattern of the nodes and keeps track of the number of packets sent, number of hops between two nodes and connection type at each instance of time.

3.1.1 802.11b Channel

There are three basic models for the propagation of the radio signals of mobile nodes: free space, two-ray ground reflection, and shadowing radio models. In a simple deterministic model, the received power P_r at a certain distance d is the same along all directions in the plane. For example, in case of line of sight propagation of the signal, the Friis formula predicts the received power as:

$$P_r(d) = P_t - \beta \quad (\text{dB}) \tag{1}$$
$$\beta = 10 \log \left(\frac{(4\pi d)^2 L}{G_t G_r \lambda^2} \right)$$

where G_r and G_t are the antenna gains of the receiver and the transmitter, respectively, λ is the wavelength of the signal and L the insertion loss caused by feeding circuitry of the antenna, and β is the propagation path loss exponent. For omni-antennas, $G_R = G_t = 1$. The signal decay is then proportional to d^2.

In our simulation system, we are using a more accurate model called Shadowing radio model to simulate one 802.11b channel. Shadowing model simulates shadow effect between the transmitter and receiver. It mainly is used to simulate wireless channel in indoor environment.

3.1.2 Shadowing Radio Model

The shadowing model consists of two parts. The first one is known as path loss model, which also predicts the mean received power at distance d, denoted by $\overline{P_r(d)}$. It uses a close-in distance d_0 as a reference. $\overline{P_r(d)}$ is computed relative to $P_r(d_0)$ as follows.

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = \left(\frac{d}{d_0} \right)^{\beta} \tag{2}$$

β is called the path loss exponent, and is usually empirically determined by field measurement. The path loss is usually measured in dB. So, from Eq. (2) we have:

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)} \right]_{dB} = -10\beta \log \left(\frac{d}{d_0} \right). \tag{3}$$

[a]Nam is network visualization tool of ns-2.

The second part of the shadowing model reflects the variation of the received power at certain distance. It is a log-normal random variable, that is, it is a Gaussian distribution and is measured in dB. The overall shadowing model is represented by:

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \tag{4}$$

where X_{dB} is a gaussian random variable with zero mean and standard deviation σ_{dB}. Eq. (4) is also known as a log-normal shadowing model. The shadowing model extends the ideal circle model to a richer statistic model: nodes can only probabilistically communicate when they are near the edge of the communication range. We set the values of the path loss exponent β to 4.0, shadowing deviation σ_{dB} is 9.6.

3.2 NS-3

The ns-3 simulator is developed and distributed completely in the C++ programming language, because it better facilitated the inclusion of C-based implementation code. The ns-3 architecture is similar to Linux computers, with internal interface and application interfaces such as network interfaces, device drivers and sockets. The goals of ns-3 are set very high: to create a new network simulator aligned with modern research needs and develop it in an open source community. Users of ns-3 are free to write their simulation scripts as either *C++ main()* programs or *Python* programs. The ns-3's low-level API is oriented towards the power-user but more accessible "helper" APIs are overlaid on top of the low-level API.

In order to achieve scalability of a very large number of simulated network elements, the ns-3 simulation tools also support distributed simulation. The ns-3 support standardized output formats for trace data, such as the pcap format used by network packet analysing tools such as tcpdump, and a standardized input format such as importing mobility trace files from ns-2.

The ns-3 simulator is equipped with *Pyviz* visualizer, which has been integrated into mainline ns-3, starting with version 3.10. It can be most useful for debugging purposes, i.e. to figure out if mobility models are what you expect, where packets are being dropped. It is mostly written in Python, it works both with Python and pure C++ simulations. The function of ns-3 visualizer is more powerful than network animator (*nam*) of ns-2 simulator.

The ns-3 simulator has models for all network elements that comprise a computer network. For example, network devices represent the physical device that connects a node to the communication channel. This might be a simple Ethernet network interface card, or a more complex wireless IEEE 802.11 device.

The ns-3 is intended as an eventual replacement for popular ns-2 simulator. The ns-3's wifi models a wireless network interface controller based on the IEEE 802.11 standard [15]. Ns-3 provides models for these aspects of 802.11:

(a) Basic 802.11 DCF with infrastructure and ad hoc modes.
(b) 802.11a, 802.11b, 802.11g and 802.11s physical layers.
(c) QoS-based EDCA and queueing extensions of 802.11e.
(d) Various propagation loss models including Nakagami, Rayleigh, Friie, LogDistance, FixedRss and Random.
(e) Two propagation delay models, a distance-based and random model.

(f) Various rate control algorithms including Aarf, Arf, Cara, Onoe, Rraa, ConstantRate, and Minstrel.

3.2.1 Log-distance Path Loss Model

The log-distance path loss model is a radio propagation model that predicts the path loss a signal encounters inside a building or densely populated areas over distance. This propagation model is applicable for indoor propagation modeling. Log-distance propagation loss model is formally expressed as:

$$L = L_0 + 10nlog_{10}(\frac{d}{d_0}) \tag{5}$$

where:

- n: the path loss distance exponent,
- d_0: reference distance [m],
- L_0: path loss at reference distance [dB] ,
- d: distance [m],
- L: path loss [dB].

When the path loss is requested at a distance smaller than the reference distance, the value of Tx power is returned.

3.2.2 Random Waypoint Mobility Model

The random waypoint mobility model is commonly used in most simulations. In this model, the nodes move from one waypoint to another independently from each other. For every node, a random waypoint and a random speed is chosen and the node starts moving toward the chosen waypoint with the chosen speed. After that the node randomly chooses a pause time and after pause time, it starts to move to the next randomly chosen waypoint. In random waypoint mobility model, all nodes are considered mobile.

4 Overview of MANET Routing Protocols

4.1 AODV Protocol

AODV is a combination of both Dynamic Source Routing (DSR) and DSDV protocols. It has the basic route discovery and route maintenance of DSR and uses the hop by hop routing, sequence numbers and beacons of DSDV. The node that wants to know a route to a given destination generates a Route Request (RREQ). The RREQ is forwarded by intermediate nodes. It also creates a reverse route for itself from the destination. When the request reaches a node with route to destination it generates a Route Reply (RREP) containing the number of hops requires to reach destination. All nodes, that participate in forwarding this reply to the source node, create a forward route to destination. This state created from each node (from source to destination) is a hop by hop state and not the entire route is done in source routing.

4.2 OLSR Protocol

The link state routing protocol that is most popular today in the open source world is OLSR. OLSR with Link Quality (LQ) extension and fisheye-algorithm works quite well. The OLSR protocol is a pro-active routing protocol, which builds up a route for data transmission by maintaining a routing table inside every node of the network. The routing table is computed upon the knowledge of topology information, which is exchanged by means of Topology Control (TC) packets. The TC packets in turn are built after every node has filled its neighbors list. This list contains the identity of neighbor nodes. A node is considered a neighbor if and only if it can be reached via a bi-directional link.

OLSR makes use of HELLO messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its Multi Point Relays (MPRs) based on the one hop node which offer the best routes to the two hop nodes. By this way, the amount of control traffic can be reduced. Each node has also an MPR selector set which enumerates nodes that have selected it as an MPR node.

