論文内容の要旨

論文題目　**Design and Implementation of the SORA Multipath Virtual Network Layer**

（**SORA** マルチパス仮想ネットワーク層の設計と実装）

氏名　　レーン　ジョン　ラッセル

It has been shown that a number of shortcomings exist in the interdomain routing regime of the current Internet which hinder its growth as well as the implementation of reliable, high performance services using it.　These include poor control over interdomain traffic (e.g., selection of AS paths is limited to the options provided by neighbors), slow recovery in the face of faults (e.g., three minute average and 15 minute worse case convergence times), poor isolation of faults from one part of the network to another (e.g., a misconfiguration error anywhere in the network can cause a routing black hole), error-prone configuration and management (e.g., non-convergent configurations are generally possible), an inability to respond to differing application requirements (e.g., one-size-fits-all routing, where neither application, end-system or even end-network has much input on the path a packet is to traverse) as well as economic issues such as an inability for content producers to receive payment for the bandwidth content consumers consume when accessing their services, making many business models difficult or impossible, and an inability for ISPs to always be able to match user network charges with user traffic usage rates.

It has been demonstrated that many of these shortcomings have their roots in: (1) the Internet's conflation of control and data planes, (2) the Internet's dependence upon convergence and hop-by-hop routing and (3) the Internet's current payment structure.　The conflation of control and data plane refers to the fact that nearly all forwarding decisions (e.g., best path decisions) are made by the very machines that perform the actual forwarding—the routers themselves.　This tends to force the network to use hop-by-hop routing and distributed best-path computation protocols that require convergence.　Hop-by-hop routing makes it difficult for network operators to engineering traffic or set complex routing policies.　Moreover, the dependence on convergence leads to slow recovery times

because the result of each round of the best-path computation must be computed and redistributed. Finally, the problems described above with the current economic regime have been explained by its failure to allow payment for network services to flow along the entire path that data traverses, which in turn can make it difficult for large content providers to engage in certain business models. Moreover, the current regime's lack of a fine-grained billing model allowing per-volume payment has been linked to current ISP traffic throttling issues.

Multipath routing has appeared in many forms in system proposals to address all of these issues. Examples include, ATM, IPv6 (via enablement of source routing), Nimrod, 4D, RCP, MIRO, pathlet routing and MPLS as well as path brokering, to name only a few. Multipath routing has been proposed to address many of these issues by exploiting multiple paths for better performance, reliability, as well as by providing an economic primitive for Internet network commerce (i.e., a path or "pathlet"), thereby enhancing competition by allowing payment to be performed along all path segments (not just to the ``network core'' or tier-1 operators). However, despite the promise multipath solutions have been said to have in meeting future Internet requirements and the centrality which multipath routing may therefore play in any future Internet, there is a paucity of tools for testing new multipath-aware transport protocol and application implementations. Without the ability to test new multipath approaches, it is not possible to guarantee their efficacy or efficiency. Moreover, while many testbeds already exist to provide general experimentation services to researchers (e.g., Planetlab, Onelab, CoreLab, Emulab, etc.), there still exists no general framework for using multipath-specific features with them or for pooling the network resources they collectively may offer. Moreover, while new network testbeds are currently being constructed (e.g., GENI) specifically to enable research into next-generation network issues, research into architectures that enable the study of multipath routing and its approaches is lacking. An increasing number of resources to study next-generation network issues do exist, however, frameworks and research tools for using these resources to study multipath routing issues as well as for comparing the results obtained from them currently do not exist.

This dissertation proposes a novel multipath virtual network architecture called SORA as a means for meeting this shortcoming in the network research community's ability to develop and test new multipath networking approaches as well as next-generation protocols. It also presents research findings, obtained using SORA, into a next-generation network service for supporting legacy networks that use multipath facilities.

SORA is a multipath virtual network architecture and associated API for implementing multipath-aware transport protocols and applications. It extends recent research regarding a clean control and data plane split with forwarding decisions not necessarily made on routers, but anywhere in the network. This dissertation describes SORA's design as well as its core API elements; it also provides key details on a prototype that has been constructed and an analysis of SORA's design as well as the performance of the prototype.

SORA features a control plane API which provides for the creation and management of virtual networks—logical networks that are created atop existing networks (such as IP) and effect transport using packet encapsulation between virtual network routing entities (e.g., endhosts).

