TRIAD:
A New Next-Generation Internet Architecture

Computer Science Department
Stanford University
{cheriton,mgritter}@dsg.stanford.edu

Abstract

Today, the primary use of the Internet is content distribution — delivery of web pages, audio and video streams to client browsers. However, scaling to meet the enormous demands of the web have required *ad hoc* and, in some cases, proprietary protocols and mechanisms to be deployed. Unfortunately, these *ad hoc* mechanisms have scaling problems and conflict with the original Internet architecture. IPv6, the current leading candidate for a next generation Internet architecture, provides more addresses but does not help with the content problem, given that its design predates the web.

In this paper, we present TRIAD as a new next generation architecture. A key aspect of TRIAD is the explicit inclusion of a *content layer* that provides scalable content routing, caching and content transformation. TRIAD also provides extensible path-based addressing using a simple "shim" protocol on top of IPv4. We claim that TRIAD not only provides scalable content distribution, but also solves the Internet problems with supporting *network address translation (NAT)* and provides innovative solutions to mobility, virtual private networks, policy-based routing and source spoofing. Its compatibility with IPv4, TCP, DNS and other dominant Internet protocols facilitates incremental deployable.

1 Introduction

With the emergence of the web, the primary use of the Internet is content distribution, i.e. web pages, and increasingly audio and video streams. Some measurements indicate that 70 to 80 percent of Internet traffic is HTTP traffic (and most of the rest of the traffic is routing and DNS). That is, almost all of the traffic in the wide area is delivery of content, and ancillary traffic to locate it and determine how to deliver it. Today, millions of clients are accessing thousands of web sites on a daily basis, with the top 20 web sites supplying about 10 percent of the content. Moreover, new popular web sites and temporarily attractive web sites can prompt the arrival of a so-called *flash crowd* of clients, often overwhelming the resources of the associated web site.

To scale content delivery to support these demands, a variety of *ad hoc* and, in some cases, proprietary mechanisms have been deployed. For instance, content is geographically replicated at multiple sites with specialized name servers that redirect DNS lookups to nearby (to the client) sites based on specialized routing, load monitoring and Internet "mapping" mechanisms, so-called *content routing*. Further, proxies in the network provide transparent caching of content, further reducing the load on web sites and the network resources between the cache and the web site. Finally, load-balancing switches at each web site allow the *virtual host* for the web site to be realized as N physical hosts, distributing the content requests across these hosts based on individual server load and the content it holds. Future extensions are expected to provide content transformations proxies that will transform content to a representation suitable for the requesting client, such as a mobile PDA.

These mechanisms violate the original Internet architecture in various ways, do not fully address scalable content distribution, and compromise the basic philosophy of the Internet as being based on open, community-based standards. In particular, content routing requires a DNS server that accesses some form of routing information, a layer violation (or duplication of the routing layer), yet still requires the world-wide clients of a site to access a single centralized server as part of accessing that site. This roundtrip to a central server is quickly becoming the dominant performance issue for clients as Internet data rates move to multiple gigabits, reducing the transfer time for content to insignificance. Transparent caching requires hijacking transport-level connections, violating the end-to-end semantics and causing connection failures when the routing changes to route around the cache. Moreover, these caches introduce extra delay in connection setup for non-cachable content while they collect the HTTP header information required to determine whether the content is cachable or not. Load balancing switches rely on *network address translation (NAT)* which requires rewriting addresses, port numbers, transport-layer checksums and even packet data, making mockery of the original end-to-end semantics. (Network Address Translation [11, 12] (NAT) is also being widely deployed for other reasons, such as *address allocation autonomy*, multi-homing, concealing endpoints and increasing the number of addresses available in an edge network.

In 1992, work on IPng, next-generation IP, was initiated based on concern that the Internet was running out of addresses. This effort, predating the web, focused on providing more addresses but failed to anticipate the strong emergence of content as the primary use of the web and thus failed to address the content distribution issues described above.

In this paper, we present TRIAD[1], as a new next generation architecture. TRIAD defines an explicit *content layer* that provides scalable content routing, caching, content transformation and load balancing, integrating naming, routing and transport connection setup. In particular, all end-to-end identification in TRIAD is based on names and URLs, with IP addresses reduced to the role of transient routing tags. At this content layer, The integrated directory, routing and connection setup to provide efficient content routing to replicated content and eliminate roundtrip times to access the content in most cases. Finally, TRIAD supports path-based addressing using a simple "shim" protocol on top of IPv4 called WRAP, for content routing control and extensible addressing. This extensible addressing eliminates the need to transition the Internet to IPv6. We claim that TRIAD not only provides scalable content distribution, but also solves the Internet problems associated with *network address translation (NAT)*. TRIAD also provides attractive solutions to mobility, virtual private networking, policy-based routing and source spoofing. TRIAD can be incrementally deployed, initially without changes to end-hosts or applications beyond that already required for NAT.

The next section describes the TRIAD content layer. Section 3 describes the named-based identification of endpoints and its implications in more detail, illustrating with the associated modifications to TCP. Section 4 describes the integrated naming and routing in TRIAD. Section 5 describes the extensible path-based addressing using WRAP, its expected implementation and WRAPsec, an IPsec-like mechanism for end-to-end security. Section 7 discusses the implementation of TRIAD and some measurements to evaluate it we have made to date. Section 6 describes the TRIAD approaches to mobility, VPNs, policy-based routing and extended forwarding checks. Section 8 describes an extended NAT router that allows incremental deployment of TRIAD. Section 9 describes the incremental deployment of TRIAD. Section 10 describes related work, and we close with conclusions.

2 TRIAD Content Layer

At the content layer, a client either requests to access some content, identified by some name specification, to read or to write (or both). For example, a client may request the CNN news, which is delivered as some combination of web pages and audio and video streams. In contrast, lower levels of the architecture simply transmit or receive data packets.

The TRIAD content layer may return a *network pointer* through which content may be read or written, rather than the content itself, especially when the content is large or indeterminate in length.

For compatibility with the World-wide Web, the TRIAD content layer uses the Web Uniform Resource Locator (URL) as the format for content identification, optionally augmented with web "cookies". Further, this "network pointer" is realized as an HTTP/TCP connection through which the content is read or written.

The content layer is implemented by *content routers* that direct the requests towards content servers storing the content, *content servers* that provide the content services, and *content caches* that transparently store some content nearer to the client than the servers themselves. It may also include *content transformers* that transform the content from one form to another in response to characteristics of the client and its network connection. For instance, a content transformer may reformat content for presentation on a PDA, and transmission to this PDA over a limited bandwidth wireless link.

[1]TRIAD is an acronym, standing for *Translating Relaying Internet Architecture integrating Active Directories.*

For example, in TRIAD, a client sends a lookup request with a URL for the CNN news page to its content router. The content router looks up this URL (typically just the DNS portion of the URL) and determines a nearby replica for this content, forwarding the request to a next hop accordingly. The next hop content router looks up this URL as well and may determine it has this content cached locally. In this case, it returns TCP connection information to the client, allowing the client to read this content from its cache over this (HTTP/TCP) connection.

In this way, the content layer explicitly supports the key requirements for scalable content distribution. In particular, it explicitly supports replicated content, leveraging the routing information to locate the nearest replica. It further provides hierarchical transparent caching, minimizing the implosion of demand on popular content servers and their Internet connections without requiring explicit configuration of proxies in the browsers while still avoiding the "route-around" danger with current transparent caches. Finally, it eliminates a roundtrip relative to the current Internet by combining the name lookup with connection setup.

The TRIAD content layer also supports multicast-based content distribution, by allowing a content lookup to subsume the functionality of the current multicast subscription, connecting client into the multicast session and the associated distributed tree.

For simplicity, one can view the DNS portion of a URL as mapping to the content server and the file name portion selecting the file within the content server. However, there is some flexibility in this partitioning of a URL into DNS name and file name portion. For instance, a prefix of the DNS name can determine a content volume within a specific content server. Similarly, a content router can direct an HTTP connection to a separate server based on the file portion. The primary constraint now is that the file portion is not available until the HTTP connection is established. For simplicity, we describe TRIAD assuming the conventional use of DNS name specifying the content server/volume, and file name portion specifying the content within this volume.