In short, every node computes the path towards a destination by means of a simple shortest-path algorithm, with hop-count as target metric. In this way, a shortest path can result to be also not good, from the point of view of the packet error rate.

5 Simulation Design and Description

We have implemented a wireless mobile ad-hoc network simulation system which is a platform for analyzing various aspect of these networks, including stationary or mobility models, propagation loss models and propagation delay models.

5.1 Evaluation of Simulation Systems

To evaluate the simulation systems, we make the following considerations.

- We consider an indoor environment with different number of nodes.
- We investigate the simulation runtime and number of received packets and hop distance versus grid sizes.
- We constructed a simulation model as Grid Topology (GT), which is shown in Fig. 1.

For the performance evaluation with ns-2 and ns-3 simulators, we set-up the GT. We use OLSR as a routing protocol. We measured the simulation runtime during data transmission. The distance of nodes from each-other is 100 meters. For this evaluation, we use the $N \times N$ topology, where N is the grid-size, and $N = \{5, 6, \ldots, 15\}$. Thus, the network size (N^2) varies from 25 to 225 nodes. The hardware specification are shown in Table 1. The simulation parameters are shown in Table 2.

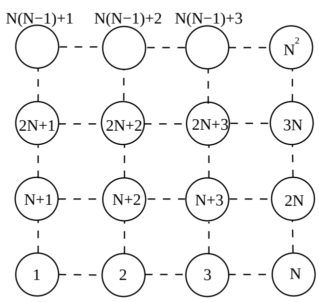


Fig. 1. Grid topology.

Table 1. Hardware information

Function	Value
CPU	Intel C2D E7300 2.66GHz
Memory	2GBytes
HDD: Capacity	160GBytes
HDD: Rotation	7200RPM
HDD: Buffer	8MBytes
HDD: Interface	Serial ATA300
Partition Type	Ext4 (version 1.0)
OS	Ubuntu 10.04
Kernel	2.6.32-31

Table 2. Common parameters for both network simulators to evaluate the simulation systems.

Parameter	Value
Topology Size	$N \times N$ (N is from 5 to 15)
Grid Distance	100 m
Number of Nodes	N^2
Flow Type	HTTP
Source Node ID	$\#N^2$
Destination Node ID	#1
Packet Rate	20 pkt/sec
Duration	10 sec
Number of packets	200
Packet Size	1088 bytes
Routing Protocol	OLSR

5.2 Evaluation of Multiple Flows Communications Considering UDP

We use AODV and OLSR routing protocols. The source-destination pairs are fixed over the network. We send CBR data from source to the destination with a theoretical throughput of 2Mbps for each traffic. CBR Interval ($\mathrm{CBR_{INTERVAL}}$) is define as shown in Eq. (6).

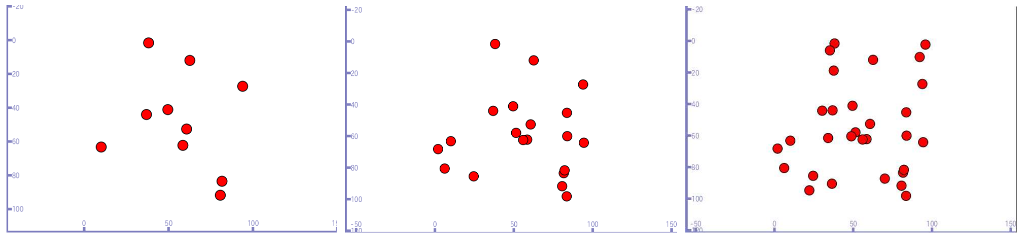
$$\mathrm{CBR_{INTERVAL}} = \frac{\text{Packet size [byte]} \times 8}{\text{Theoretical throughput [bps]}} \tag{6}$$

Simulation parameters are shown in Table 3. For our simulations, we make the following considerations.

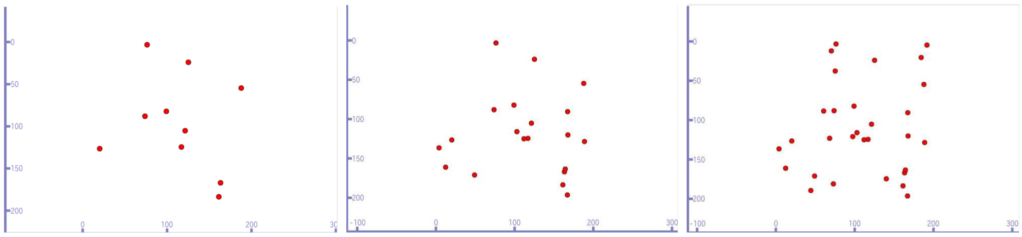
- We consider a small size environment with different number of nodes: 10, 20 and 30 nodes, respectively.
- We investigate the average throughput of multi-hop wireless network and percentage of received CBR packets ($\mathrm{P_{CBR}}$).
- Nodes on the network move according to random waypoint mobility model in a rectangular field and their initial position are shown in Fig. 2.
- Each node moves from a random location to a random destination with a randomly chosen speed uniformly distributed between 1-6 m/s or 1-12 m/s.
- Simulation time is 150 seconds.

Table 3. Simulation parameters for evaluating multiple flows (UDP).

Parameter	Value
Mobility model	Random Waypoint
Min and Max node speed	1.0 m/s and 12.0m/s
Node pause	3.0 sec
Constant Bit Rate Traffic	200 pkt/sec
Duration	10 or 150 sec
Number of packets	20000 pkts
Packet Size	512 bytes
Propagation Loss Model	Log-distance Path Loss Model
Propagation Delay Model	Constant Speed Model



(a) 10 nodes in AS=100m×100m (b) 20 nodes in AS=100m×100m (c) 30 nodes in AS=100m×100m



(d) 10 nodes in AS=200m×200m (e) 20 nodes in AS=200m×200m (f) 30 nodes in AS=200m×200m

Fig. 2. Initial position of nodes in each area sizes.

5.3 Evaluation of Multiple Flows Communications Considering UDP and/or TCP

We have implemented a MANET simulation system including mobility models, propagation loss models and propagation delay models. This system is a platform for analyzing various aspect of these networks. We use OLSR routing protocol. The source-destination pairs are fixed over the network. We send CBR flow transported over UDP and/or TCP data from source to the destination with a theoretical throughput of 2.5Mbps for each traffic. CBR Interval ($CBR_{INTERVAL}$) is define as shown in Eq. (6). Simulation parameters are shown in Table 4. For our simulations, we make the following considerations.

- We consider a small size environment with different number of nodes: 10, 20 and 30 nodes, respectively.
- We investigate the throughput of multi-hop wireless network and CWND.

Table 4. Simulation parameters for evaluating multiple flows (UDP and/or TCP).

Parameters	Value
Area Size	100m×100m or 200m×200m
MAC	IEEE 802.11b
Wireless Interface Mode	Ad-hoc
Propagation Loss Model	Log-distance Path Loss Model
Propagation Delay Model	Constant Speed Model
Mobility Model	Random Waypoint
Number of Nodes	10, 20 and 30
Min Node Speed	1.0 m/s
Max Node Speed	6.0 m/s or 12.0 m/s
Node Pause Time	3.0 sec
Number of Flows	1 or 2
Source Node ID	#7 (and #10, multiple flow)
Destination Node ID	#1

Table 5. Flow setting.

