Middleware Lab 06 :
Protocol enhancements for improved robustness and better performances

Guillaume-Jean Herbiet
<guillaume.herbiet@uni.lu>

December 12, 2008
December 19, 2008

Abstract

During the previous labs, you implemented a complete protocol model for ad hoc instant messenging in OMNeT++ , an event-based discrete network simulator, including realistic mobility and radiowaves propagation model and instrumentation for statisctics collection.

Using this model, you ran different simulation campaigns in order to assess the performance of the protocol you implemented under different conditions.

In this lab, you will use your interpretation of the previous simulation campaigns in order to improve the performance of the application protocol and make it more robust and reliable.

1 Objectives

During the five previous labs, you designed the ad hoc instant messenging protocol and assess its performances under different conditions.

Based on your interpretation of the simulation campaigns results, but also of your understanding of the protocol, your experience and your insights, you should find one or several ways to make it more robust and perform better.

In this lab, your objective is to use all the knowlegde your gain about this protocol, and the OMNeT++ network simulator so to design and implement protocol improvements. Contrary to previous labs, no guided advice will be given to you, so you are really free to let your creativity drive you.

2 Possible ways of impovements

In this section, only ideas on what can be improved will be given to you. You don't have to follow all those suggestions, and your work should not be limited to them. They are just insights on what can be done, based on typical design rationale from real-world ad hoc protocols.

2.1 Robustness to losses

You might have seen that the application protocol is very sensitive to packet loss (one beacon loss is sufficient to consider a neighbor as gone, one call data loss might cause a ongoing call to be canceled,...).

Some approaches that are used to cope with this problem are the following:

- Make **deletion timers longer** than their associated message periodicity (e.g. keep info about users more than a beacon period)[1]. The good question is how long this deletion timer should be. Consider the drawbacks of setting it too short/to long...
- Use an **ARQ** (Automatic Repeat-reQuest) mechanism[2], so when an expected message doesn't arrive, it is repeated. Some well-known ARQ mechanisms are stop-and-wait, go-back-n and selective repeat.

2.2 More efficient data delivery

Wireless ad hoc networks operates in a constrained environment where many nodes are contending for the same channel (a given radio frequency) for communication. Therefore, optimizing the bandwidth usage is a crucial design objective.

To reduce the bandwidth requested by your protocol, you can use the following ideas:

- **Message piggybacking**: as many messages are network broadcast, it might be more efficient to wait and send several application messages in a single packet. This will reduce the MAC layer overhead. But this also comes with some drawbacks (think in terms of robustness and protocol reactivity).
- Some messages are clearly **unicast** (addressed from a single server to a single receiver). So a simple idea would be not to use broadcast for those messages. But when messages have to be relayed (i.e. the sender is not directly in sight of the receiver), you need to know to who you might send the packet for relaying (this is called a routing decision).

2.3 Other ideas

These are other general ideas, you can choose to apply:

- The more dense is a network area, the more collisions can occur and the less bandwidth is available for each node. So number of messages sent for a given period of time could be decreased in case of high density.
- On the contrary, when a node is moving fast (i.e. faster than a given threshold), it is likely to change neighborhood very often, so beaconing messages should be sent faster;
- When nodes are inside a spot, they might not be interested any longer by interacting with nodes outside this spot. You could hence limit communications to nodes inside the same spot.

[1] This idea is used, for instance, in the OLSR routing protocol. See `http://tools.ietf.org/rfc/rfc3626.txt`

[2] See RFC3366 at `http://tools.ietf.org/rfc/rfc3366.txt`

3 Simulations with the propagation model

Once you have ensured that your improvements perform correctly, you can launch a real simulation campaign of at least 20 runs using the same configuration files you used for the simulation campaigns of previous lab.

Analyze and consolidate your results as you did in the previous labs to compair the obtained results and assess the impact of your improvements on the protocol performance.

4 What to turn in

This lab is concluding this series on ad network simulation. So, once you have completed this assignment, please ensure to turn in the following items:

- **Your complete model code**, zipped in a single archive file. It may contain your complete project directory (or directories if you changed working directory at each lab). Please remove unecessary files (temp objects, dump of traces, dat files for plotting) but keep the important ones (your code, configuration files, scalars and vectors file, outputs of plots). I should be able to replay your simulations and obtain identical results with what you provided.
- A **one page conclusion** on your analysis of the protocol behavior and performances. What is working, what is not. What where the impacts of mobility, propagation. What improvements have you implemented, what effect did they have on the performances. And, last, but not least, what will be the major take-away from this series of lab. Use bullets to present those items and go straight to the goal.
- There might be a **lab evaluation sheet**, requesting for your opinion on this series of assignments. You can fill it anonymously and turn it back at the end of last lab.

Conclusion

This series of lab introduced you with network simulators, an important tool in computer science research.

While playing with a modest application-layer protocol, you envisioned all the challenges related to the ad hoc nature of our network and were faced to the importance of finely modeling the application behaviour but also its environment (users mobility, propagation model).

All those challenges are currently the object of many research work, as ad hoc network and associated services still remain a wide open area.