For deployability, TRIAD is designed to be highly compatible with existing Internet infrastructure, requiring extensions primarily at the directory system level. The TRIAD content layer subsumes the DNS directory system, providing DNS name lookup as a subset of its services. It also includes the wide-area routing system, allowing it to map content name to the closest replica based on the proximity information (accessed from the routing system). Finally, the content layer includes a mechanism for creating the so-called network pointers, more conventionally known as transport connections, thus subsuming the connection setup portion of TCP. The TRIAD content layer interfaces to the conventional transport and (inter)network layers of the Internet, allowing IP and TCP to be used, but with some modifications.

The following sections describe specific changes in more detail.

3 Named-based Identification Endpoints

In TRIAD, endpoints are identified by name. At the content level, the endpoint is a source of content and is identified by a URL. At the host interface level[2] the endpoint is identified by a hierarchical character-string (DNS) name. For example, "pescadero.stanford.edu" identifies a host interface on a host at Stanford. In contrast, an address for this interface may change over time, even during the lifetime of transport layer connections.

This *name* is the basis for all end-to-end identification, authentication, and reference passing. There is no other identifier that is global and persistent, unlike addresses in IPv6 and in the original Internet architecture. (IPv4) addresses serve as *routing tags* and have no end-to-end significance.

A multicast channel as in the EXPRESS single source multicast (SSM) model [4] is identified by a name subordinate to the name of the source host. For example, "chan1.pescadero.stanford.edu" could be the name of a multicast channel with "pescadero.stanford.edu" as the source. (TRIAD only supports the single-source model of multicast.).

3.1 Name-based Transport Connection Setup

In TRIAD, a transport connection is setup by a name lookup and connection setup protocol at the content layer called *Directory Relay Protocol (DRP)* [5]. The client sends a lookup request containing a URL and possibly other information, such as "cookies" to the content layer's directory system. For brevity, we refer to this information generically as the "name", rather than detailing URL and cookies each time. The directory

[2]Packets are sent to a host interface, not to a host, the same as the original Internet architecture.

system routes the name request through to a content server able to provide this named content, if any, which then either returns the content directly or else sets up transport connection state and returns the associated connection information to the requesting client.

DRP serves as a connection setup protocol for TCP, carrying the same sequence number, port and option information as a TCP SYN packet along with the content identification. The content identification is stored with the connection state and is used to re-bind the connection to new addressing if the connection is failing to get packets through. In the expected case, this rebinding maps to the endpoint that is storing the transport-level state of the connection, allowing the transport-level connection to continue with the new address binding, similar to an ARP-level rebinding. In particular, the rebinding uses the canonical name for the content server so it first attempts to rebind to the same replica. When the original content server is not reachable, a end-to-end name lookup can bind to a new server, restarting the content access from the beginning.

Using DRP, a TRIAD TCP connection to content can be established with a single round-trip from client to server. This contrasts with the two roundtrips commonly required with the current DNS/TCP separation. Moreover, a transparent content cache can intercept a DRP request and service this content request without having to hijack the transport packets because, at the packet-level, the packets are addressed to the transparent cache. This avoids the connection failures that can occur currently when packets are rerouted around a transparent cache. The transparent cache can also receive all the HTTP information in the DRP request, eliminating the need to incur an extra roundtrip to get this information, as with the current TCP/HTTP setup behavior. Also, TRIAD TCP can transparently recover from changes to the addresses used to reach the endpoints, whether caused by intermediate node timeouts or reboots or link failures (assuming an alternate path is available). This rebinding makes the translation in the network effectively *soft state*, preserving one of the key properties of the Internet. UDP-based services are expected to similarly rebind from the name, either periodically or on timeout, in the case of a reliable protocol built on UDP. Finally, DRP carries the client name and identification, allowing the content server to determine this information without callback or reverse name lookup. (The network can validate the DNS-level identification as part of forwarding the request.)

This same name-based connection setup with DRP works for joining a multicast session as well. The name lookup is sent to the source, which returns the required information to join the multicast session, setting up requisite multicast state at the network layer along the path.

For backwards compatibility with current Internet, TRIAD hosts also support (and fail-over to) conventional TCP connection setup when communicating with unmodified IPv4 hosts.

3.2 Name-based Transport Pseudo-Header

In TRIAD, the transport layer checksum covers the name of the source and the name of the destination and *does not* include the packet address. In the case of multicast, the pseudo-header is based on the name of the channel. The addressing allows efficient forwarding of the packet to a destination; the name-based pseudo-header ensures it arrived at the correct destination (even though the names are not present in the header). Thus, a transport connection is between two endpoints identified by name, not address.

A transport-level checksum based on this pseudo-header provides end-to-end reliability because it detects corruption of the packet addresses in transit yet does not need to be modified in transit as part of relaying (or forwarding) the packet even though the packet addresses are modified.

This name-based pseudo-header checksum also allows end-to-end security with NAT because changing the addresses does not require updating the checksum. TRIAD includes a security layer similar to IPsec, WRAPsec, which uses names for identification and the same pseudo-header for integrity check verification (ICV), and provides end-to-end security. Note that dispatch to connection state before validating the checksum is required in TRIAD for both secure and insecure connections, unifying the processing in both cases. With conventional TCP implementations, the checksum is often checked before mapping to the associated TCP connection state.

On connection setup, the local endpoint computes and saves a *precomputed pseudo-header checksum value (PHCV)* based on the name of the source and the destination, similarly at the remote endpoint. The source and destination names are also stored in the connection state record with the PHCV. When a packet is transmitted, the checksum is computed starting with the PHCV, effectively incorporating this name-based pseudo-header into the packet checksum. On reception, the packet is demultiplexed to the TCP connection state based on the packet addresses and port numbers. This receiving connection state contains the same PHCV, allowing the receiver to (re)compute the packet checksum efficiently. If the computed checksum fails to match that in the packet, the packet is considered corrupted in transmission.

If the packet does not map to a valid connection state, the receiver does a reverse name lookup to determine the source name and looks up the connection by name. If the name lookup fails, the packet is discarded as a corrupted packet. An endpoint may receive a packet for an existing connection that does not match based on addressing (perhaps because of a reboot of an intermediate node). The name-based mapping allows the connection state to be located and the address mapping to be corrected.

To allow connection setups with minimal modifications to current TCP clients, TRIAD-TCP includes a (new) option that carries the PHCV in the SYN packet, rather than relying on DRP. For backward compatibility, TRIAD-TCP can also use the current TCP checksum algorithm for a connection where the source and destination are in the same realm.

TRIAD-TCP provides a negotiated "unreliable" mode, which simply disables retransmissions. This minor extension to TCP as a negotiated option allows applications such as real-time VoIP and video to use TRIAD-TCP and automatically get the rebinding behavior described above (as well as the TCP congestion avoidance mechanisms). With this provision, TCP can replace the wide-area use of UDP in all cases that we are aware of. Then, UDP is only used local to a realm, if at all.

The behavior in TCP of allowing infinite timeouts when the connection is idle is supported in TRIAD by a timestamp on the name stored in the connection record, and rebinding when a connection becomes non-idle if the connection has been idle with no relookup for some excessive period of time.

TRIAD routers include support for WRAMP[3], an ICMP-like protocol for sending "destination unreachable" messages, similar to ICMP, thus informing hosts (on a best efforts basis) when the addressing is no longer valid.

With these changes, applications on WRAP-aware hosts using TCP have end-to-end semantics and end-to-end reliability, and are oblivious to loss of intermediate translating state except possibly for the performance impact. These changes to TCP do not change the packet format, are local to the implementation, and allow unmodified hosts to communicate within a realm.

