DEMO: Reducing Latency in MANET File Access

Gabriella Dodero, Vittoria Gianuzzi, Daniele Venzano
{dodero, gianuzzi}@disi.unige.it , venza@brownhat.org
DISI, Università di Genova, Via Dodecaneso, 35 - 16146 Genova, Italy

Abstract This demo aims to show faster file access over a MANET by use of a proxy cache middleware, called MobEYE[1].

Keywords: *proxy cache, middleware for wireless networks, MANET*

File transfer is one of the more time (and power) consuming features in a MANET. When an application running on some node (say A) requests an image (File.jpg) to the owner node (B), such a file has to be routed through all nodes in the path connecting the file owner to the requesting one.

To optimize this costly access, we implemented a system providing a cooperative cache implementation over a MANET, called MobEYE (Mobile intErcepting proxY cachE). It woks in a WEB environment, and it supports read only access to remote files by means of any kind of browser. MobEYE implements a light web server on every node where it is installed, and allows to share the local file cache with other nodes, answering to requests from other peers even if originally directed to another file server.

MobEYE, described in [Dod06], may work with any kind of routing protocol and operating system, without requiring explicit information from the underlying layers, nor any modification to them. It works even if not all the nodes of the MANET are collaborative.

MobEYE performance has been estimated by simulations with ns2[ns], considering a number of nodes varying from 40 to 80, moving in accordance with the Random Waypoint model. Transmission uses the two-ray ground propagation model, and 180 meters radius. Dense simulations scenarios were generated by BonnMotion[bonn], considering rectangular areas where each node has out degree in [6, 7], speed chosen randomly between [0.5, 1.38], and pauses in [0, 60]. Disconnection probability for every pair of nodes is about 0.1, and average file size is 10KB. From 60 to 80 files are initially stored in 6 to 8 servers, and each file is requested 31 times (average) to the respective server, after a random choice, from all nodes. Caching policy is LRU, and different cache sizes have been tested, containing from 5 to 20 percent of the total file size.

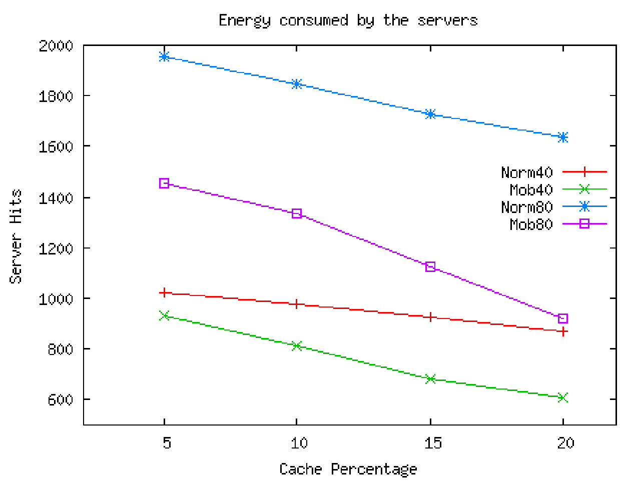

Figure 1 (a) Number of requests to servers

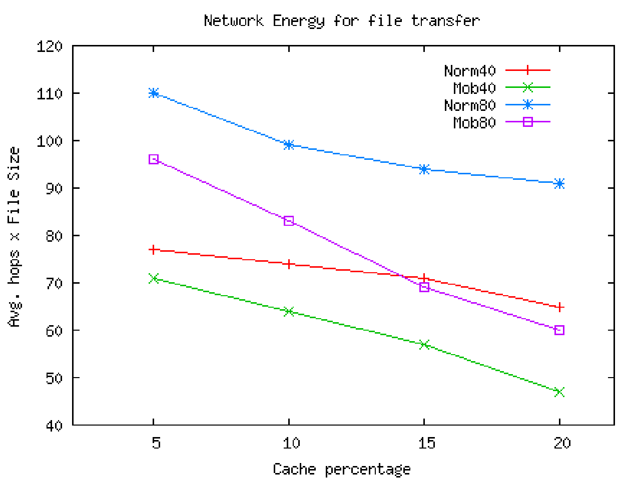

Figure 1 (b) Energy needs in the whole network

1This work has been financially supported by CNR, the Italian National Research Council, as a part of the IS-MANET project, and by MIUR, the Italian Ministry of Research, by the FIRB WEB-MINDS projects

The above figures respectively show main simulation results, that is the decrease in number of file requests satisfied by the file servers (i.e. satisfied by some node along the routing path, closer to the client location) in Figure 1 (a), and the decrease in energy consumed by the whole network to transmit files over the routes (directly depending on the probability to find a replica in the local cache, and on the length of communication paths) in Figure 1 (b).

