Performance Comparison of a Network Layer ACP and an Application Layer ACP via Simulations

ROHAN DE SILVA
School of Information Systems, Technology & Management
University of New South Wales
Sydney 2052
AUSTRALIA

Abstract:- In this paper, we present the performance of a new inter-agent communication protocol that operates in the network layer. Its performance is compared with the performance of an inter-agent communication protocol that operates in the application layer. It improves the efficiency and reduces latency in finding a product match. The packets that are sent across the network using our protocol will be recognised by active routers, which will then be processed to identify different types of agents. Information of the merchant agents is stored in an active router that acts as an e-marketplace.

Key-Words:- Active Routers, Inter-agent Communication Protocol, E-marketplace

1 Introduction
Agent technology will play a key role in future e-businesses. Agents need to interact with their peers using a communication language. There are many agent frameworks, languages and protocols available today, specifically designed for inter-agent communication. Some of these are language-specific, while others are more general in nature. Agent developers look for the properties such as ease of use, ability to exchange text and binary data, type of message passing/queries and implementation possibility [1] when developing agent applications.

An Agent Communication Language (ACL) that allows agents to interact while hiding the details of their internal operations will result in agent communities that can handle problems that no individual agent could handle. Knowledge Query and Manipulation Language (KQML) [2] and Foundation for Intelligent Physical Agents (FIPA) ACL [3] are two such languages developed.

The concept of mobile agents has been developed to improve the communication between agents [4, 5]. Mobile agents can temporarily stop their execution at the client machine, propagate through the network to a distant location and start execution at the new environment.

It has been identified in [6] that to improve the communication between agents, and in particular to reduce latency, the agents should be allowed not only to propagate through the network but also to meet and perform negotiations within the network. These agents are called active agents. The active agents negotiate in e-marketplaces, in practice in an active node in the network. This active node may be a switch or a router. In [6], a framework for inter-agent communication in an active node has been introduced. This paper describes an Agent Communication Protocol (ACP) that can be used for such communication. We have developed this protocol and simulated on a tested.

The advantage of this method is that, by just examining few extension headers, the router and the customer’s agent can decide whether it is worth to continue the comparison or not. Apart from that, the router does not have to intercept and read the entire data stream of the agent. This reduces the processing time of the router and, in turn, the cost of using the active router. Moreover, the packets of the agent will not travel end-to-end across the whole network to communicate with the other agent. All these reasons contribute to the reduction of network traffic as well as to the decrease of response time.

The rest of the paper is organised as follows. In Section 2 of the paper, we discuss some recent work on agent communication languages and protocols and in Section 3, we introduce our proposed inter-agent communication protocol. After discussing the simulation issues in Section 4, we present the results and analyse them in Section 5. In Section 6, we conclude the paper and discuss possible future work.

2 Background
An ACL provides agents with a means of exchanging information and knowledge. There are also other
methods used to seamless exchange of information and knowledge between applications. From Remote Procedure Calls (RPCs) and Remote Method Invocations (RMIs) to Common Object Request Broker Architecture (CORBA), the goal has been the same. What distinguish ACLs from such past efforts are the objects of discourse and their semantic complexity. ACLs stand a level above CORBA.

An ACL has many dialects and variants, and KQML illustrates the basic concepts of all these. KQML is a high-level, message-oriented communication language for information exchange independent of content syntax and applicable ontology. Thus, KQML is independent of the transport mechanism, independent of the content language and independent of the ontology assumed by the content. Conceptually, we can identify three layers in a KQML, namely, message content, communication, and the message itself. One of the design criteria for KQML was to produce a language that could support a wide variety of interesting agent architectures. However, after eight years of experimentation and experience, there are still serious signs of immaturity. In general, different KQML implementations cannot interoperate. There is no fixed specification sanctioned by a consensus-creating body. Finally, there is no agreed upon semantics foundations.

Java Agent Template Lite (JATLite) is a package of java programs developed at Stanford University that allows users to create communicating agents quickly [7]. Agents run as applets launched from a browser, and for that reason all agents register with an agent message router facilitator that handles message delivery.