3.3 Aliases and Content Routing

Content routing is supported in TRIAD by associating an *alias name* with the content. The DRP lookup of an alias name is mapped to the associated canonical name that is closest to the requesting client. (An alias can represent a host or a set of hosts.) For example, "yahoo.com" can be an alias for a set of canonical names "yahoo1.com", "yahoo2.com", etc. representing the set of content servers for Yahoo. For example, a lookup of "yahoo.com" might select "yahoo3.com", causing the DRP request to be routed towards this specific server. If this content server does not respond, the first-hop content router can select a different content server. The client can also request exclusion of a non-responsive server in a DRP request.

In this fashion, client requests are routed over the best path to the desired content in the normal case yet can recover from a failing content server. In this sense, TRIAD provides an "anycast" capability at the directory level with network and client control to reselect alternatives based on its direct experience with the chosen server. In contrast, conventional anycast mechanisms at the network layer, such as that provided by IPv6, may repeatedly route anycast packets to a server that is not functioning adequately, based on a higher-level (than network layer) evaluation.

This named-based approach relies on the name mapping being as reliable and as secure as packet forwarding, so applications can use names without loss of reliability or security. It also requires that the directory service have sufficient network routing information to determine the proximity of the interfaces identified by the alias, relative to the requesting client. This is supported in TRIAD by integration of the naming and routing. (Of course, higher-level checksumming, encryption, and authentication can provide additional security and reliability when needed.)

4 Integrated Naming and Routing

In TRIAD content layer, naming and routing are integrated in three senses. First, the router implements name lookup service, rather than a separate server that is off the data path. Second, a name lookup can

[3]The *Wide-area Relay Addressing Management Protocol*

return a route or *path specification* to the client. Finally, the routing mechanisms and directory mechanisms are strongly coupled in the router: the routing table maps from name to next-hop, rather than mapping address to next-hop, the routing information identifies endpoints and next-hop nodes by name, and name mapping information is disseminated by routing advertisements throughout the network. Individual hosts can participate in the routing by indicating the names or *content volumes* they support and the "distance" to that content, indicating the cost of reaching that content through them.

In practice, it seems unnecessary and excessive management overhead to actually have every router involved in the directory service[4]. In TRIAD, the key routers involved in the directory service are the firewall/NAT routers between realms and BGP-level routers between autonomous systems. We collectively refer to routers involved in the integrated naming and routing system by the term *content router (CR)*. We use the term *content resolving router (CRR)* for a router that just participates as a DNS resolver. This, an ISP would typically have a small number of CRs, corresponding to its current BGP routers, plus conventional routers within one realm, obvious to the content layer.

Each CR or CRR acts as a name server for each realm to which it is directly connected. For names in the same realm as the requester, the TRIAD directory service behaves exactly the same as current DNS for IPv4 clients making DNS lookup requests. That is, a DNS request with $QTYPE = A$ simply returns the IPv4 address of the associated local host, as determined by the local name database. In particular, a content router can use name lookup to locate other CRs in the same ISP (and thus presumably connected to the same ISP realm). For inter-realm lookups, the CR may return a similar local IPv4 address that it translates to the remote destination address or it may return an *address path*, as supported by the WRAP protocol. For now, consider an address path as a sequence of addresses, typically spanning several address realms, designating a loose source route to the packet destination. (Like *split DNS* in NAT, different realms have different paths associated with the same name.) This protocol is described in more detail in Section 5.

Typically, a firewall router participates as a CRR in the ISP realm to which it connects. A name lookup request it receives from a client in its private realm for a name outside of this realm is passed to a content router in the ISP, which returns an address path for a content server to the firewall. The firewall then modifies this address path information by adding its addressing information as appropriate and passes the result back to the client. Thus, each such client of the ISP behaves similar in name lookups and routing participation to that of a conventional NAT router.

4.1 DRP Implementation

A name lookup is performed by recursively relaying the DRP request across CRs from the requestor to an content server for the specified content. At each node, the name request is logically relayed along the path that packets are to take, based on local knowledge of which peer is the "next hop" towards the requested content. After the request reaches an appropriate content server, a response is returned along the reverse path through the CRs, with each one modifying the addressing to finally produce the address path suitable for the requestor to use.

By relaying the name lookup request across the same path as the packets are to flow, any necessary forwarding state can be set up in intermediate CRs. Also, this means that DNS is as available as endpoint connectivity, i.e. if the endpoint is reachable, name lookup to that endpoint works. Moreover, the trust in name lookup matches the trust in delivery because both depend on the same set of network nodes. Also, the name lookup load for a path is imposed just on the routers on that path so upgrading a router on that path in data capacity can also upgrade the name lookup capacity on that path. Moreover, transparent caches can added to a loaded path, and off-load name lookup and content delivery from subsequent portions of the path. Thus, one can balance data throughput and directory service capacity, similar to how this is handled with file systems. Consequently, in TRIAD, you can rely on names as much as you rely on addresses in the current Internet architecture, and name lookup capacity scales with the Internet.

A CR can also provide reverse (i.e., address-to-name) lookup by forwarding a reverse lookup request along the same path as a packet with the same address.

In a multi-homed realm, such as an enterprise network served by two ISPs, the internal naming and routing selects one of the CRRs for the name lookup based on local routing information. This mechanism is

[4] For instance, within one realm, any of the routers supporting a directory service is reachable by address without the directory service, allowing a client to access this service if there is connectivity to it.

also expected to detect when the selected node fails or becomes disconnected, causing traffic to be rerouted through the other CRR. Because the name is the primary identifier and can be rebound without losing the connection state, the connection can survive this redirection to the other router similar to a connection surviving rerouting in the current Internet. With routing updates significantly damped in the Internet to avoid oscillations, especially at the BGP level, we expect TRIAD name rebinding to provide recovery latency that is comparable, if not superior, to that of the current Internet.

4.2 Name-based Routing

The TRIAD *Name-Based Routing Protocol (NBRP)* [3] performs routing by name with a structure similar to BGP. Just as BGP distributes address prefix reachability information among autonomous systems, NBRP distributes *name suffix* reachability among address realms. Routing information is distributed among CRs and maintained locally with next hops and destinations specified in terms of names and name suffixes. With this step, the name directory and the routing table logically become a single entity, reducing the overall complexity of the CR software[5].

This *name-based routing* distributes name mapping information to ensure its availability, to distribute the name lookup load, and provide faster name lookup response. Identifying endpoints by name is also necessary because addresses are not unique across the multiple realms in TRIAD. (Intra-realm routing can use existing routing protocols. Intra-realm reliability of name service can be ensured by duplicate servers as now.)

The key challenge with name-based routing is maintaining the routing database efficiency in the presence of names that do not match the routing hierarchy. To reduce routing table size to a feasible level, name-level redirection mechanism is used to handle hosts whose names do not match network topology. For example, all hosts with Harvard names not in the same address realm as the authoritative server for `Harvard.EDU` would have redirect records at that server. Consequently, the routing database only needs to deal with `Harvard.EDU`, not hosts subordinate to this domain. Nevertheless, TRIAD can deal with third-level names as necessary, such as `europe.ibm.com`, `japan.ibm.com`, etc.

NBRP also supports combining collections of name suffixes that map to the same routing information into *routing aggregates*. For instance, we expect an ISP relay node to group all of the names from its customers into a small number of aggregates. With aggregates, routing updates typically update a small number of aggregates rather than the large number of individual name entries contained in each aggregate. Moreover, all names in a routing aggregate may be treated identically in routing calculations, thus reducing load at CRs. Aggregate membership should be relatively long-lived, so that CRs can amortize the cost of learning the aggregate membership over many routing updates. Section 7 describes measurements of a preliminary evaluation of the benefit of route aggregates.

Although TRIAD name lookups may contain a full URL, cookies and other content level identification, the key information maintained by network nodes is essentially the same as current DNS. Most more detailed content information is maintained at end nodes, and cached in transparent proxy caches.

To keep the number of NBRP neighbors small so that the routing overhead is acceptable, nodes on the interior of a realm can be added that perform only route updates and name lookups but do not otherwise participate in the routing. Current BGP speakers could be upgraded to perform as NBRP speakers as well participating in the routing. The CRR of a typical ISP customer is a degenerate case of this approach.