Case ID	Number of flows	Flow type	
UDP	1 (#7→#1)	UDP (Flow1)	
TCP	1 (#7→#1)	TCP (Flow1)	
UDPxTCP	2 (#7→#1, #10→#1)	UDP (Flow1)	TCP (Flow2)
Dual TCP	2 (#7→#1, #10→#1)	TCP (Flow1)	TCP (Flow2)

- We evaluate four kind of flows as shown in Table 5. We study the effect of fairness for four cases: UDP, TCP, UDPxTCP and Dual TCP.
- Nodes on the network move according to random waypoint mobility model in a rectangular field and their initial positions are shown in Fig. 2.
- Each node moves from a random location to a random destination with a randomly chosen speed uniformly distributed between 1 m/s and 12 m/s.
- Simulation time is 150 seconds. Each flow starts at 50 seconds and stops at 150 seconds at the source node.

6 Simulation Results

6.1 Results for Simulation System

In Fig. 3 are shown the simulation results of ns-2 and ns-3 for simulation runtime versus grid size. In this evaluation, we use the GT. In ns-2 environment, we set the GT manually. For ns-3, we use the *ns3::GridPositionAllocator* class to create the GT. By ns-3, it is possible to set-up the grid position for large scale network simulations. The ns-3 is also equipped with command arguments to set-up the number of nodes, position of nodes, source node, destination node and so on. From Fig. 3, we can observe that the time for executing the simulation in ns-2 is bigger than that of ns-3. The simulation runtime of ns-2 seems to be increased exponentially, while for ns-3 it is almost linear.

In Fig. 4 are shown the simulation results for number of packets received and average hop

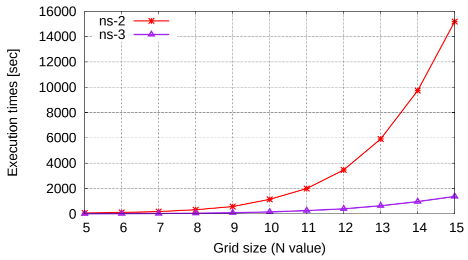


Fig. 3. Runtime results of ns-2 and ns-3.

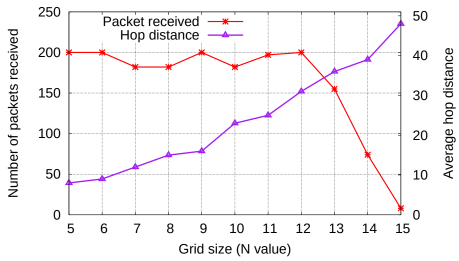


Fig. 4. Number of received packets and hop distance vs. grid size.

distance versus grid size. The average hop distance is increasing in almost linear progression when grid size increases. The number of received packets is affected by the hop distance, because of the TTL value reaching 0 and consequently the packet being dropped. As shown in Fig. 4, when grid size increases more than 12, the hop distance reaches values more than 32. Thus some packets start to drop and the number of received packets starts to decrease. When the grid size is less than 12, the number of received packets is stable and almost the maximum.

6.2 Results for Multiple Flows Considering UDP

For performance evaluation, we use two metrics: average throughput and percentage of received CBR packets (P_{CBR}), which is defined as shown in Eq. (7).

$$P_{CBR} = \frac{\text{Number of received CBR packets [pkts]}}{\text{Number of total received packets [pkts]}} \times 100 \qquad (7)$$

"Number of total received packets" shows the total number of packets received, including control and routing packets. The results are shown from Table 6 to Table 9.

Regarding AODV (see Table 6 and Table 7), for small-size area with maximum speed set to 6 m/s, the performance is almost the same for different number of nodes. When the maximum speed is set to 12 m/s, the performance is increased when number of nodes increases. The best value of average throughput for single flow is around 22% lower than theoretical throughput, and for multiple flow is around 32% lower.

While for large-size area when the maximum speed is 6 m/s, the performance is increased when number of nodes increases. On the other hand, when the maximum speed is 12 m/s, the performance is better when number of nodes is 20, but we can notice that both throughput and P_{CBR} are much more lower than the case of 100m×100m area. In the case of large-size area the route discovery process becomes difficult, because there are more disconnections.

For OLSR (see Table 8 and Table 9), for small-size area with maximum speed set to 6 m/s, the average throughput is decreased when number of nodes is 30. Compared with cases of 10 and 20 nodes, in the case of 30 nodes, the throughput decreases about 1.5Mbps and 3Mbps, for single and multiple flows, respectively. When maximum speed is 12 m/s, the throughput is higher when the number of nodes is 10.

For large-size area with multiple flow, the throughput of OLSR is higher than that of AODV, when the number of nodes is 10. However, when the number of nodes is 20 or 30,

Table 6. Simulation results of AODV for single flow.

NumNodes	Area size: 100m×100m				Area size: 200m×200m			
	Max Speed 6 m/s		Max Speed 12 m/s		Max Speed 6 m/s		Max Speed 12 m/s	
	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]
10	45.91	2.352	47.70	2.042	42.93	1.452	38.61	1.206
20	48.75	2.346	48.86	2.346	43.32	1.591	45.99	1.982
30	48.25	2.346	48.12	2.346	49.13	2.372	38.92	1.767

Table 7. Simulation results of AODV for multiple flows.

NumNodes	Area size: 100m×100m				Area size: 200m×200m			
	Max Speed 6 m/s		Max Speed 12 m/s		Max Speed 6 m/s		Max Speed 12 m/s	
	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]
10	48.44	3.605	49.21	3.794	45.06	1.562	42.71	1.280
20	49.27	4.086	49.16	3.861	47.53	2.996	47.22	2.889
30	46.96	3.690	48.97	4.082	46.68	2.774	44.31	2.577

Table 8. Simulation results of OLSR for single flow.

NumNodes	Area size: 100m×100m				Area size: 200m×200m			
	Max Speed 6 m/s		Max Speed 12 m/s		Max Speed 6 m/s		Max Speed 12 m/s	
	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]
10	49.66	2.063	49.71	2.304	35.41	0.734	41.72	0.459
20	49.34	2.346	49.17	1.977	32.95	1.002	49.26	1.274
30	35.51	0.881	48.12	2.217	35.60	0.612	32.38	0.452

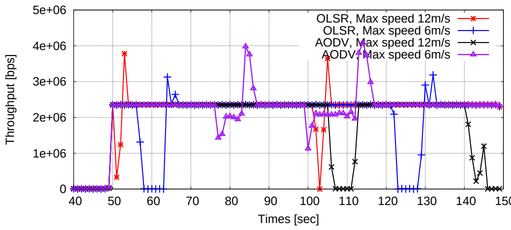
Table 9. Simulation results of OLSR for multiple flows.

