Dynamic Network Renaming using Space-Time Contexts

Venkata K. Pingali
USC Information Sciences Institute,
4676 Admiralty Way,
Marina del Rey, CA
pingali@isi.edu

Joseph D. Touch
USC Information Sciences Institute,
4676 Admiralty Way,
Marina del Rey, CA
touch@isi.edu

Abstract— **This work suggests extending namespaces in protocols such as IP in layered networks with a context of interpretation. These contexts allow conflicting distributions of names within a namespace to overlap in space and time. By altering what is discovered, i.e., context in addition to name, and decoupling the discovery process from the rest of the communication, space-time contexts reduce disruption during renaming in layered networks. The expressiveness of the layered network in terms of control is also simultaneously enhanced enabling a variety of advanced network management capabilities to be built.**

Keywords-component: Communications Applications, Distributed systems

I. PROBLEM OF RENAMING

Names such as IP addresses must be discovered by protocol nodes in a network in order to communicate i.e., construct message headers and determine the availability and route to the destination. A browser (Figure 1), for example, must discover the IP address or domain name of the webserver before it can communicate. The discovery process is expensive in terms of computation and time and often involves a variety of non-trivial single and multi-hop protocols such as DNS and other directory services, and out-of-band mechanisms such as human. The discovered names are stored in the network at endpoints in configuration files, directories such as DHCP and DNS, and control nodes such as firewalls, often in an uncoordinated way. These stored names create information dependencies across protocol nodes (Figure 1).

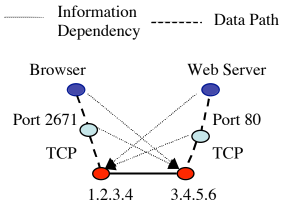

Figure 1 Dependencies created by storage of names

Renaming protocol nodes, e.g., changing the IP address of a host, has two consequences. First, the ongoing communication is disrupted e.g., TCP connections are terminated. Second, the information dependencies fail forcing rediscovery of names. The browser must rediscover the IP address of the webserver, for example. While the protocol state and messaging can be restored using well-known techniques such as sessions and transactions, the discovery issue has not been adequately addressed. The duration of disruption due to discovery could range from few milliseconds to hours depending on spatial scope of the impact of the renaming and the nature of the discovery mechanism used. Often the discovery processes themselves are impacted by the change. Routing may be disrupted or specific discovery nodes may be unreachable. Errors in discovered names may result in sink states from which the host may not be able to recover. In addition to correctness aspects, the performance requirements also increase with scale of the network, frequency of changes, and cost of disruption. Making the discovery process aggressive results in additional complexity in the network.

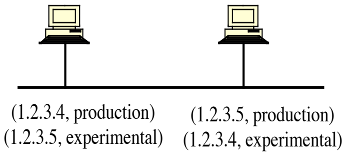

Figure 2 IP extended with contexts

II. SPACE-TIME CONTEXTS

The objective of space-time contexts is to extend the validity of the discovered names across renaming operations. Each name n in any namespace N is associated with a space-time context C, that identifies the space-time region of validity of n. Examples of context names include sequence numbers, version numbers, and descriptive strings such as "production" and "experimental" (Figure 2). Protocol state machines, messages and data-structures are extended to incorporate the context. For example, endpoints are specified in messages using the two-tuple (C, n). New protocols are introduced to manage contexts across hosts and integrate them using inter-context routing mechanisms. The performance and correctness requirements of these protocols are, however, much weaker than those for the discovery protocols in unmodified network.

Renaming operations create new contexts and names within instead of replacing names. The old context, e.g., "production", and names within are still valid while new contexts such as "experimental" are created, used and destroyed. As a result, ongoing communication that uses a context and name combination can continue with little or no disruption. In effect, contexts enable decoupling of the time of name creation from the time of their discovery and use, and therefore remove it from the critical path of renaming. Non-disruption has value for large enterprise networks that have very high disruption costs and/or zero downtime requirements such as the stock exchanges, global logistics organizations and defense establishments.