4.3 Host Advertising, Aliases and Content Routing

A host can advertise an alias associated with its canonical name(s) to the NBRP system together with the hop count cost of accessing the associated content through this host. This cost is in terms of "hop-equivalent" response time units. That is, if the response time cost of going an extra hop is estimated as 2 millisecond, this advertised cost is k if the server is averaging $2 * k$ milliseconds to respond to a request. The routing system adds this cost to the routing cost of accessing this host as it disseminates the naming and routing information. Explicit advertising of host cost is limited to content servers, a very small fraction of the hosts on the Internet.

A CR receiving advertisements of two or more canonical names with the same alias name groups these canonical names for lookup under this alias. On receipt of a name lookup for the alias, the CR orders

[5]Note that a conventional routing table is a simple directory: It is queried with an IP address to determine the forwarding information. With TRIAD, the equivalent directory in a relay node is queried using the DNS name.

the associated canonical names by proximity (determined from the routing database) to the requestor and forwards the request to the closest proxy.

This approach has several benefits for content routing. Because this information is pushed out to each CR, a client request is immediately routed to a close-by content server that than having to first go to a distant central server. This distribution also avoids excessive load and failure-dependence on a central resource. Moreover, the proximity information is based on server load and availability, not just network access.

4.4 Security

The directory service supports message authentication using public-key and shared-key cryptographic signatures. This allows clients to determine that the answer they get from the directory service is authentic, and allows relay nodes to identify a particular principal associated with a client.

NBRP updates are authenticated by cryptographically signing "delegations" of part of the namespace to a CR's peers, in a manner similar to Secure BGP [18].

Unlike DNS security[7], a single name-to-address mapping cannot be signed by the authoritative server for a name because the address also depends on the intervening CRs. Instead, CRs must establish trust relationships.

5 WRAP and Path-based Addressing

In TRIAD, WRAP, the *Wide-area Relay Addressing Protocol (WRAP)*, is a "shim" protocol that specifies, together with the IP packet source and destination addresses, a *path* to desired content. It carries the transport header and data as its payload, similar to other IP encapsulation protocols.

The WRAP header contains a pair of *Internet Relay Tokens (IRTs)*, the *reverse token* and *forward token*. The forward token represents the path the packet is to take and the reverse token indicates the path the packet has taken to this point. An IRT is a potentially opaque variable-length field that specifies a path from the source to the destination. It may also be simply a sequence of IPv4 addresses.

A WRAP packet is formatted as in Fig. 1 as the payload of an IPv4 packet.

0-7	8-15	15-23	24-31
protocol	**length**	**foffset**	**reserved**
reverseToken			
forwardToken			
data			

Figure 1: WRAP Packet Format

The `protocol` field specifies the higher-layer protocol in the "data" field using the same types as for IP, e.g 6 for TCP. The `length` field is the number of 32-bit *components* in the header. Thus, the WRAP header length in octets is 4 + length * 4. The components specify the reverse and forward tokens. The `foffset` field is an offset into the list of components where the `forwardToken` starts, with 0 referring to a null `reverseToken`, so the forward token starts in the first 32-bit field. The `forwardToken` is used by the node addressed in the IP packet destination address to determine how to relay the packet to its destination. The `reverseToken` is a value used by the node addressed by the IP packet source address to refer to where the packet came from.

The data field contains a TCP, UDP, or other transport protocol packet.

A WRAP source sends packets to a destination by forming an IPv4 packet with the IP destination address set to the address of the next relay, the WRAP header containing the IRT in the forwardToken of the destination relative to this relay node, and a null reverseToken. Thus, the length field is the length of the forwardToken and the foffset field is 0. (The foffset value is also the length of the reverseToken.)

A node, on receiving an WRAP packet:

1. maps the (SA,DA) of the packet to a *virtual interface* (VI) that represents the local endpoint of the realm "channel" on which the packet arrived. It maps the next k 32-bit components of forwardToken

field (assuming foffset is less than length) to a corresponding relay entry in a relay table associated with this VI.

2. determines from this entry the next IP source address (the "egress" interface), the next relay's IP address, the new forward token, and the rewrite of the reverse token to perform.

3. forwards the modified packet to the next relay node, with the IPv4 destination address as that of this next relay and the IP source address determined above, and increments the foffset field.

Each relay node thus "consumes" one or more 32-bit components from the forwardToken and adds an equal number of components to the reverseToken. (However, both these fields may be translated according to the information in the relay node's lookup table.) The reverse token, when component-reversed, must be recognized by this node when used as a forward token, to send packets back toward the source.

Normally, an node just rewrites the first forward token component and increments the foffset by 1. In the simplest case, the rewritten component is just encodes the IP source address of the incoming packet (that is, the last relay point.) This restricted form of relaying, though less general than WRAP allows, is more amenable to hardware implementation, because less of the packet needs to be rewritten.

If the node does not recognize the forward token, it drops the packet and may send a WRAMP message back to the previous node. The relaying state may include filters on sources from which to accept packets and destinations allowed for given sources.

The receiver of a WRAP packet is a node that receives the packet with a null forwardToken, or receives a packet with a multicast destination and subscribes to that multicast source.

The actual source of the packet is identified by the reverseToken and the IP source address. The receiver can contact this previous relay identified by the IP source address to do a reverse name lookup on this IRT to determine the name of the actual source.

With WRAP, a packet is reassembled from fragments at each intermediate relay node, because each is a destination from the IP standpoint. This feature reduces the risk of carrying packet fragments all the way to the destination only to discover some fragment is missing. Each WRAP node sets the TTL of a WRAP packet according to its estimate of hops to the next relay. Packets cannot loop at the WRAP level because some non-zero portion of the WRAP IRT is consumed at each relay node.

Backbone or wide-area ISPs can connect at peering points, the same as today, but with high-speed relay routers at these points. At this level, the firewall or border router is extended to act as a TRIAD relay between realms, translating packet addresses as it relays packets between the realms that it interconnects.

Between the IP addressed nodes, a packet is routed by the normal IPv4 routing protocols used within the realm. Thus, WRAP is similar to loose source routing with the relay nodes as the designated nodes on the path it is to follow.

Within a realm, the operation of naming, addressing and routing operates the same as currently with IPv4. Thus, there are no host or router changes required. A packet that does not travel outside of a single address realm can omit the WRAP header entirely.

5.1 WRAP Example with Multiple NAT Realms

Fig. 2 illustrates the operation of TRIAD between realms with two hosts, `src.Harvard.EDU` and `dst.Ietf.ORG`, assuming `Harvard.EDU` and `Ietf.ORG` are two separate realms connected via a single intermediate realm, the "external" Internet. (For simplicity, we illustrate just with DNS names, not full URLs.) For `src` to send to `dst`, the name lookup of `dst.Ietf.ORG` is handled by the relay node `relay.Harvard.EDU` for this realm, with internal IPv4 address $RA1$ and external IPv4 address $RA1'$. This relay determines the appropriate next relay from its directory mapping of `Ietf.ORG` and then communicates the name lookup across the Internet to the `relay.Ietf.ORG`, the relay for the `Ietf.ORG` realm. (This relay has internal IPv4 address $RA2$ and external IPv4 address $RA2'$.) In response to this query, `relay.Ietf.ORG` communicates with `dst` to set up connection state and then returns to `relay.Harvard.EDU` an IRT `f'` that designates `dst` relative to $RA2'$ and the associated transport connection information. Then, `relay.Harvard.EDU` returns an IRT `f` to `src` which designates `dst.Ietf.Org` relative to $RA1$, creating any state it needs to map `f` to `f'`, passing the transport connection information through.