NumNodes	Area size: 100m×100m				Area size: 200m×200m			
	Max Speed 6 m/s		Max Speed 12 m/s		Max Speed 6 m/s		Max Speed 12 m/s	
	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]	P_{CBR} [%]	Avg. Thr. [Mbps]
10	50.04	3.875	49.84	4.028	46.18	1.760	48.81	2.178
20	49.64	4.167	49.59	3.773	34.49	1.190	47.69	2.073
30	38.17	0.957	48.83	3.476	41.89	1.367	44.18	1.836

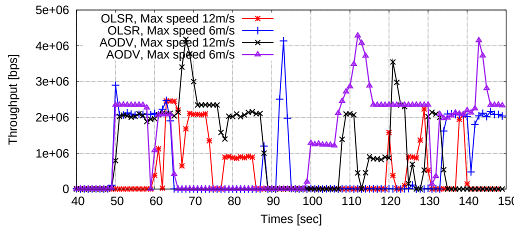
AODV has higher throughput than OLSR. In these cases, OLSR uses more control packets to keep the topology updated. This is also shown by the decrease of P_{CBR}.

In Fig. 5, we show the throughput of small-size area, for both AODV and OLSR protocols. We analyze the performance of the network by observing the disconnection time between source and destination. For AODV protocol, there are disconnections only when number of nodes is 10 (see Fig. 5(a)). While OLSR has many disconnections during simulation time for different number of nodes (see also in Fig. 5(c) and Fig. 5(e). In this case, we can say that AODV has better performance than OLSR.

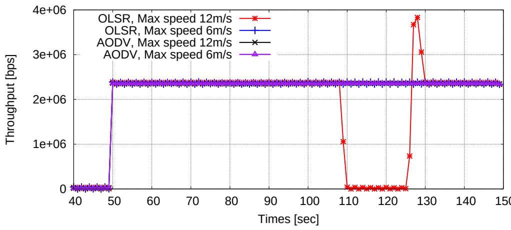
While for 200m×200m size area, when number of nodes is 10 (see Fig. 5(b)), OLSR shows more disconnections than AODV. In Fig. 5(d), where the number of nodes is 20, there are less disconnections, for all cases. However, OLSR has more disconnections than AODV. As the number of nodes increases, the simulation area is covered by more nodes so the cases of disconnection decrease. As we can see in Fig. 5(f), disconnection time is shorter than in



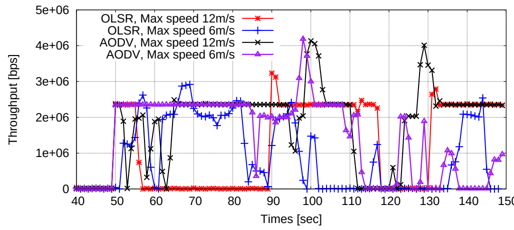
(a) Nodes=10, AS=100m×100m



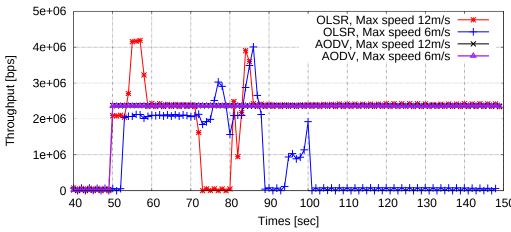
(b) Nodes=10, AS=200m×200m



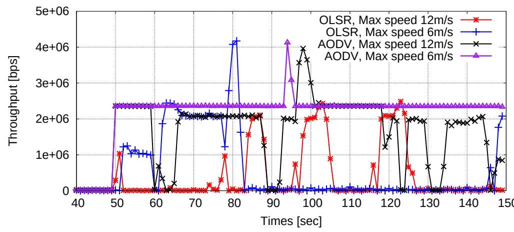
(c) Nodes=20, AS=100m×100m



(d) Nodes=20, AS=200m×200m



(e) Nodes=30, AS=100m×100m



(f) Nodes=30, AS=200m×200m

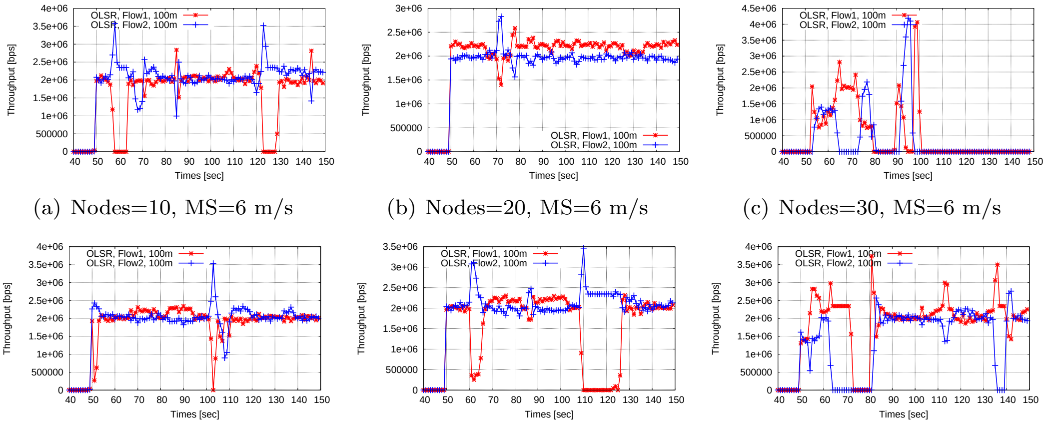
Fig. 5. Throughput results with single flow.

previous cases. Regarding the maximum speed of movement, for different number of nodes (10, 20 and 30), when speed is faster the performance decreases.

In Figs. 6 to 9, we show the results in the case of multiple flows transmission. We evaluate the performance based on disconnections between source and destination and on the fairness between the two flows sent from sources to destination.

First, we would like to discuss the results of the small-size area (100m×100m). If we compare Fig. 6 with Fig. 8, respectively, OLSR has better performance than AODV for 10 nodes, the same performance for 20 nodes and worse performance than AODV for 30 nodes, for both maximum speed set to 6 m/s and 12 m/s. Compared to the case when the area size is 100m×100m, the performance comparison between AODV and OLSR for area size 200m×200m is different (see Fig. 7 and Fig. 9). For 10 nodes, OLSR has better performance than AODV, and for 20 and 30 nodes AODV has better performance than OLSR.

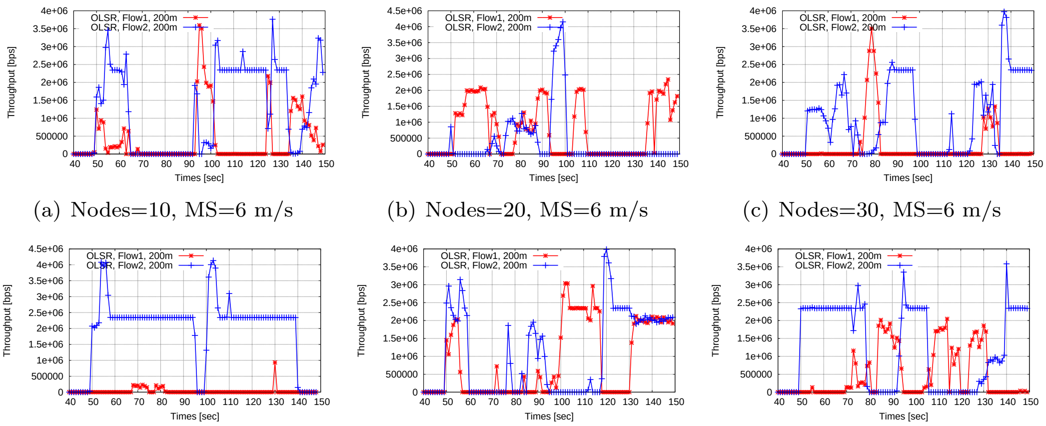
If we compare the two different area size in terms of fairness between two flows sent in the network, we can see that in the small area (see Fig. 6 and Fig. 8), the flows are fair. While for area size 200m×200m (see Fig. 7 and Fig. 9), the flows performance is not fair, for both protocols, number of nodes and maximum speed of movement.



