ROUTING DOMAIN AUTOCONFIGURATION
FOR MORE EFFICIENT AND RAPIDLY DEPLOYABLE MOBILE NETWORKS

K. Manousakis[i], A. McAuley, R. Morera, J. Baras[j]

{kerk@glue.umd.edu, mcauley@research.telcordia.com, raquel@research.telcordia.com, baras@isr.umd.edu}
Telcordia Technologies, Inc.
445 South Street, Morristown, NJ USA

ABSTRACT

One approach to achieving scalability in rapidly deployed dynamic networks, such as Future Combat Systems (FCS), is to automatically divide nodes into small (e.g., 30 node) interconnected IP domains and assigning each domain a routing protocol that best meets the characteristics of that domain (Morera and McAuley, 2002). However, as there has been no capability to do this within network configuration protocols, this approach has never been tried. This paper[*] presents the first realization of domain autoconfiguration through extensions to the IP Autoconfiguration Suite (IPAS) (McAuley et al., 2001; McAuley et al., 2002; Cheng et al., 2002). While IPAS already configures and reconfigures information such as interface IP address and server locations, it assumes a single domain. We describe IPAS enhancements that support the automatic creation and configuration of multiple domains and describe a prototype implementation where interfaces are dynamically assigned to run different routing protocol. Finally, we show some initial performance results for the configuration time and bandwidth overhead.

1. INTRODUCTION

Rather than designing domain autoconfiguration protocols from scratch, our objective was to enhance existing solutions. Commercial IP autoconfiguration protocols (e.g., Dynamic Host Configuration Protocol (DHCP) (Droms, 1997) and IPv6 Stateless Autoconfiguration (SA) (Thompson, 1998)) do not provide sufficient basis to build a domain autoconfiguration solution. The only current basis for domain autoconfiguration is to extend the IPAS (McAuley et al., 2001; McAuley et al., 2002), used in the CECOM MOSAIC Advanced Technology demonstration (Cheng et al., 2002). However, some enhancements are required in order to create multiple domains, as currently IPAS has no notion of domains or border nodes.

2. AUTOCONFIGURATION SUIT

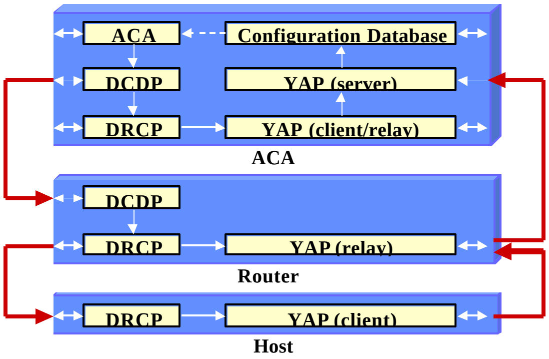

Figure 1 IPAS Elements

In IPAS, the ACA (Adaptive Configuration Agent) distributes new configuration through DCDP (Dynamic configuration Distribution Protocol) to nodes in each subnet. DRCP (Dynamic and Rapid Configuration Protocol) configures the interfaces within a subnet. Interfaces, configured by DRCP, report configuration information and nodes capabilities to the configuration server via the YAP protocol. The configuration server stores this information in the configuration database. To complete the cycle, the ACA node contacts the Configuration Database locally or remotely to get the latest configuration information. This is shown in figure 1. More details of the IPAS components can be found in the references (McAuley et al., 2001; McAuley et al., 2002; Cheng et al., 2002).

3. TOPOLOGICAL DOMAINS

Domains follow the architectural framework presented in (Morera and McAuley, 2002), which extends the use of domains to provide scalability and increse efficiency to the network layer protocols. We define different types of domains depending on the networking protocol we focus on, i.e. configuration, routing, security, QoS and multicast domains. A domain hierarchy is defined to allow for network and protocol scalability.

[*] This work was supported by the U.S. Army Research Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

[i] Kyriakos Manousakis is pursuing a Ph.D. at the University of Maryland. This work was done during a summer internship at Telcordia.

[j] Proffessor Baras is head of the Center for Satellite and Hybrid Communication Networks at the University of Maryland.

4. DOMAIN INFORMATION DISTRIBUTION

The information needed to properly configure domains must at least contain the following **a)** unique domain identifier **b)** domain membership information (each element must have a unique member identifier (UID)) **c)** domain configuration information, such as domain type (e.g., routing or configuration) or routing protocol to be run in a subnet.