Then, `src` sends the first data packet over this connection as an IPv4 packet addressed to $RA1$ with `f` stored in the WRAP header. On reception, $RA1$ translates `f` into `f'` and transmits the packet with

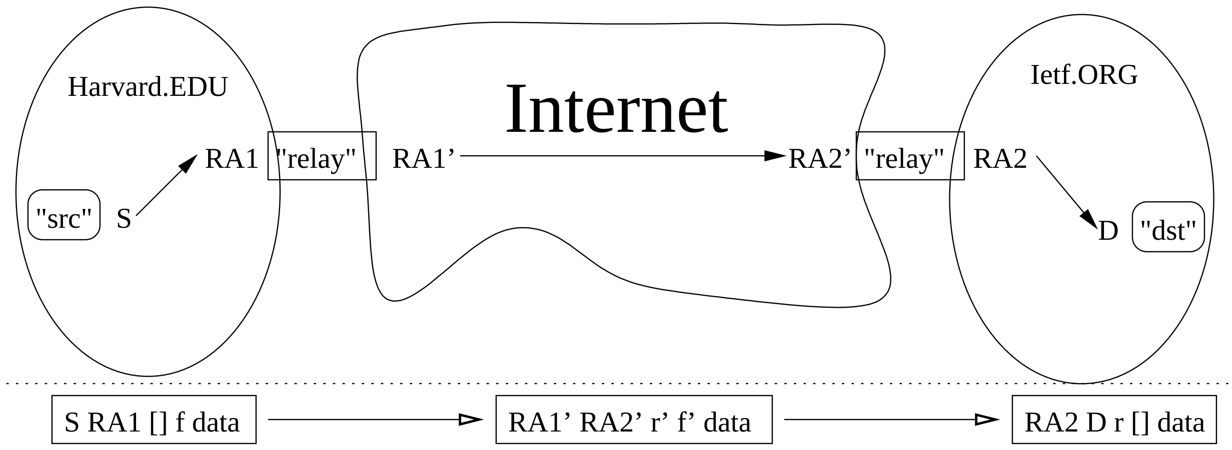

Figure 2: Inter-realm packet transmission in TRIAD: The host named "src" with IPv4 address S in realm Harvard.EDU sends to the host named "dst" and IPv4 address D in realm Ietf.ORG. The packets below the dotted line indicate how the IP and WRAP headers changes as it crosses the three realms, with the header listed as source address, destination address, reverse IRT, and forward IRT.

destination address $RA2'$ and source address $RA1'$, as shown in the middle packet in the figure. The WRAP header also contains the reverse IRT `r'`, which indicates the source of the packet relative to $RA1'$.

On reception at $RA2'$, the reverse IRT in the packet is translated to a new value `r` which represents `src.Harvard.EDU` relative to $RA2$. This relay then transmits the packet with an empty forward IRT, IPv4 destination address D and and source address $RA2$, as shown in the rightmost packet in the figure.

Thus, `dst` receives the packet as a normal IPv4 packet sent by its relay $RA2$ but also containing an IRT that identifies the actual source of the packet relative to its relay. The packet is then passed up to the next higher layer processing, such as TCP or UDP.

The destination can respond to a packet directly by "reversing" the IRT and sending the packet to the local relay with this reversed IRT. This causes the packet to return along the reverse of the relay path on which the original packet was received. Alternatively, the destination can perform a lookup on the source name to get a separate IRT and RA to reach this host. This alternative is more flexible, allowing for asymmetric routing at the cost of an extra name lookup.

An address realm at the leaf level may correspond to an enterprise or university network, a military installation, or much smaller units like a collection of autonomous sensors or a home network or even a set of virtual hosts on a single physical machine. Higher-level address realms correspond to local and global Internet service providers (ISPs).

The IRT and the relay address are local in scope and transient. That is, the IRT is only meaningful relative to the relay and realm and is only guaranteed to be *T-stable*: it does not go from one valid association with a relay to another in less than time T, where T is typically hours. In particular, it can become invalid at any time but can only be reassigned to another use after time T. Thus, passing an IP address or an IRT in the data portion of a packet to the other endpoint is meaningless in general.

A WRAP proxy, referred to as a WRAPID gateway (see Section 8), allows existing IPv4 hosts to interact with WRAP-enabled hosts and servers without any modification. A WRAPID gateway is just an extended NAT-capable router or firewall which is able to WRAP and unWRAP packets going through it, as appropriate.

5.2 Transparent and Opaque Relaying

WRAP allows the relaying to be *transparent* in the sense that each IRT is simply a sequence of IPv4 addresses designating relay nodes and endpoints, an *Internet Relay Path (IRP)*. The IRT can also be *opaque* so that a holder of the IRT cannot determine the relay path nor can it forge a valid IRT.

Using a transparent IRT, the relay is stateless in the sense that the relaying only relies on routing/directory state and configuration state; it does not require state to be created on name lookups. In this mode of operation, the relay node is statically configured with an IPv4 address for each of the other realms it connects to, so that an address uniquely identifies which direction to relay a packet (and a particular "egress" address in that realm.) Upon receipt of a WRAP packet, the relay replaces the IP destination with the first forward token component, uses the egress address as the new IP source, and places the old IP source as the last component in the reverse token, as described in the earlier example. This is the simplest possible relaying action, requiring only 4 words in the packet headers to be modified.

To make the WRAP addressing opaque to an observer, the relay node can choose to put a random value in the IRT and translate it to/from IPv4 addresses using the relay table described earlier. This opaque form prevents a upstream source from fabricating IRTs, forcing it to rely instead on the directory service to supply IRTs. In particular, an ISP can retain control of routing, preventing customers from using unauthorized routes. It also prevents a third-party observer from determining protected information from addresses in the packets.

An IRT must have the *reversibility property*, namely that the component-wise reversal of the received IRT provides an IRT that can be used to send a packet back to the source of the packet using the relay from which the original packet was received.

An IRT normally also has the *concatenation property*, i.e. if the IRT to a relay is X from a host H and Y is the IRT to a destination D relative to this relay, then XY is an IRT to destination D from host H. The directory indicates whether the returned IRT supports concatenation or not.

5.3 Multicast

WRAP supports the EXPRESS [4] single-source model of multicast. Multi-source multicast applications can be supported by relaying the multicast through a node that is a source of an existing relay multicast channel, similar to the rendezvous point in PIM-SM, but performed at the WRAP or the application layer.

A subscriber joins a multicast channel by specifying its name in a DRP lookup, which returns the required multicast channel addressing information.

A multicast WRAP header contains the same multicast address G repeated r times, where r represents the maximum number of relay hops in the multicast tree; this allows multicast WRAP relaying to performed identically to unicast WRAP relaying. As multicast packets are relayed, group addresses may be translated so that the (S,G) pair upon which IPv4 routers do multicast delivery is unique. However, a single intra-realm channel can be reused within a realm to deliver multiple inter-realm channels.

5.4 Content Routing and WRAP

WRAP allows the directory to dictate a path for packets to take from the client to receive the delivery performance it has determined, rather than leaving the client packets to be routed by a separate mechanism, as occurs now.

WRAP also allows the directory to return a path specific to a requesting client, rather than an address that is generally common for all clients.

Finally, WRAP allows a server behind a NAT box to be addressed without creating translation state in intermediate nodes.

5.5 Secure Communication

TRIAD includes a secure communication facility similar to IPsec, i.e. end-to-end at the (inter)network layer. It differs primarily in working in the presence of NAT or WRAP translation, because the Integrity Check Value (ICV) does not include the packet addressing information, similar to the TRIAD-TCP pseudo header. Here, the principal associated with the connection can be identified by name.

6 Additional TRIAD Benefits

TRIAD provides benefits in mobility, VPNs, policy-based routing and extended reverse path forwarding checking, in addition to its support for content routing, NAT and scalable addressing, as outlined below.

6.1 Mobility

For mobile operation in TRIAD, a host visiting a guest network receives a temporary visitor name in that network (in a DNS domain of the visited network) which allows it to then communicate with the rest of the Internet. If the host needs to be reachable or authenticated as its normal DNS name, it gets its home directory service to insert a redirect this name to its current temporary visitor name in the guest network. It also notifies the guest network of its home identity, based on name. When another host attempts to contact the visitor by its normal name, the home directory provides a redirect to the temporary visitor name, causing the other host to then contact the mobile host directly in the guest network.

