Computer Networks 55 (2011) 919–935

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A Dynamic Recursive Unified Internet Design (DRUID)

Joe Touch [a,*], Ilia Baldine [b], Rudra Dutta [d], Gregory G. Finn [a], Bryan Ford [e], Scott Jordan [f], Dan Massey [g], Abraham Matta [c], Christos Papadopoulos [g], Peter Reiher [h], George Rouskas [d]

[a] USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292-6695, United States
[b] Renaissance Computing Institute (RENCI), North Carolina State University's Centennial Campus, Partners I Building, Suite 1500 and Engineering Building II, Room 1235, Raleigh, NC 27606, United States
[c] Computer Science Department, Boston University, Boston, MA 02215, United States
[d] Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, United States
[e] Department of Computer Science, Yale University, 51 Prospect Street, New Haven, CT 06511, United States
[f] 3019 Bren Hall, Department of Computer Science, University of California, Irvine, Irvine, CA 92697-3435, United States
[g] Colorado State University, Department of Computer Science, 1873 Campus Delivery, Fort Collins, CO 80523, United States
[h] 3564 Boelter Hall, UCLA, 405 Hilgard Ave., Los Angeles, CA 90095, United States

ARTICLE INFO

Article history:
Available online 24 December 2010

Keywords:
Network architecture
Future internet
Recursive networks
Dynamic stacks

ABSTRACT

The Dynamic Recursive Unified Internet Design (DRUID) is a future Internet design that unifies overlay networks with conventional layered network architectures. DRUID is based on the fundamental concept of recursion, enabling a simple and direct network architecture that unifies the data, control, management, and security aspects of the current Internet, leading to a more trustworthy network. DRUID's architecture is based on a single *recursive block* that can adapt to support a variety of communication functions, including parameterized mechanisms for hard/soft state, flow and congestion control, sequence control, fragmentation and reassembly, compression, encryption, and error recovery. This recursion is guided by the structure of a graph of *translation tables* that help compartmentalize the scope of various functions and identifier spaces, while relating these spaces for resource discovery, resolution, and routing. The graph also organizes *persistent state* that coordinates behavior between individual data events (*e.g.*, coordinating packets as a connection), among different associations (*e.g.*, between connections), as well as helping optimize the recursive discovery process through caching, and supporting prefetching and distributed pre-coordination. This paper describes the DRUID architecture composed of these three parts (recursive block, translation tables, persistent state), and highlights its goals and benefits, including unifying the data, control, management, and security planes currently considered orthogonal aspects of network architecture.

1. Introduction

The Dynamic Recursive Unified Internet Design (DRUID) is a future Internet architecture based on the repeated use of a single, flexible functional unit for different capabilities over different scopes of a communication service. DRUID allows common protocol functions and capabilities to be reused from within a single block, avoiding the need for recapitulated implementation, and allowing these functions and scopes to be dynamically determined, enabling the service to adapt to changes in the local machine and

* Corresponding author.
E-mail addresses: touch@isi.edu (J. Touch), ibaldin@renci.org (I. Baldine), dutta@ncsu.edu (R. Dutta), finn@isi.edu (G.G. Finn), bryan.ford@yale.edu (B. Ford), sjordan@uci.edu (S. Jordan), massey@cs.colostate.edu (D. Massey), matta@bu.edu (A. Matta), christos@cs.colostate.edu (C. Papadopoulos), reiher@cs.ucla.edu (P. Reiher), rouskas@ncsu.edu (G. Rouskas).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.12.016

network context. It also unifies many different aspects of networking, providing a single architecture to integrate the data, control, network management, and security planes in a single, coherent approach. DRUID allows tremendous flexibility and extensibility in network behavior and functionality, while simultaneously maintaining a simple unified architecture.

DRUID explores the impact of layering and scoping on network architecture. It is composed from a single *recursive block* together with a graph of *translation tables* representing relationships between different scopes in the network. The recursive block includes both code and data, and as it recurses, guided by this graph, it refers to and modifies *persistent state*, so these simple components can support a wide range of communication services. Paths through the graph of these tables – both at end systems and intermediate nodes, including routers, tunnel boxes, and NATs – can adjust, *e.g.*, in reaction to DoS attacks, when local or intermediate node resources change, or in reaction to network path properties.