(a) Nodes=10, MS=6 m/s (b) Nodes=20, MS=6 m/s (c) Nodes=30, MS=6 m/s

(d) Nodes=10, MS=12 m/s (e) Nodes=20, MS=12 m/s (f) Nodes=30, MS=12 m/s

Fig. 6. Throughput results of OLSR with area size = 100m×100m (multiple flows).



(a) Nodes=10, MS=6 m/s (b) Nodes=20, MS=6 m/s (c) Nodes=30, MS=6 m/s

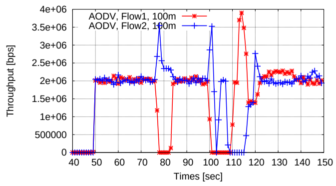
(d) Nodes=10, MS=12 m/s (e) Nodes=20, MS=12 m/s (f) Nodes=30, MS=12 m/s

Fig. 7. Throughput results of OLSR with area size = 200m×200m (multiple flows).

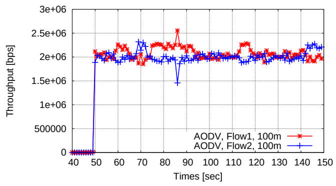
6.3 Results for Multiple Flows Considering UDP and/or TCP

For performance evaluation, we use two metrics: throughput and CWND. In Fig. 10 and Fig. 11, we show mixed results of single flows for UDP and TCP simulations. For small-size area the throughput of AODV and OLSR for both UDP and TCP, is almost the same. However, TCP flow can have better connectivity than UDP flow, as can be seen in Figs. 10(b) and 11(b).

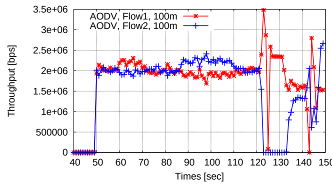
In Fig. 12 is shown CWND evolution in case of TCP single flow for AODV and OLSR. When TCP detects disconnections, CWND size decreases to half of its previous size. This decreasing of CWND size can be noticed in Fig. 12(b). For large-size area, the results of UDP and TCP are different. In case of 10 nodes, the area size is large and there are some



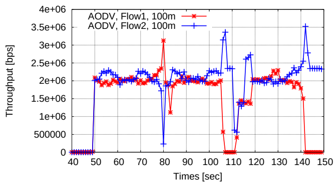
(a) Nodes=10, MS=6 m/s



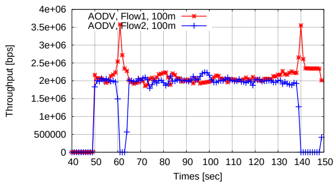
(b) Nodes=20, MS=6 m/s



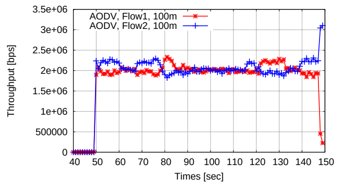
(c) Nodes=30, MS=6 m/s



(d) Nodes=10, MS=12 m/s

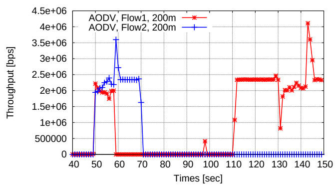


(e) Nodes=20, MS=12 m/s

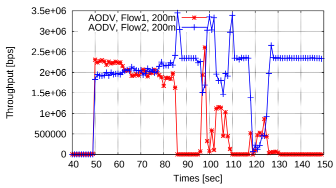


(f) Nodes=30, MS=12 m/s

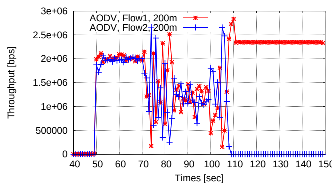
Fig. 8. Throughput results of AODV with area size = 100m×100m (multiple flows).



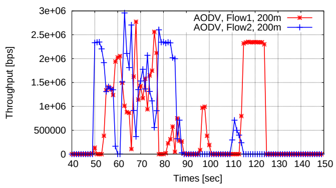
(a) Nodes=10, MS=6 m/s



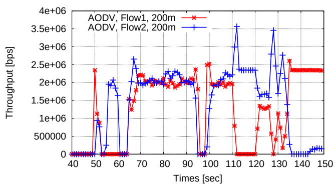
(b) Nodes=20, MS=6 m/s



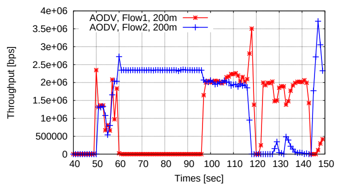
(c) Nodes=30, MS=6 m/s



(d) Nodes=10, MS=12 m/s



(e) Nodes=20, MS=12 m/s



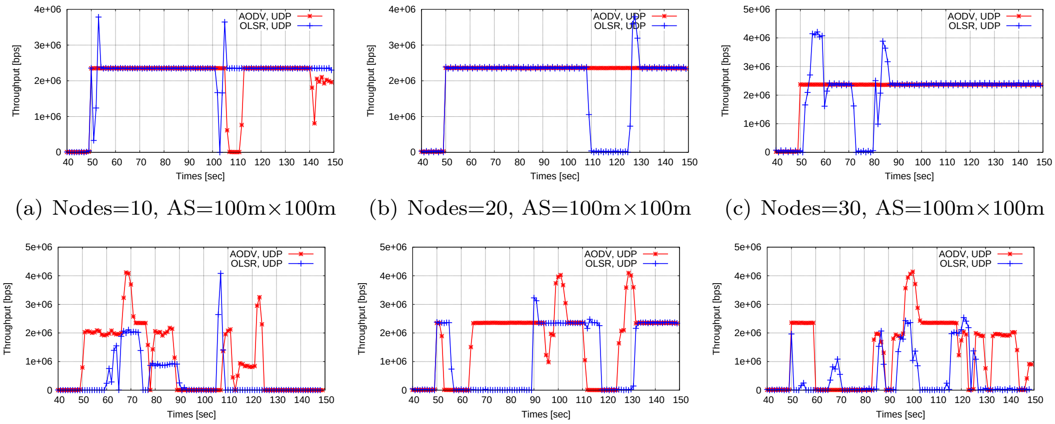
(f) Nodes=30, MS=12 m/s

Fig. 9. Throughput results of AODV with area size = 200m×200m (multiple flows).

islands in the network, so source node can not transmit to the destination. If we compare the results for AODV and OLSR, we can notice that CWND of AODV is higher.