When the mobile host moves, transport connections can continue to function even though its address may change. The mobile host simply acquires a guest name in the new network and registers its real name with the guest network and its guest name with its home network. The transport connections simply rebind based on the name identification, the same as required when NAT translation state was lost or changed in the network. For real-time hand-off between guest networks, the mobile host can request that a relay node in the old network forward its packets (using encapsulation) to the new guest network for some limited period of time. This forwarding is canceled before the relay node reuses the address and name that was used by this guest host (allowing the relay node to use common state and time-out mechanisms to control the forwarding and the reuse). A reverse name lookup can also return the "real" name that redirects to this (temporary) name, providing what might be called *reverse aliasing*. A lookup on this real name is used to validate this reverse alias.

With this approach, the key mechanism to support mobility is the adding and removing of redirects in the home directory of the mobile host and the registering of the mobile host in the guest realm. The guest network simply needs to allocate and reclaim temporary addresses and names the same as supported by current DHCP services. It does not require routing all packets to a mobile host through the home gateway for the mobile host or encapsulating traffic to and from the mobile host, as mobile IP proposals imply.

6.2 Virtual Private Networks (VPN)

Using WRAP, an ISP can provide a *Virtual Private Network (VPN)* service by creating for each enterprise a secure (virtual) transit realm that connects to each of its enterprise sites. The enterprise directory and routing system only needs to deal with the topology of the enterprise network with this "virtual" realm directly interconnecting all the sites. The different sites of the VPN can even have overlapping IPv4 address assignments (typically 10.X.X.X) yet still communicate directly without renumbering.

The ISP implements this virtual realm simply by providing routing and secure communication between each site. In this case, the overhead from WRAP is 12 bytes. As an optimization, the ISP can provide at each site a relay node address for each other remote site in the VPN so packets can be addressed as though each site was directly connected through a relay to each other site, reducing the header overhead to 8 bytes. Thus, WRAP can also obviate the need to deploy MPLS [14].

6.3 Policy-based Routing

WRAP supports policy-based routing across multiple realms[6] by using the WRAP path-based addressing to direct packets through certain relay nodes and to avoid others, with the directory mapping particular names to these policies. For instance, a special name in the name lookup can provide a special IRT to a destination that directs packets over a more secure ISP network to a particular destination rather than using a cheaper but less secure route. The ISP directory service can also provide different IRTs based on the class of service that the requesting customer is paying for. It is similar in this sense to source routing and tunneling, but with the key differences discussed in Section 10.

This routing control can also be used for traffic engineering.

6.4 Extended Forwarding Path Check

WRAP supports an *extended forwarding path (EFP)* check based on the WRAP header indicating the (relay) path it took to the receiver, not just the port that the packet arrives on. The receiver can verify that the packet was received from trusted relay node based on the IP source address, only trusting the local network realm to prohibit source spoofing. It can further rely on the relay node to only accept packets from trusted relay nodes in other realms. With this constraints, a reverse path name lookup reliably yields the name of the source, at least to within some originating domain.

With this approach, the true source of the packet is explicitly specified in the packet, up to the trustworthiness of the relay nodes. The conventional reverse path forwarding check is only used within a local realm to prevent local source spoofing. Thus, a receiver or relay can check whether the relays that the packet took

[6]Each realm can support its own local policy-based routing.

are trusted and accepted, independent of whether it would forward a packet to the source of this packet back along the same path.

Unlike conventional source routing, WRAP operates with strict reverse path forwarding (RPF) checking in place and does not allow source spoofing attacks.

RPF checks at the IP and WRAP levels are important because so-called source spoofing is the basis for many denial of service and security attacks. These attacks and various forms of network device failures and misconfiguration are a growing concern with the scaling of the Internet. A key part of handling these problems is having a reliable means of verifying the true packet source. Encryption techniques providing authentication and confidentiality can, by their cost in processing, actually make denial-of-service a bigger problem. That is, the increased time to decrypt a packet with secure communication, only to discover it is a bogus packet, means a node loses more resources in an encrypted denial-of-service attack than with plaintext messages. (Providing wire-speed hardware-supported encryption addresses this problem in part, but is an expensive solution for low-end systems and generally does not deal with setup processing, such PKE-based authentication on connection setup.) For mission-critical applications, denial-of-service may be as damaging an attack as any of the other possible security attacks. We view ERPF allowed by WRAP as an important feature for scaling anti-source spoofing and dealing with these DoS concerns.

7 Implementation and Evaluation

We have developed a prototype implementation of the extended directory and routing service required in TRIAD. The key issues are the client name lookup performance and the directory/routing storage and maintenance overhead.

Regarding name lookup, we expect most environments to use transparent IRTs, which have the *concatenation property* mentioned in Section 5. Thus, the caching of names behaves the same as with current DNS because a name server can lookup the address to a server once, then send name requests using the relay fast path to this server rather than through the name service on each intervening relay node. The concatenation property also allows addresses looked up for one client to be used for another client on the same "side" of a relay node. Caching of names thus behaves the same as with current DNS.

Furthermore, content lookups would be typically handled by a content cache on the path to a primary content server, providing faster client service than the current Internet and keeping the name lookup (or content routing) local to this portion of the Internet close to the client.

Opaquing the IRTs can defeat caching, particularly if the returned IRT encodes source dependencies, but the cost is low compared to the other overheads with secure connection setup.

On a name cache miss, in TRIAD, the name lookup may proceed through several relay nodes, causing a full name lookup at each relay node. In contrast, a conventional DNS name cache miss (within an enterprise) causes a DNS request to be sent to a root name server. Thus, TRIAD may use more cycles in total, summed across several relay nodes, but it distributes this load over the relay nodes on the path of communication. In contrast, DNS incurs fewer total lookup cycles but concentrates the demand on the smaller number of root servers.

The number of name suffixes which must be searched is large, but not unacceptably so. There are currently 1.7 million second-level names in use world-wide, e.g. Harvard.EDU, Ietf.ORG, etc. (This number closely matches the number of suffixes obtained from the experiment explained below.) Assuming 64 bytes of space per entry (including hash indexes, etc.), storing the whole name database would cost 128 megabytes, an insignificant amount of disk space, even if the number was to be 10 times as much by the time TRIAD was deployed. NBRP table lookup is not on the packet forwarding fast path, unlike IP routing, so time spent searching the table is typically only paid during connection setup rather than per-packet. Note also that a name lookup already encounters the cost of searching through a database of this size in conventional DNS.

In sum, we expect TRIAD name lookup to have comparable performance and scaling as current DNS, differing primarily for portions of the Internet configured for greater security requirements than supported by current DNS.

Considering the directory and routing overhead, at the ISP level, the name aggregation generally closely matches the address and routing aggregation. For example, Harvard.EDU corresponds to a small number of IP address ranges that further correspond to a small number of routes. This strong correspondence means the aggregation feasible with routing table entries is largely intact in going to name-based routing and directory services. Conversely, organizations with large numbers of hosts scattered throughout the Internet are uncommon.

7.1 Expected NBRP Performance

To evaluate the expected performance of name-based routing in the current Internet, we processed a comprehensive list of address-to-name mappings in the Domain Name System[19] and BGP table dumps from the MAE-East exchange point[20] by the following algorithm, making the assumption that address realm boundaries roughly correspond to current BGP autonomous system boundaries:

1. Each address range from the BGP table is matched with the DNS zones represented. (If fewer than *site_threshold* hosts in a range belong to an existing zone, they are removed from the table completely and assumed to be handled with the redirection mechanism.)

2. Names whose associated routing information is made redundant by a superzone are also removed.

3. Aggregates are created for any set of names larger than *aggregate_threshold* that have identical routing information (i.e., all known routes were identical, not just the preferred route.)

The resulting aggregates match those expected to be generated in a TRIAD relay node.

One representative set of results is shown in Table 1. These numbers indicate that NBRP results in a

site_threshold	Affixes (1000s)	*aggregate_threshold* 3	5	10	20
2	1727	19.5 (6.7)	20.1 (5.6)	25.7 (4.4)	37.0 (3.4)
3	1692	14.9 (5.9)	16.1 (5.0)	20.6 (4.0)	30.1 (3.2)
10	1679	14.8 (5.9)	16.0 (5.0)	20.6 (4.0)	30.3 (3.2)
original BGP	68.2	11.8			

Table 1: Number of routes (and aggregates) in thousands for different site and aggregate threshold values. With a site threshold of 10 and an aggregate threshold of 3, NBRP produces approximately 14,800 routing table entries (and 5,900 aggregates) which improves significantly on the original BGP number of 68,200 routing table entries.

set of destinations (and thus update frequency) comparable to BGP; higher-level aggregation may be able to reduce this yet further without resorting to renumbering or renaming.