DRUID applies the concept of recursion as a fundamental network primitive, unifying aspects of USC/ISI's RNA [56,58] and BU's RINA [44] projects, and augmenting them with more detailed description of the impact of the naming hierarchy on the recursive architecture, and discussing how this approach is related to the first principles of multiparty communication. DRUID also explores the relationship of resource discovery, routing, forwarding, and layering as aspects of a single, unified approach. This provides an opportunity to unify the data, control, management, and security planes, allowing one architecture to support coordinated transport state control, network monitoring and management, and reaction to attacks, and to integrate stateful associations at different scopes, *e.g.*, allowing end-to-end streams (*e.g.*, TCP) to easily map onto subpath streams (*e.g.*, wavelength lightpaths). DRUID further affords an opportunity to explore more dynamic service composition and service adaptation, allowing composed protocols to react to local resources, network resources, policy, economics, and threats.

DRUID's approach helps provide a basis for potentially new understanding of multiparty communication, and for integrating many extensions currently considered artificial or external. It provides a coherent, unifying view of networking, supporting the coordination of the data, control, management, and security planes rather than considering them independently. Whether successful as a replacement to the Internet or not, it represents a unique opportunity to impact the community's view of network architecture as more than mere archaeology (studying implementation artifacts), driven by a concept core to the basis of computer science – recursion.

The remainder of this paper presents the motivation for a recursive architecture in Section 2, the architecture itself in Section 3, including how it addresses trust, a key deficiency in the current Internet. Section 4 discusses issues and challenges in realizing the architecture, and Section 5 presents other discussion. Section 6 summarizes the current implementation status, which focuses on our individual current projects. Our future plans for the implementation are discussed in Section 7, and some related work on which DRUID is based is presented in Section 8.

2. Motivation

DRUID is partly motivated by some fundamental observations about multiparty communication [61]. Consider first the standard Shannon two-party communication channel model. In this model, the exchange of data between two endpoints is typically characterized in terms of the channel error and encoding overhead. The model and its descendents derive numerous properties about such a two-party channel, but this is of little direct relevance to network architecture because the two endpoints are considered known *a priori*. The most challenging part of network architecture – knowing who you want to talk to – is removed from the model by the initial conditions.

When going beyond two parties, multiparty communications are driven by three properties – the heterogeneity of the parties, the potential that any subset might want to communicate, and the dynamics of the ways in which communicating parties associate. Given a set of M heterogeneous endpoints, communication either requires $O(M^2)$ translators, or M translators that all support a common interchange format, as shown in Fig. 1.

In either case, at some layer the data is converted between a local representation and one used to reach the destination that requires a different format, so as data traverses between layers, it needs to be converted, not only in the common interchange format of the layers, but also between the names or identifiers available at a given layer (Fig. 2). Layering itself thus leads to the need for a resolution mechanism.

All parties might want to interact with each other, but just as it is not feasible to have M^2 translators, it is not always feasible to assume there are $O(M^2)$ direct links. Supporting arbitrary pairwise communication thus requires some sort of forwarding as well (Fig. 3). RNA demonstrated that forwarding is equivalent to tail recursion [56].

Finally, the desire to vary the association of groups dynamically itself leads to recursion, because each group can easily be considered a recursive component of the larger set, as a virtual subset (Fig. 4) [15,56]. Such recursive virtualization was explored in the X-Bone [53], and is currently a fundamental part of several emerging extensions to the Internet, including Rbridges (TRILL in the IETF [60]) and LISP (in the IETF [21]).

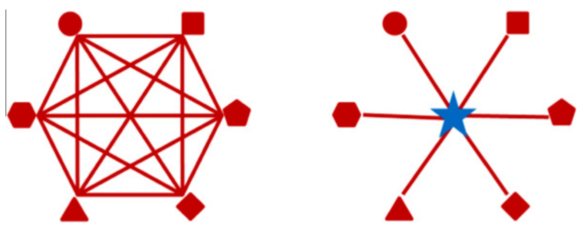

Fig. 1. Heterogeneity of parties leads to $O(M^2)$ translators (left) or a layer of M translators (right).