In Fig. 13 and Fig. 14, we show the throughput results of UDPxTCP multiple flow data for AODV and OLSR, respectively. In case of AODV in large size area the fairness is higher than in small area size. For OLSR, we can notice that TCP performance is better in the beginning phases of transmission, because of the slow start mechanism. In all cases, we can notice that throughput of TCP reaches high values, since 50 seconds, when the transmission of data starts. Moreover, after 50 seconds, the throughput of TCP is more stable than that of UDP.

In Fig. 15 is shown CWND evolution in case of UDPxTCP for AODV and OLSR. From



(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m

(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

Fig. 10. Throughput results in case of single flow (UDP).

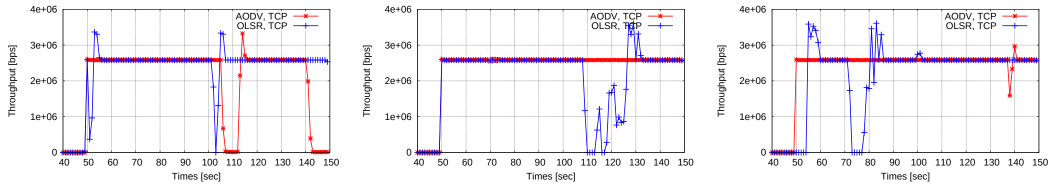
Fig. 15(c), we can notice that CWND of AODV increases during simulation. The increasing of CWND reflected the fairness problem in Fig. 13(c). In case of AODV in large size area the TCP congestion control performance is better than OLSR.

In Fig. 16, are shown the results of throughput of OLSR for Dual TCP flows data and in Fig. 17 are shown its CWND. The performance looks almost similar with the previous case, where we used UDP and TCP flows. But the fairness between flow1 and flow2 is low and flow1 cannot reach the destination for most of the time. When drop-tail queues are almost totally used by flow2 data, the data of flow1 can not access the queue. This effect is called *lock-out problem* and causes the low throughput and unfairness between flows. When a multiple flows communication is sent to a destination, and packets from two TCP flows are dropped, there is a policy, which forces TCP senders to start a *global synchronization* phase. In this phase, all TCP senders enter a slow-start status, by decreasing their CWND. In our dual TCP flow case, the *global synchronization* does not happen.

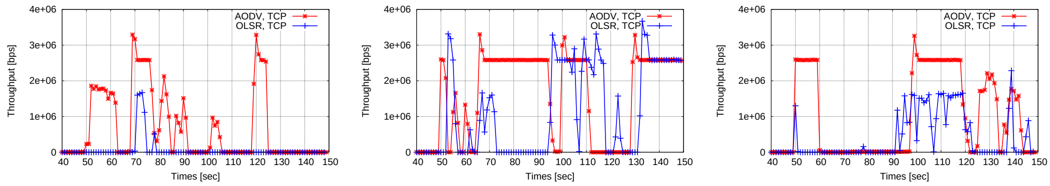
In Fig. 18 are shown the results of throughput of AODV for Dual TCP flows data and in Fig. 19 are shown its CWND. In large size area the fairness is better than in small area size. From Figs. 18(a) to 18(c) and Figs. 13(a) to 13(c), we can notice that the throughput results are almost same. While for large size area when number of nodes increases, the performance also increases (see Fig. 18(d) to Fig. 18(f)). From Fig. 19, we can notice that CWND of Flow1 is unfairness. In case of small area size when the number of nodes is 20 and 30, we can not get any data. When number of nodes is 20, the source node can not communicate with the destination because there are island in the network. When number of nodes is 30, we can notice the it lock-out problem.

7 Conclusions

In this paper, we evaluated by simulations the simulation runtime, memory usage, (average) throughput, percentage of received CBR packets and CWND of a MANET, considering AODV and OLSR protocols and sending multiple flow transported over UDP or TCP. We made simulations for different number of nodes in the network (10, 20 and 30) and for different

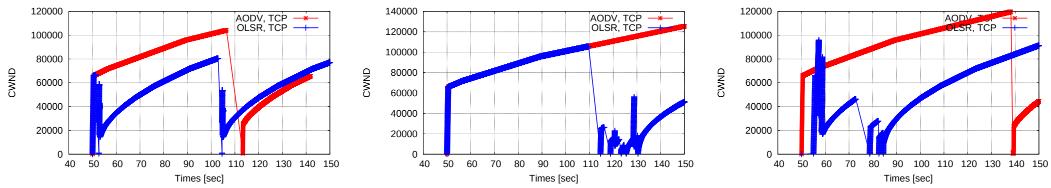


(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m

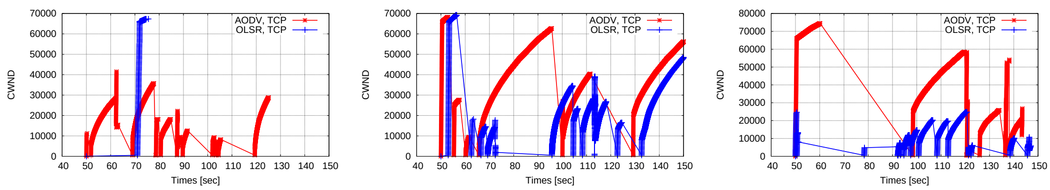


(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

Fig. 11. Throughput results in case of single flow (TCP).



(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m



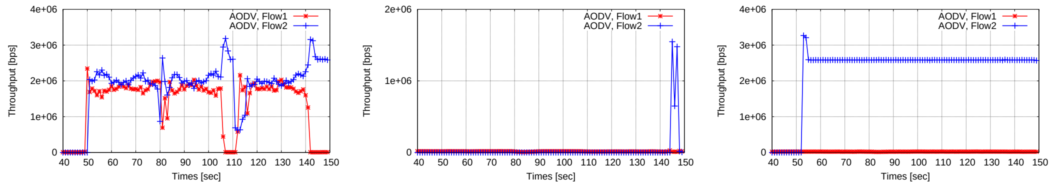
(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

Fig. 12. Congestion window evolution in case of TCP single flow.

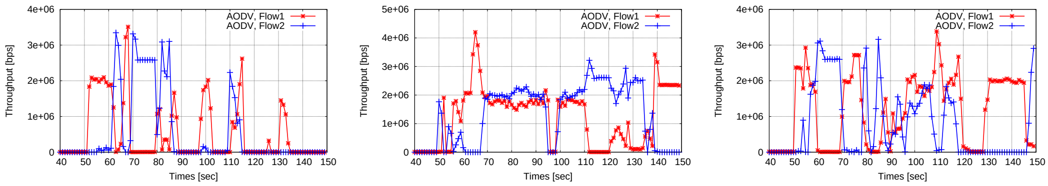
area sizes (100m×100m and 200m×200m) and compare the results for both protocols. We used random waypoint mobility model with a randomly chosen speed, uniformly distributed between 1-6 m/s or 1-12 m/s, log-distance path loss model and constant speed delay model.

From our evaluations, we found the following results.