BGP does have a limited mechanism for aggregation: a single route update may include several address prefixes. It is not clear the extent to which BGP software makes use of this to optimize update calculations: there is no requirement that advertisements keep these address prefixes together, and the address ranges must appear separately in the IP routing table. The entry in Table 1 corresponding to "original BGP" with an aggregation threshold of 3 indicating 11,800 entries indicates the best possible number of routes with BGP aggregation.

Addition of a new name is common, unlike addition of new BGP prefixes, and this name information must propagate to all relay nodes. However, addition of new names is done on human time scales; during the recent past, third-level domain names have been added at about 12 per minute. To put this in perspective, a backbone router may receive more than 2,000 routing updates per minute. Also, the actual level of routing updates necessary for new names is lower because changes to aggregates can be "batched" to reflect many new names with one update.

7.2 WRAP Implementation and Performance

WRAP incurs a low space and time overhead for communication on average because communication within a realm just uses the conventional IPv4 header. Given that most communication is local and the current Internet with NAT boxes is effectively at most 3 relays to anywhere, the packet header overhead on average is expected to be significantly less with WRAP than with IPv6[7].

[7]One could argue that the Internet does not actually need more global addresses, by relying on efficient allocation and NAPT, given only about 1 percent of the IPv4 addresses are actually in use. However, WRAP is still beneficial for other reasons, such as connecting private address domains, VPNs and content routing.

This header overhead is significant because most packets are small and per-packet processing is a significant cost with small packets. This optimized local communication also suits small embedded systems, many of which use or will use limited bandwidth wireless communication. Moreover, it is readily hardware implementable because of the size of relay address to lookup can be fixed size.

In comparison to conventional forwarding, relaying requires an additional lookup of the next forwarding component in the context of the virtual interface to which the packet is mapped with the (SA,DA) lookup. A hardware implementation may add an additional lookup resource to handle this or simply perform two lookups on the same memory, depending on the speed of this memory and the forwarding performance requirements. We expect initial hardware implementations may restrict the *foffset* they support, throwing packets with larger values to software.

Multicast relaying uses additional state in the forwarding path but is the same implementation and performance.

Our Linux implementation of WRAP added about 1,500 lines of code as a kernel module and incurred an extra 2.2 microsecond overhead (or 2.6 percent) for relaying compared to conventional IP forwarding.[8] Thus, the complexity is minimal for either software or hardware, and the software performance overhead is minimal, and comparable to that required for NAT forwarding.

8 WRAPID Gateways

A WRAPID (WRAP-to-IP-Domain) gateway allows existing IPv4 end hosts to operate with TRIAD without modifications to their software. WRAPID provides translation between IPv4 addresses and WRAP addressing similar to the IPv4 to IPv4 translation provided by NAT boxes.

A WRAPID gateway handles outgoing conventional DNS lookups, performing a DRP lookup using just the DNS name. The WRAPID gateway allocates an IPv4 address for this remote server and sets up a mapping to the appropriate WRAP header. This header may map directly to the remote host or to a WRAPID gateway that serves that host. When an (IPv4) packet is sent to this allocated address, the WRAPID gateway translates the packet to a WRAP packet with the appropriate header and forwards it onwards. On receiving a WRAP packet from an external host, the WRAPID gateway translates the packet to a simple IPv4 packet with the IP source appearing as this locally allocated address.

The WRAPID gateway also handles incoming DRP requests, performing the name lookup internally, and then requesting a connection setup at the host, using the URL information in the DRP request and sends an HTTP request to the client, if that information is present in the DRP request. It then returns this connection information and splices the client connection into the connection it has established to the server.

The WRAPID gateway can also implement WRAPsec, providing secure communication to the other WRAP endpoint, either a WRAP-enabled host or another WRAPID gateway.

The WRAPID gateway allows WRAP to be deployed incrementally. In particular, one can have hosts on the same subnet being WRAP-enabled while others are not, yet still able to communicate with each other as well as hosts in other address realms. The optimization of eliminating the WRAP header when communicating within the same address realm means that a WRAP-enabled host never sends WRAP packets to other hosts in the same realm, so there is no need to discriminate between these hosts as part of local communication. Only the directory service interfacing to the rest of the Internet needs to distinguish. However, a full TRIAD implementation (with the attendant host changes) is required to provide end-to-end security and reliability.

9 TRIAD Deployment

TRIAD has a simple deployment path, based on user need, allowing TRIAD to be realized as an incremental evolution of the current Internet.

At the content layer, DRP and NBRP can be implemented in firewalls and inter-realm routers with the additional capability to fail over to using the current Domain Name System, etc. to implement the functionality in regions of the network that do not support TRIAD. Similarly, content resolvers can make use of the existing naming infrastructure to locate other TRIAD gateways rather than participating in a dynamic routing protocol. We expect deployment to occur mainly at the edges of the network, and thus cannot depend on ISPs providing new infrastructure. Such a scenario could lead to an topology that would

[8]The test machine was a 333 MHz Celeron with 128 MB of RAM, running Linux 2.2.13.

have many thousands of realms peering in the "global" Internet, but there is little need for NBRP in such an environment. The amount of topological change in such a situation is small, and multihomed sites can easily list all their gateways as NS records rather than constantly updating the DNS. Later, these CRRs can be upgraded to content routers as ISPs begin offering NBRP.

Deploying TRIAD for NAT is also compelling. Consider as an extreme example a foreign country with limited IPv4 addresses such as Thailand. The limitations of conventional NAT make it questionable as a solution to providing more addresses yet moving to IPv6 does not make sense either, given the limited deployment of IPv6, the limited product support, and the need to communicate with the IPv4 portion of the Internet. However, with TRIAD, each such country can install a WRAP relay router that interfaces to the Internet. Attached to this top-level relay are one or more WRAPID gateways that include conventional NAT capability. The conventional NAT capability allows these hosts to communicate with the existing conventional IPv4 Internet. Each ISP, country or even organization that adopts TRIAD is able to communicate with other organizations using TRIAD *without* consuming any of its global IPv4 addresses[9]. For instance, if Thailand and Indonesia both adopt TRIAD, they then have virtually unlimited addresses internally and between themselves, and are only constrained on the number of addresses they have available to communicate with the current Internet. (This is actually the same situation as if they had internally converted to IPv6, given they would still have to communicate with the rest of the planet using IPv4. But, with IPv6, they would also have to upgrade all their existing hosts and networking infrastructure.)

Thus, each organization is motivated to adopt TRIAD because it allows them to communicate with other TRIAD organizations without using their limited global IPv4 addresses, and because it makes it easier for other TRIAD users to communicate with them. So, those organizations that are currently short of addresses are motivated to move to TRIAD and those that are not are still motivated if they are interested in having the former communicate with them. Given that most of the major web sites are in the United States, and the U.S. companies have been in the lead to build Web-based operations, there would be considerable commercial motivation to support TRIAD in the American web sites once foreign companies were using TRIAD among themselves.

This initial deployment requires no real changes to end hosts and no change to the basic IPv4 routers and switches constituting the infrastructure of the leaf and backbone networks. It only requires the deployment of content routers and WRAPID gateways, but these are modest extensions of the current NAT-enabled routers. Here, we assume that end-user applications have been or will be modified in any case to deal with the lack of meaning of addresses across NAT boundaries.