In contrast, [6] describes an inter-agent communication protocol in the network layer (NL-ACP) that exploits the advantages of active networks and mobile agents. Here the ACP has a set of fields that are arranged in an inverted-tree hierarchy. The topmost field or the root field corresponds to the agent itself. A path from the root to a leaf in the merchant’s product tree specifies a particular product that the merchant has to sell and a path from the root to the leaf in the customer’s product tree specifies a particular product required by the customer. When the agents generate the packets, they simply create a main Internet Protocol (IP) header and a set of extension headers. The first extension header corresponds to the second level of the product tree, the second extension header to the third level and so on. For the current version of the Internet Protocol that uses IPv4, each of these IP datagrams are encapsulated in IP (IP in IP encapsulation). With the deployment of IPv6, the transmission of these packets will be straightforward as the IPv6 has the provision for extension headers [8].

Each merchant agent that is registered with an active router knows the IP address of the router and will send its active agent to the router. Once these packets arrive at the active router that is pre-loaded with the required code, it will intercept the data stream and store the information contained in the extension headers. The active router is now ready to work as an e-marketplace. If a customer’s agent passes through this router, the router intercepts the data stream and looks for a product match.

3 Proposed ACP Concept

The discrete approach (programmable switches) of active networking nodes, where service deployment is performed separately from service processing [9], is used in the design of this ACP.

![Fig 1 Agent interaction model with ACP](image.png)

When a packet compatible with the ACP protocol arrives, its header is examined and a program is dispatched to operate on the contents. The router will deliver the received packet to a transient execution environment so the pre-loaded program can actively process the packet. If our protocol is not recognised...
by a router, the router will forward the packets without processing the contents.

Fig 1 shows the communication of agents using our ACP in an active network. When merchants A, B and C desire to sell and/or advertise their products, they send a packet containing the ACP message to the active router (AR) to which they have permission to gain access. Those routers will then look for the authorisation code embedded in the ACP header, which then enable them to verify whether the merchants have the permission to gain access to the routers.

On the customer’s side, Customers A and B send ACP messages that contain details of the products that their users are looking for to a default active router. Generally a group of users may register for the service of an active router and always forward their buyer agents to that router. This active router has a built-in program which allows it to broadcast those messages to the active routers of other user groups that contain the merchant’s information. These active routers are programmed to recognise and decode ACP messages and hence, while helping merchant agents find potential customers, they also allow buyer agents to find sellers. As described above, active routers create an e-marketplace environment for agents to communicate and exchange information.

All ACP messages include an ACP header. Information of the agents is stored in this header so that the active router can quickly process and retrieve the necessary information it needs for the agent communication to take place.

4 Simulations
To compare the performance of inter-agent communication in the application level and that of the network level, we simulated the communication of a set of customer agents and merchant agents in a network comprising of active and legacy routers. The propagation delay from the customers to the router and back was set to 1.5 seconds. The simulation tool was written using Java.

In the simulation Scenario 1, we used a single customer agent and different numbers of merchant agents that communicate at a single e-marketplace. Only the 10th, 20th etc. of the merchant agents offer the product sought. The customer agent returns to the corresponding customer after the first successful product matching. We collected the delay experienced by a customer agent for the two cases when the communication takes place in the application layer as well as in the network layer.

Simulation results shown in Fig 2 clearly indicate that the delay for a transaction is much less if the ACP operates in the network layer. This is because the NL-ACP can identify that the product or service that is sought after is not available by just matching only a few fields in the header of the packets.

In the simulation Scenario 2, we changed the number of customer agents that visit the same e-marketplace in search of the same product. The round trip propagation delay for all the customers is 1.5 sec.
The number of merchant agents that reside in the e-marketplace is two. Only the second merchant agent offers the product sought by the customers. The customer agents return to their corresponding customers after the first successful product matching.

Fig 3 shows the simulation results for Scenario 2. It indicates that if the ACP operates in the network layer, the delay for a transaction is reduced even when the number of customers is more than one.

5 Conclusion
In this paper we have shown via simulations that the negotiation of agents in the network layer in an active networked e-market place can result in reducing latency of transactions. If the product or service does not match, it can be verified by comparing only a few fields in the case of IP version 4 and a few extension headers in IP version 6. Therefore, this method is very useful. Furthermore, it is also a natural choice for future e-market places as each product is anyway distinguished by a bar code.

References: