University College London
Department of Computer Science
MSc Data Communications, Networks and Distributed Systems

Executive Summary

A Spatio-Temporal Middleware for Delay Tolerant Systems

MiDAS:

Middleware Implementation for DTN's with Autonomic Selection

Group Members:

Jessica Allen
Jiaxi Fu
Chukwuka Oyem
Dmytryo Stasyuk
Xueyu Jin

Supervisor:

Dr. Cecilia Mascolo

September 2, 2005

In recent years there has been an increase in the use of mobile and wireless networks, ranging from mobile phone networks to wireless sensor networks. These networks have new communication and co-ordination patterns such as asynchronous and location based communication, concepts that are not taken into consideration by the communication primitives used in fixed, wired systems. In order to make effective use of these networks, new communication primitives that take these factors into account must be developed.

The main objective of our project was the design, implementation and evaluation of a spatio-temporal middleware for delay tolerant systems. More specifically, we aimed to implement a middleware that provides application level communication primitives that take into account the spatial and temporal dimensions of delay tolerant systems and are based on factors such as message delivery reliability, sender and receiver location, and temporal validity of messages. We chose to develop the middleware for delay tolerant systems as they can be considered as an abstract generalisation for systems characterised by frequent disconnections and long, variable delay paths, which among others, include both mobile and wireless networks. Our middleware is a novel concept in the sense that rather than providing a network level solution for a specific situation, as other projects in this area have done, it provides an application level solution intended for a variety general scenarios. Our work was based on a paper entitled "Spatio-Temporal Interaction Primitives for Delay Tolerant Systems" which presented a formalisation for a send, receive and anycast primitive tailored to delay tolerant systems.

Further to the development of the actual middleware, we implemented the epidemic dissemination strategy, destination sequenced distance vector (DSDV) routing and context-aware adaptive routing (CAR) as protocols to disseminate messages. We designed the middleware in a component-based manner thus allowing other dissemination strategies to be plugged-in as required. Furthermore, we designed and implemented an autonomic selection engine, which chooses the appropriate dissemination strategy for a message based on user requirements and context, network topology and availability.

We performed an extensive set of functional tests on our middleware and dissemination strategies in order to validate the implementation. Functional testing was a rather important part of the testing process as our middleware is a first prototype, and in order to be of any relevance or merit future work it had to be shown that it functioned correctly. We also simulated our middleware using a discrete event simulator for Java called SWANS. The chosen simulation tool was rather difficult to learn and use which combined with time constraints prevented us from obtaining as many simulation results as anticipated. That being said, we simulated and gathered results for what we considered to be the most important non-functional aspects of the middleware, which are scalability, message delivery ratio, and density.

At the end of our project we did a critical analysis of the project, which examined both our successes and challenges. We felt that the most successful aspect of the project was the design and development of a middleware that met the requirements and specifications set out at the beginning of the project. The main goal was to develop a middleware that provides application programmers with a set of communication primitives for delay tolerant systems to use in general scenarios, which was by all means accomplished. Furthermore, we consider our original implementation of three separate dissemination strategies to add to the overall success of the project. Not only were these strategies quite difficult implement correctly, but they allowed us to test our autonomic dissemination engine, which was an original concept.

We found that our greatest challenge was in the area of simulation. Although we started learning about simulation early on in the project, it had to be abandoned to complete the development of the middleware and dissemination strategies. Later in the project when we returned to the simulation, we found out that the chosen simulation tool was not as easy to use as it claimed to be and lacking in documentation and examples. Because of this, it turned out to be very challenging to port our middleware onto the simulator. Due to the difficulties we had using the tool combined with time constraints, we were not able to obtain as many results as anticipated. That being said, we did effectively evaluate what we consider to be the most important aspects of the middleware, which are scalability, message delivery ratio and density.

We feel that the functional and non-functional evaluation of our middleware shows that it merits future work. In the main report we have documented several ideas for the latter, which include an enhancement of the autonomic dissemination engine, the inclusion of more sophisticated mobility models as well as further work on the location based feature of the middleware.

Overall we feel that the project was very successful. Not only did we design, implement and test a framework for a novel middleware, but we learned a lot about delay tolerant systems, communication primitives, dissemination strategies and simulation.