III. ISSUES

The introduction of the contexts results in an explosion of information. At any point in time, several contexts may be active, and a protocol node might have names from multiple contexts (Figure 3). The routing information required to route to every name and context combination is prohibitive and useless. Contexts are useful only when used in coordinated way across nodes, layers and hosts. Several new mechanisms may be therefore required for (1) basic support for contexts, (2) organizing the contexts to reduce the information that any node must know, (3) automated management of the contexts, (4) cross-context communication, and (5) integration with legacy networks. Various approaches for each of them are being investigated.

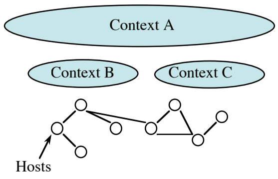

Figure 3 Distribution of contexts across hosts

As mentioned above support for contexts is required with protocol state machines, data structures and messages. The exact approach depends on the details of the protocol involved. In case of protocol messages a context may be represented as protocol option, within an extension header or in a protocol shim layer. Within the host, support for context-specific data structures may require selection from existing isolation capabilities, introduction of a level of indirection or a complete reimplementation.

While contexts by themselves do not carry semantics, it might be useful to relate contexts in terms of topology, time, performance, cost and nature in order to reduce routing information and frequency of the discovery. Contexts may be organized, for example, as a directed acyclic graph where nodes are contexts and edges indicate the “inclusion” relationship between contexts.

Automated management of space-time contexts requires a combination of the ability to dynamically compose protocol modules within a host, and distributed algorithms to coordinate that composition across hosts. We identify three building blocks, each of which addresses a different part of the problem: flexible compositional framework within the hosts, association of context-specific state across hosts, and cross-host coordination to support context-level group operations.

Communication can cross context boundaries, i.e., the source can be in one context and the destination is in another context. This is similar to forwarding of IP packets across LANs. Similar to IP forwarding, at any point in time the locus of control remains within one context except during the brief interval when the inter-context forwarding lookup is performed. Routing requires knowledge about the existence and location of other contexts and efficient paths to them. This information is propagated between nodes using a space-time inter-context routing protocol. The key aspects in the protocol include (1) association between context-specific state across nodes, (2) rules when nodes can exchange context information, (3) the path information that must be exchanged (4) forwarding algorithm at each node, and (5) extended message header structure.

IV. EXPRESSIVENESS OF SPACE-TIME CONTEXTS

Non-disruptive renaming enables other advanced capabilities that aim to modify the network in specific ways. Each step of the renaming process including deployment, discovery, selection and integration is a design space that will allow multiple point solutions that can be composed in various ways. Some of the new capabilities that will be enabled, atleast partially, with space-time naming include (1) automatic and self-configuration that involve a search for appropriate configurations, (2) market mechanisms for protocols that require low switching costs and easy integration, (3) advanced testbeds that allow for experimentation on a production infrastructure, and (4) agile information infrastructure that can be modified to reflect changing needs. Additional mechanisms are required to handle other complementary aspects such as data consistency, network-wide discovery, and balance the contradictory needs of stability and change. Beyond minimally disruptive renaming, contexts allow us to relate names from past, present and future, and separate the control path from data path in time without altering the basic structure of the Internet. These qualities may be combined in various ways to create advanced network management capabilities.

V. STATUS

A prototype is currently under development to investigate various aspects of the space-time contexts. The prototype extends IPv6 with contexts. Clonable stacks [1] is used to support isolation between context-specific state within the kernel. A new IPv6 routing header and associated forwarding table is used to achieve inter-context forwarding. The integration between intra-context and inter-context forwarding is primitive. Automated deployment of the context is achieved through a coordinated invocation of host-local context-management primitives. This coordination is provided through a simple distributed consensus algorithm [2].

REFERENCES

[1] M. Zec, "Implementing a Clonable Network Stack in the FreeBSD Kernel," in Proceedings of the 2003 USENIX Annual Technical Conference, FreeNIX Track, June 2003, San Antonio, TX.

[2] G. R. Andrews, "Paradigms for Process Interaction in Distributed Programs," ACM Computing Surveys 23:1, March 1991, 49-90.