The availability of interactive multimedia applications (skype, wengo, LiveCom, etc.) is going to modify the usage of future Internet. Such Internet is likely to be more heterogeneous, consisting of different islands of ad-hoc networks. This argues for the growth of interactive applications traffic at the ad-hoc networks level. Providing Quality of Service (QoS) support, in particular low loss rate and low end-to-end delay, in such networks is a challenging task. Limited bandwidth resource and high mobility are two major characteristics of such networks. Moreover, the link breakage rate is high, which leads to high loss rate in the network. In the literature, researches generally span over routing protocols and MAC Layer mechanisms [1] to guarantee few QoS aspects by introducing signalling traffic.

Wireless Alternative Best Effort (WABE) [2] is an initiative to introduce QoS in ad-hoc networks, without guarantees but only via an alternative to classical Best Effort IP. WABE aims to decrease real-time traffic delay and by the way increase throughput of non-isochrone’s traffic: This is achieved without inducing any overhead. In this paper, we consider supporting QoS in ad-hoc networks, with a case study of an application layer multicast. We study how beneficial introducing queuing techniques at the level of ad-hoc nodes routing, could be for a delay-sensitive application running on ad-hoc network. In particular, we demonstrate the effectiveness of WABE on the ad-hoc overlay multicast protocol (AOMP) through simulations.

1. AOMP: AD-HOC OVERLAY MULTICAST PROTOCOL

AOMP [3] is an efficient topology-aware Ad-hoc Overlay that aims to construct application-layer multicast trees without inducing measurements overhead. AOMP exploits the properties of IP-routing to extract underlying topology information and construct a virtual topology closer to the actual underlying network topology. The basic idea is to match nodes path to the source in order to detect near neighbours in the physical topology, by relying on reactive routing protocols that maintain route paths from a source to a destination. Then, in a dynamic and decentralized way, AOMP constructs a minimum cost mobility-aware delivery tree, connecting nodes that are close to each other. A tree improvement algorithm enhances the global performance of AOMP during data distribution, while adaptation to nodes’ mobility is detected and triggered by the network layer. 2.

2. THE WABE ALGORITHM

Wireless Alternative Best Effort service [2] offers applications two alternatives; either transmitting packets with low end-to-end delay or receiving more overall throughput. WABE is an adaptation of the Hurley ABE service. Since, one objective of WABE is to decrease the impact of nodes mobility; routing packets have the highest priority. Typically, they are inserted at the head of the WABE queue. Every best effort packet is tagged as either green or blue (representing respectively isochrones and non-isochrones traffic flows). Green packets should receive a low, bounded queuing delay. To ensure that blue packets do not suffer as a result, green flows receive lower throughput during bouts of congestion. Figures 1 and 2 provide an overview of the WABE algorithm.

We implement the WABE service with two queues (blue and green) and a virtual queue. The virtual queue represents the traditional FIFO buffer. It determines the type of packets (blue or green) that should be served. Figure 1 resents the enqueuing algorithm in WABE.

Figure 1: Enqueuing algorithm.

At each packet arrival, the algorithm duplicates it and tries to enqueue the duplicate into the virtual queue. If the latter is full, the packet and its duplicate are dropped. Else, the type of the packet is examined. If it is a real-time (green) packet, it must be served within d seconds since its arrival. If this condition cannot be satisfied, the packet is dropped; else, the packet is enqueued in green queue. The main intuition behind the condition above is to avoid transmitting useless packets and thus to decrease network load, since isochrones packets would be dropped, in all cases, by
their final destination if their end-to-end delay exceeds some threshold. We introduce then the condition above in order. If the packet is blue, we compute its deadline as follows:

$$\text{deadline}(\text{blue}) = t + \left( \frac{q(t)}{c} \right) \times N(p)$$  \hspace{1cm} (1)

This deadline represents the waiting time of blue packets in the virtual buffer size (i.e., in flat best effort network). It is equal to $t + \frac{q(t)}{c}$, where $t$ is the arrival time, $q(t)$ is the virtual buffer size and $c$ is the service rate. Taking into account the number of nodes that the green packet crossed, a new coefficient $N(p)$ is added to the deadline. In this way, $N(p)$ increases deadline of each blue packet when a green packet in the buffer crosses several nodes and consequently, are getting closer to their destination. In other words, green packets gain higher priority on the blue packets when they are getting closer to their final destination. Figure 2 describes the serving algorithm. If the head of the virtual queue is green, the green queue is served and the duplicate green packet in the virtual queue is dropped. If the queue head is blue, the algorithm checks if the deadline of the in-head blue packet expired. As long as the deadline did not expire, green packets are served. We say that blue packets can "wait". So, green packets benefit from a lower delay. When no more green packets exist or the blue packet’s deadline expires, these packets are served and their duplicates are dropped from the virtual queue.

3. SIMULATION

We used NS-2 (Network Simulator) to build our simulations, with 60 mobile nodes moving in a 700m x 700m. Each simulation lasts 900 seconds. Initial locations of the nodes are obtained using a uniform distribution. All nodes have the same transmission range of 150 meters. We used the random waypoint model as the mobility model, assuming each node moves independently with the same average speed of 2m/s and the same pause time of 2s. The simulated traffics are: (i) Green (real-time) traffic: with CBR traffic for the overlay source. 30 nodes participate in the AOMP multicast session. The sending throughput of the CBR source is vary- throughputs. One major observation is that, the preventive drops of real-time packets (which waiting time in the buffer exceeds d) in WABE, increases the success ratio when the network is overloaded. The difference between the three approaches is more significant when the sending throughput exceeds a threshold value that saturated the network. This observation is compounded for the WABE1 case, where all nodes support WABE, as it outperforms WABE2.

3.1 Success packet ratio

Figure 3 depicts the success packet ratio of all packets (both green and blue) as function of the CBR sending throughput. One major observation is that, the preventive drops of real-time packets (which waiting time in the buffer exceeds d) in WABE, increases the success ratio when the network is overloaded. The difference between the three approaches is more significant when the sending throughput exceeds a threshold value that saturated the network. This observation is compounded for the WABE1 case, where all nodes support WABE, as it outperforms WABE2.

3.2 End-to-end delay

Figure 4 shows the end-to-end delay variation according to the CBR sending throughput. Again, we observe that WABE allows the application to perform better, thanks to its preventive drops, and prioritization of green packets (real-time) that are getting closer to their destination. In WABE2, the incurred delay is roughly 25% less than FIFO, with a sending throughput of 70 kbps.

4. REFERENCES