- When the network size increases, the simulation runtime for ns-3 simulator is smaller than ns-2.
- As the grid size increases the hop distance also increases linearly. The number of received packets starts to decrease for grid size more than $N = 12$.
- Regarding AODV for small-size area with maximum speed set to 6 m/s, the performance is almost the same for different number of nodes. When the maximum speed is set to 12 m/s, the performance is increased when number of nodes increases.
- For 30 nodes in the network, OLSR uses more control packets to keep the topology updated. This is also shown by the decrease of P_{CBR}.

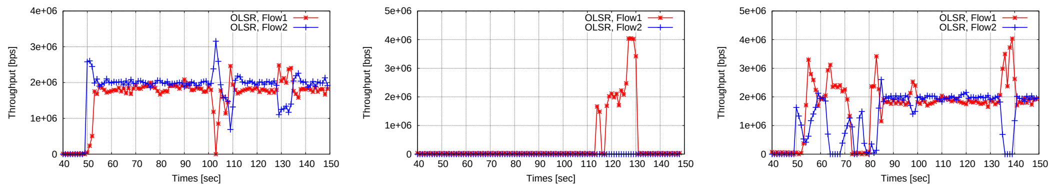


(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m

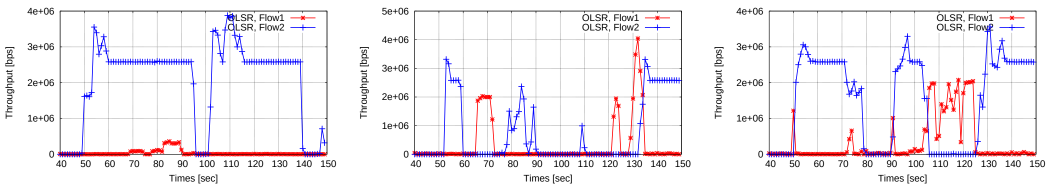


(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

Fig. 13. AODV throughput results in case of UDPxTCP: UDP (Flow1) and TCP (Flow2).

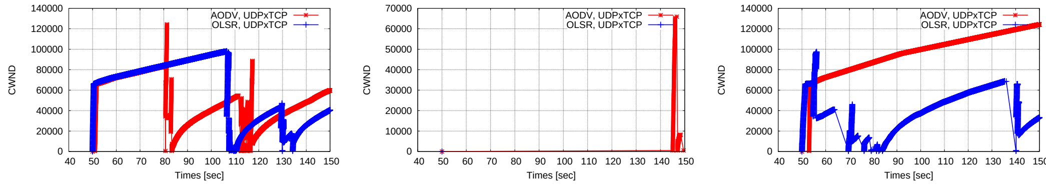


(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m

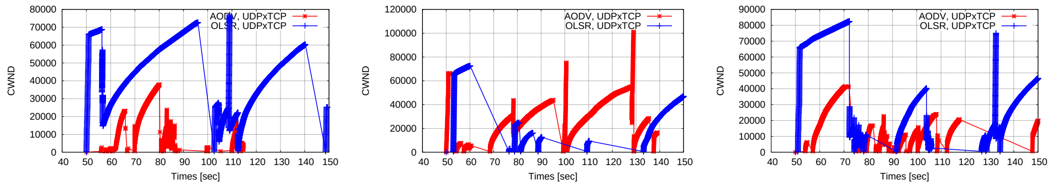


(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

Fig. 14. OLSR throughput results in case of UDPxTCP: UDP (Flow1) and TCP (Flow2).



(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m



(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

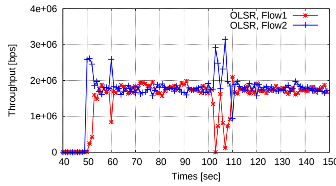
Fig. 15. Congestion window evolution in case of UDPxTCP (Flow2).

- Regarding the maximum speed of movement, for different number of nodes (10, 20 and 30), when speed is faster the performance decreases.
- For large-size area, there is no fairness between flows and OLSR had more disconnections than AODV.
- In case of single flow data, for small-size area the throughput for both UDP and TCP is almost the same. However, TCP flow can have better connectivity than UDP flow.
- While for large-size area, in case of 10 nodes, there are some islands in the network, so source node can not transmit to the destination.
- For dual flow UDPxTCP, TCP performance is better in the beginning phases of transmission, because of the slow start mechanism.
- Also after the slow start, the throughput of TCP is more stable than UDP.
- In case of OLSR, when dual TCP flows are used, fairness between flow1 and flow2 is low.
- Flow1 cannot reach the destination, because of the *lock-out problem*.
- In our dual TCP flow case, we could not see the *global synchronization*.
- In case of AODV for UDPxTCP in large-size area the TCP congestion control performance is better than OLSR.
- While for dual TCP in large-size area the fairness is better than in small-size area.

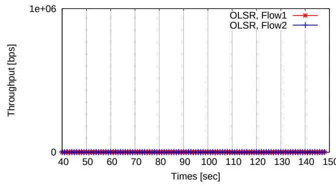
These evaluation where performed using a single and multiple traffic. In the future, we would like to extend our simulation systems, evaluate other mobility models and compare simulation results with testbed results.

Acknowledgements

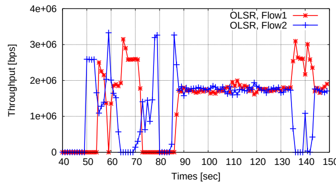
This work is supported by a Grant-in-Aid for scientific research of Japan Society for the Promotion of Science (JSPS). The author would like to thank JSPS for the financial support.



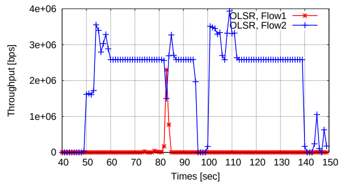
(a) Nodes=10, AS=100m×100m



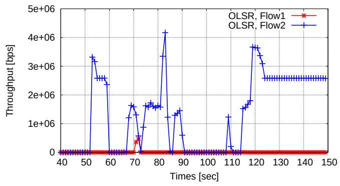
(b) Nodes=20, AS=100m×100m



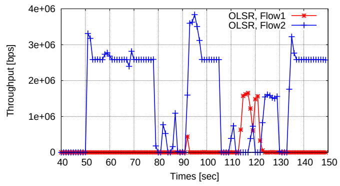
(c) Nodes=30, AS=100m×100m



(d) Nodes=10, AS=200m×200m

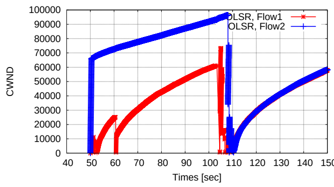


(e) Nodes=20, AS=200m×200m

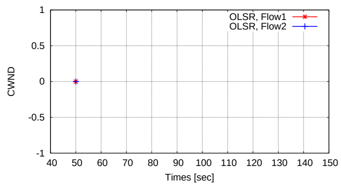


(f) Nodes=30, AS=200m×200m

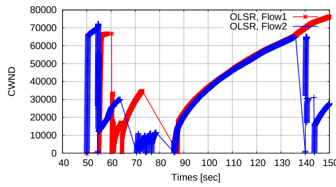
Fig. 16. OLSR throughput results in case of Dual TCP.