Lines labeled as "Norm" refer to results obtained with n nodes MANETS without MobEYE (i.e. with only a local browser cache), while lines labeled "Mob" consider MANETs with MobEYE (i.e. with a cooperative cache mechanism). Figure 1(b) documents also latency reduction, in terms of number of hops, since the values reported in the y axis are scaled by file size. From Figure 1(b) we see that actual reduction in number of hops is between 1 and 2 hops in the average, so this quantity is interesting to be measured to evaluate expected reduction in file transfer timing.

MobEYE handles the caching mechanism by employing a simplified version of the ICP protocol, i.e. decoupling file requests (acknowledged by all nodes having a cached copy) from actual file transfer (which takes place only from the closest node). So we expect to pay a small penalty in terms of time and energy for the additional acks, negligible with respect to the shorter file transfers. However query timings are interesting as well to be measured.

These results, obtained by simulating a dense scenario with tens of nodes, could be validated by actual measurements, collected within demos, even with a smaller number of participating nodes, as remarked above. So we are going to measure how much file transfer timings may be reduced by using MobEYE in a multi-hop ad-hoc network, composed by 2 laptops and 2 PDAs, demonstrating in practice how much would be gained on a MANET by installing such middleware on all or some nodes.

Our measurements can be summarized by the following considerations:

- Time requested to query a file (if it exists, i.e. some server responds in the affirmative) is actually negligible with respect to file transfers, when size is significantly bigger than 10KB. We have obtained an average query time of 0,0128 seconds in a laptop-to-laptop connection, and a higher query time of 0,107 seconds in a laptop-to-PDA connection (approx 8 times slower).
- Latency is comparable to twice the query time, since the ICP protocol sends a file request, receives an answer and starts transfer from the first answering node.
- File transfer timings are in the average proportional to file length, but with a significant variance depending on actual network conditions.

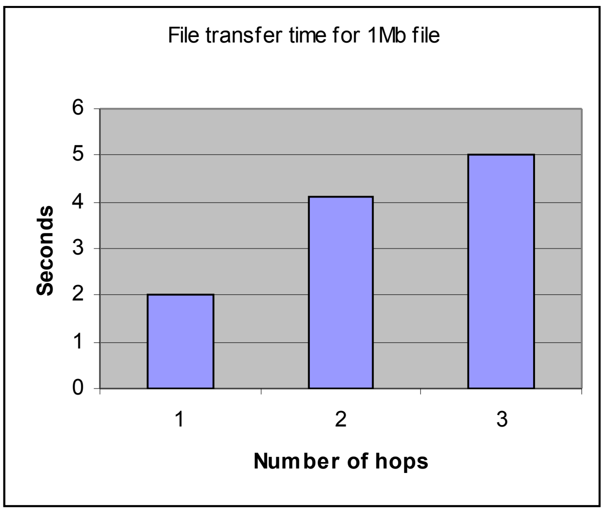

Figure 2. File transfer time for 1 MB file on ad-hoc networks with 1,2 or 3 hops

Figure 2 shows the relationships between file transfer timings and actual number of hops, for a 1MB file being transferred with one, two or three hops. The actual gain in timigs, expected by the use of MobEYE, ranges around 3 seconds for 2 saved hops.

The above measures for file transfers have been taken in a relatively "clean" environment, i.e. in an indoor lab, yet the data is consistent with measurements taken in the open air, noisy environment (harbor), and reported in [Venz05].

On PDAs, we may also measure actual energy saving, following the procedure explained in [Dev05]. Actual measures on laptops are not so easy to obtain, since it is more difficult to evaluate the amount of battery spent by the hard disk and the display.

References

[bonn] http://web.informatik.unibonn.de/IV/Mitarbeiter/dewaal/BonnMotion

[Dod06] G.Dodero, V.Gianuzzi, Saving energy and reducing latency in MANET file access, to appear in IWDDS Proceedings, Lisbona, July 2006.

[Dev05] P.Devevey, N.Lorenzon, C.Tambary, Measuring wireless energy consumption on PDAs and on laptops, IS-MANET project Technical Report, Nov.2005, http://zeus.elet.polimi.it/is-manet/Documenti/techRep-disi-3.pdf

[ns] http://www.isi.edu/nsnam/ns/

[Venz05] D.Venzano, *Dynamic routing algorithms applied to wireless networks in harbour areas* (in Italian), Graduation Thesis in Computer Science, University of Genova, November 2005.