SORA's control plane API features three basic types of virtual network abstractions: (1) *attributes*, (2) address *mappings* and (3) *links*. Attributes are a method of storing short text-based configuration information about a specific virtual network instance. Address mappings are abstractions for mapping the multiple physical addresses of an entity to a single identifier of a virtual network creator's choosing—simultaneously allowing network creators to use their own addressing scheme while also allowing a single identifier for the given entity on the created virtual network. Links are abstractions for representing various means of sending packets between two virtual network routers (e.g., UDP/IP tunneling, MPLS, etc.) and their attributes (e.g., loss rate and latency). These three types of abstractions comprise a virtual network configuration. Configuration instances are stored in a *state database service*, for which the API also provides methods for management, access and implementation. Paths through SORA virtual networks are comprised of series of virtual network links; paths are computed via a *path computation service*, for which the API also provides methods for query access and path computation engine implementation.

The current control plane prototype is implemented in an object-oriented scripting language for extensibility and rapid development. The prototype control plane implements the state database service via one or more relational databases, which are accessed via RPC. It implements the path computation service using RPC in conjunction with a C++-based module, which offers fast path computation within the run-time environment of the scripting language.

SORA features a data plane API that enables simultaneous forwarding for multiple virtual networks via a SORA packet header which contains a network identifier identifying a given virtual network instance and an extensible set of packet options. One important option is a *path*, which is specified as a list of link identifiers; the path option allows end-systems (and their applications) to select paths while eliminating the need for virtual network routers to run distributed routing computations for each virtual network or otherwise perform forwarding decisions regarding the best path for a given packet. Finally, SORA features a novel paradigm for multipath programming in its conduit endpoint interface, which acts as a convenient basis for the implementation of new multipath services.

The current data plane prototype API, `libsora` is implemented in C/C++ and provides methods for packet processing, path management as well as an interface to query respective control plane services for required attribute, link and path data. Path management features include path set selection, path monitoring and feedback of monitored data to senders via packet options. Furthermore, a SORA router and virtual network ingress for unmodified IP endhosts was implemented within the Click modular router framework. Click is a framework for constructing routers from small packet processing elements; elements implementing SORA functionality were created atop `libsora`. The forwarding performance of the data plane API implementation in terms of sustained throughput has been found to be just slightly (6-7%) less than what is currently possible with routing via the Linux kernel using off-the-shelf hardware, the small gap in performance relative to Linux being largely due to reduced utilization caused by the use of encapsulation and the SORA packet header. A prototype implementation of the SORA control and data planes has been deployed on Planetlab and Emulab and used to obtain experimental results.

Architecturally, SORA logically resides between the transport layer and existing network technologies such as IPv4 and IPv6 and exists to provide enhanced, experimental routing services. It is for this reason and because it implements its own forwarding control plane (e.g., virtual network setup and management and path computation) and data plane (e.g., packet forwarding via the SORA header), that SORA is referred to as a multipath network layer. SORA is a tool for network researchers wishing to implement their ideas and run them on actual networks. It allows researchers to quickly setup and tear down virtual networks on shared testbeds as well as to pool resources between testbeds. SORA also represents a large-scale experimental architecture and prototype featuring full path selectability, end-host-based path selection, which allows applications to have more control in forwarding decisions, and a clean control plane/data plane split.

In addition to a description of SORA's design and prototype, this dissertation describes research conducted using a SORA prototype and presents results obtained from it. Specifically, the research described investigates the feasibility of using a *network layer packet reordering service* to mitigate poor TCP performance in the presence of varying degrees of out-of-order packet delivery. Packet dispersion—the striping of packets from the same flow over multiple paths simultaneously—is adopted as a highly typical example of aggressive multipath use; a packet reordering service is implemented using SORA such that *no endhost modifications are required*, thus allowing the study of TCP *in situ*—using well-known measurement tools. While previous work on the subject only simulated out-of-order delivery at the ingress, did not characterize the type of reordering observed and only tested one TCP variant, these shortcomings are addressed and results are presented from experiments to test the performance of two common TCP variants under packet dispersion with differing numbers of paths and amounts of inter-path latency variance, as might be expected from use of packet dispersion on a wide-area network. Results presented herein demonstrate that it is possible to insulate unmodified TCP from the effects of packet reordering through the use of a simple reordering service.

In summary, SORA represents a significant contribution to the network research community in the following ways. First, it enables simultaneous, large-scale multipath experiments to be performed more easily than ever before and using network resources from a differing testbed environments, which has not heretofore been possible. Second, by the combination of its control plane, which allows control and management of virtual network topologies as well as its data plane API and conduit endpoint interface, SORA provides a generalized infrastructure and platform for multipath transport protocol and application development and testing, which has, heretofore not been available. Finally, SORA represents a design and prototype implementation of a novel network architecture featuring end-host path selection and a clean split between control plane and data planes—one allowing forwarding decisions to be made not at the routers but at the end-hosts, thus offering applications and transport protocols more control over routing.