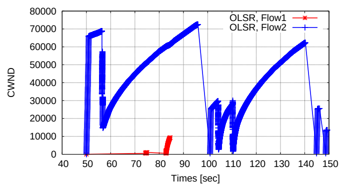
(a) Nodes=10, AS=100m×100m



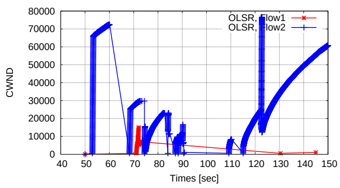
(b) Nodes=20, AS=100m×100m



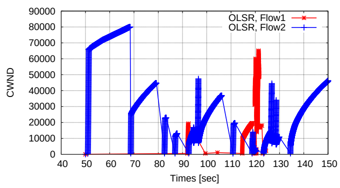
(c) Nodes=30, AS=100m×100m



(d) Nodes=10, AS=200m×200m

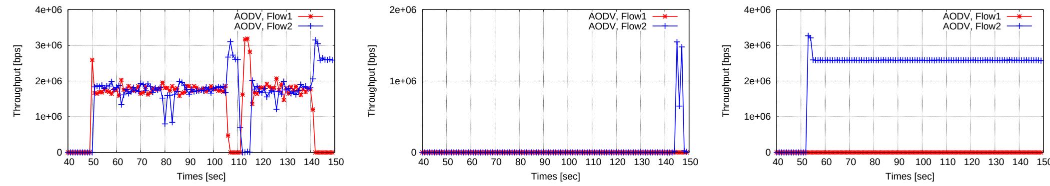


(e) Nodes=20, AS=200m×200m

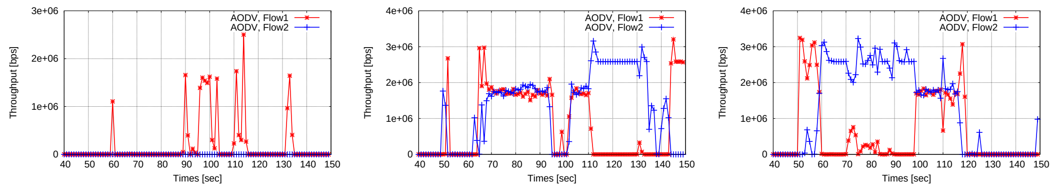


(f) Nodes=30, AS=200m×200m

Fig. 17. OLSR congestion window evolution in case of Dual TCP.

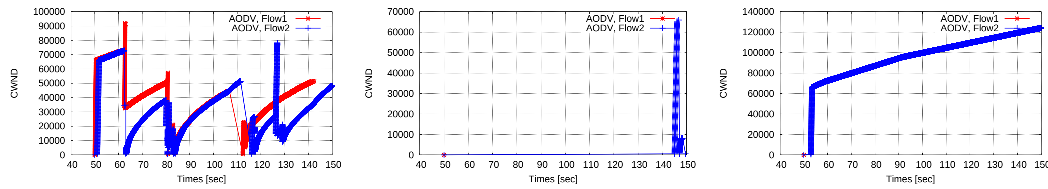


(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m

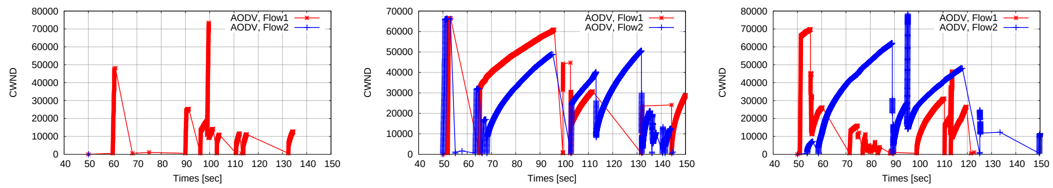


(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

Fig. 18. AODV throughput results in case of Dual TCP.



(a) Nodes=10, AS=100m×100m (b) Nodes=20, AS=100m×100m (c) Nodes=30, AS=100m×100m



(d) Nodes=10, AS=200m×200m (e) Nodes=20, AS=200m×200m (f) Nodes=30, AS=200m×200m

Fig. 19. AODV congestion window evolution in case of Dual TCP.

References

1. J. L. Font, P. Iñigo, M. Domínguez, J. L. Sevillano, and C. Amaya, "Architecture, design and source code comparison of ns-2 and ns-3 network simulators," in *Proceedings of the 2010 Spring Simulation Multiconference (SpringSim '10)*, no. 8, 2010, pp. 109:1–109:8.
2. M. Ikeda, L. Barolli, G. De Marco, T. Yang, and A. Durresi, "Experimental and simulation evaluation of OLSR protocol for mobile ad-hoc networks," in *Proc. NBiS-2008*, Turin, Italy, September 2008, pp. 111–121.
3. M. Ikeda, E. Kulla, M. Hiyama, L. Barolli, and M. Takizawa, "Analysis of manet routing protocols for indoor environment," in *The 5th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA-2010)*, November 2010, pp. 9 – 16.
4. M. Ikeda, E. Kulla, L. Barolli, and M. Takizawa, "Wireless ad-hoc networks performance evaluation using ns-2 and ns-3 network simulators," in *The Fifth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS 2011)*, June 2011, pp. 40 – 45.
5. E. Nordström, "Ape - a large scale ad hoc network testbed for reproducible performance tests," Master's thesis, Uppsala University, 2002.
6. R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *SIGCOMM '04*, 2004, pp. 133–144.
7. A. M. Hanashi, A. Siddique, I. Awan, and M. Woodward, "Performance evaluation of dynamic probablilistic flooding under different mobility models in manets," in *The 13th International Conference on Parallel and Distributed Systems (ICPADS-2007)*, vol. 2, December 2007, pp. 1–6.
8. A. M. Hanashi, I. Awan, and M. Woodward, "Performance evaluation with different mobility models for dynamic probabilistic flooding in manets," *Mobile Information Systems*, vol. 5, no. 1, pp. 65–80, 2009.
9. T. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The optimized link state routing protocol, evaluation through experiments and simulation," in *Proc. of IEEE Symposium on Wireless Personal Mobile Communications*, September 2001, http://hipercom.inria.fr/olsr/wpmc01.ps.
10. E. Weingartner, H. Vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," in *IEEE International Conference on ICC'09*, 2009, pp. 1287–1291.
11. A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008)*, March 2008, pp. 60:1–60:10.
12. NS-3, "The ns3 network simulator," http://www.nsnam.org/, 2011.
13. K. Müller, "SimPy simulation package homepage," http://simpy.sourceforge.net/, 2010.
14. M. Ikeda, E. Kulla, M. Hiyama, and L. Barolli, "Mobile ad-hoc network routing protocols performance evaluation using ns-3 simulator," in *The Third IEEE International Conference on Intelligent Networking and Collaborative Systems 2011 (INCoS 2011)*, 2011, pp. 14–20.
15. *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society Std., June 2007. [Online]. Available: http://standards.ieee.org/getieee802/download/802.11-2007.pdf