Domain information is very critical so the information distribution mechanism must ensure that all elements in the domain receive the proper information at the right time. We considered different mechanisms to distribute the information **a)** unicast to the specific nodes (or all the neighbor nodes) within that domain; **b)** IP multicast tree and **c)** flooding. We qualitatively assessed the performance of these three mechanisms according to four parameters: bandwidth efficiency, information distribution delay, simplicity, and reliance on the routing protocol. We concluded that the best mechanism to distribute domain configuration information is by flooding. It is simple, it depends least on network dynamics and it is robust.

5. ENHANCEMENTS ON IPAS MODULES

In IPAS, the DCDP module distributes the configuration information. However, while the current DCDP implementation uses specific IP addresses and the underlying routing protocol to deliver the configuration information, we enhanced DCDP to flood all configuration information. DRCP has also been enhanced so it processes and stores domain configuration information. The interface between DCDP/DRCP at each local node must now inform DCDP about the configured interfaces, as DCDP only uses configured interfaces to flood domain information.

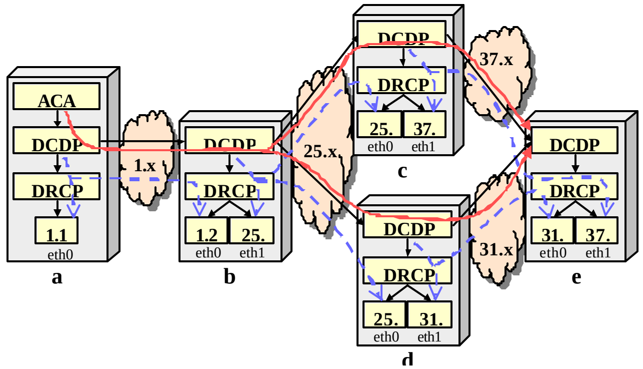

Figure 2 Information flow

DCDP information is analyzed at every node. When configuration messages reach the corresponding nodes, the local DCDP module passes these messages to the local DRCP process. DRCP then takes the appropriate configuration actions based on the received message. Figure 2 shows the information flow for a small network.

In order to enable this sequence of communication we had to design three groups of messages **a)** messages from ACA to DCDP **b)** messages from DCDP to DCDP **c)** messages from DCDP to DRCP. These messages are generated at the ACA and include **a)** domain generation **b)** domain rename, **c)** routing protocol assignment

6. PROTOTYPE PERFORMANCE ANALYSIS

We built a testbed to demonstrate the effectiveness of the domain configuration mechanisms and configuration information distribution. We run an experiment were reconfigured a single routing domain into two independent routing domains, each running a different routing protocol and a border router acts as a gateway between the two domains. As the configuration messages are a few bytes and information is distributed by broadcasting, networks of a thousand nodes can be configured in a few seconds.

REFERENCES

Cheng T., Gurung P., Lee J., Khurana S., McAuley A., Samtani S., Wong L., Young K., Bereschinsky M., Graff C., "Adhoc Mobility Protocol Suite (AMPS) for JTRS radios," Software Defined Radio (SDR) Forum, San Diego, November 2002.

Droms R., “Dynamic Host Configuration Protocol,” RFC 2131, March 1997.

McAuley A., Misra A., Wong L., Manousakis K., “Experience with Autoconfiguring a Network with IP addresses,” IEEE Milcom, October 2001.

McAuley A., et al., “Automatic Configuration and Reconfiguration in Dynamic Networks”, To appear Army Science Conference (ASC), December 2002.

Morera R., McAuley A. “Flexible Domain Configuration for More Scalable, Efficient and Robust Battlefield Networks” MILCOM, October 2002.

Thompson S., “IPv6 Stateless Address Autoconfiguration,” RFC 2462, December 1998.

CONCLUSION AND FUTURE WORK

This work represents the first part of an effort to incorporate the notion of domains into autoconfigured adhoc networks. Our implementation proves that the mechanisms can be incorportated into existing IP autoconfiguration protocols with little additional complexity, delay or bandwidth. Additional research is needed to measure how much domain autoconfiguration can improve the performance of large adhoc networks, whether in terms of robustness, scalability, throughput, security, or other measures. We must not only design decision/optimization processes, but also decide kind of information we need to collect from the network in order to reach into the optimal domain decisions quickly[i].

[i] The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government