Once WRAP is deployed to some degree in the Internet, first host implementations are expected to arise with large-scale servers where eliminating the extra overhead, delay and point of failure of a WRAPID gateway may be warranted. Making an externally accessed server WRAP-enabled also eliminates the server use of an externally visible (IPv4) address which, with an active server, would be essentially allocated indefinitely to this server. During this transition, conventional IPv4 hosts and TRIAD-aware hosts can easily and efficiently co-exist in the same address realm. Given that WRAP appears relatively straight forward to implement, the main delay in getting all hosts upgraded to WRAP is expected to be the basic inertia in getting changes into commercial software and getting administrators of systems to upgrade their software. Hosts that need end-to-end security and reliability are also motivated to upgrade to native WRAP.

Consequently, TRIAD is readily deployable incrementally. There is no need to change the network infrastructure within an address realm or to change backbone routers and management. The boundary (NAT) routers are upgraded to support TRIAD and then the hosts can then be individually upgraded to use WRAP natively.

10 Related Work

The original Internet directory service was supplied by a "hosts.txt" file that listed all hosts in the Internet. As the Internet grew, this approach was replaced by DNS [6] in 1985. Subsequent work on so-called network directories such as X.500 have suffered from misguided objectives of supporting naming of other types of objects such as mailboxes and providing more flexible ways of specifying identification, such as lists of attributes.

[9]This assumes the gateway already has one such address if the WRAP relays communicate over the existing wide-area IPv4 infrastructure.

TRIAD draws on the direction established in the web of treating both host name and file name as part of the path name to content, and unifies the handling of these two portions of the URL. It further builds on the *decentralized naming* approach advocated from experience in the V distributed system [21], and effectively implemented in file systems, which can be summarized as: Names are external identification of objects and a server names what the objects it implements — nothing more and nothing less".

Current wide-area content routing depends on HTTP or DNS-level redirect. For instance, Cisco's Distributed Director (DD) redirects a name lookup from the main site to a replica site closer to requesting client address, based on responses from a set of participating routers running an agent protocol, supporting DD. Unfortunately, the client incurs the response time penalty of accessing this main site DD before being directed to the closer site. Proprietary schemes by Akamai, Sightpath, Arrowpoint and others appear to work similarly.

Web caching was introduced by the Harvest project, spawning a whole industry of vendors. Transparent caching evolved to eliminate the need for explicit configuration and the difficulty of configuring hierarchical caches.

Protocols such as ICAP, Cisco's WCCP and Arrowpoint/Cisco's CAPP define communication between web caches, web caches and routers, and other content distribution devices. To date, these and other proprietary protocols have not be architected into a coherent Internet architecture.

The XNS [23] Sequenced Packet Protocol (SPP) used a separate connection setup protocol, similar to the separation between DRP and TRIAD TCP we are proposing.

NAT was introduced into the Internet in Jacobson's insightful early proposal [11] although the same techniques appear in some earlier distributed systems work [10]. Since 1992, various RFC's [12] have clarified the use of NAT, provided for private addresses [13] and clarified the terminology, use and problems [17]. Industry has deployed a variety of products supporting network address translation, including firewalls, routers and server load balancing switches. More recently, work on IPsec and others have recognized the problems with basing identity on IP addresses and the conflict of end-to-end security with the increasing deployment of NAT.

RSIP [15] is an approach to dealing with NAT, where a host in a NAT realm explicitly obtains an external IP address, tunnels packets through the NAT gateway using this external IP address and thus can use IPsec and other protocols without requiring NAT translation. However, RSIP requires host modification to operate in this mode and it does not increase the number of external IP addresses. With all the extra benefits that TRIAD provides, it seems more effective and lower risk to modify the hosts to support native TRIAD.

The path-based WRAP relay model of addressing as an extension of the basic forwarding level of conventional routers is similar in some respects to the Sirpent [1] form of loose source routing except WRAP is designed to work with IPv4. WRAP relaying is similar to loose source routing except the packet is forwarded at each relay with the source IP address that of the relay, not the original source address. Moreover, each specified address may be in a separate address realm, with translation between address realms occurring at each realm boundary. Source routing provides a source-controlled path but does not cross realms and does not change the source address on each hop, as is required for inter-realm communication. TRIAD path-based addressing could be provided as a new IP-level option. However, using a separate shim header seems preferable because of the inefficiency of routers handling packets with IP options, given that some options need to be handled by each router and others only need to be handled by the IP-addressed endpoint. Moreover, WRAP makes it easier for a hardware implementation to determine the offset of the transport header, which is important for layer 4 access control lists. WRAP is similar in this respect to IPv6 header extensions.

IP tunneling has been used to effectively extend addressing by tunneling from one realm to another. However, tunneling makes layer 4 filtering harder because, with multi-hop tunneling, the location of the layer 4 header involves parsing each encapsulation. Also, unlike WRAP, the path the packet takes is lost with tunneling. Moreover, tunneling incurs greater overhead than WRAP and requires that the source know the path. Moreover, the packet size does not change with WRAP, unlike encapsulation and de-encapsulation that occurs with tunneling.

MPLS [14] provides tagging of packets similar to WRAP, but below the IP level. MPLS does not provide more addresses beyond that provided by NAT, unlike WRAP. On the other hand, WRAP can be used intra-realm and inter-realm for traffic engineering and VPNs, reducing, if not eliminating, the need for MPLS[10]. MPLS also requires special support in the forwarding path of *all* routers on the path, whereas WRAP and TRIAD only require support at the border or relay nodes. MPLS also requires a new mechanism for distributing tags. MPLS does not save the path a packet followed either. While the WRAP header does impose a higher overhead than an MPLS tag, it is less than IPv6 and less than conventional IPv4 tunneling, especially with multi-path tunnels. Thus, IP4 plus MPLS is not a solution to scaling and IPv6 plus MPLS carries all the disadvantages of both. Both WRAP and MPLS make the offset of the TCP/UDP ports variable within the packet, affecting the design of access control filters on packets. However, with the length field in the WRAP header at a fixed offset, it is straightforward for even a hardware implementation to determine the actual offset of layer 4 ports, as required for access control processing. Moreover, in initial deployment, we expect that firewalls may simply restrict WRAP packets to specific WRAP-enabled hosts, such as WRAPID gateways, which can filter further as needed.

Recent IETF work has promoted "transparency" as an important property to achieve in the Internet, defined as "a single universal logical addressing scheme and the mechanisms by which packets may flow from source to destination essentially unaltered" [16]. We view that TRIAD provides transparency under this definition, viewing the "logical addressing scheme" to be DNS naming and the transmission of data without changing the data or its checksum as "essentially unaltered". The changing of the addressing in the packet is not real alteration because corruption by intermediate points is as detectable as with conventional end-to-end delivery.

The restriction of IP multicast to single-source was proposed in EXPRESS [4]. This single-source multicast approach is now being deployed.

11 Concluding Remarks

TRIAD is a promising candidate for the next generation Internet architecture. It addresses the key problem of scaling content distribution by defining a *content layer* that directly supports efficient content routing, transparent caching and content transformation. It further supports network address translation while providing end-to-end semantics, reliability and extensible addressing. Besides these benefits, TRIAD also provides a significantly improved directory service as well as innovative approaches to mobility, virtual private networks, policy-based routing and source spoofing. Compared to IPv6, TRIAD is more backwards compatible, more deployable, more efficient and more secure while providing the same end-to-end semantics and recovery relative to network failures.

TRIAD, as the name suggests, is based on three key ideas. First, TRIAD places the external character-string name as the means of identification of endpoints, relegating the packet address to the role of a transient routing tag. In doing so, it makes network naming consistent with that used in file systems, where similar hierarchical names map to files, and internal system identifiers or handles are generated and used for efficiently in the file access operations. In fact, one can recognize both network and file-level names designate content or state, and see their combined usage in web URLs.

Second, TRIAD integrates naming, routing and connection setup into a content layer, recognizing name lookup needs routing information to locate the closest replica to the requesting client. It also needs the same reliability, security and performance as routing. The integration of transport-layer connection setup with the name lookup allows the name lookup to be end-to-end while reducing the roundtrip delays for content access. As the bandwidth of the Internet increases to multi-gigabit rates, roundtrip times are becoming the dominant client performance issue. This integration also facilitates better failure handling between the directory and transport layer.

Finally, TRIAD extends packet addressing with the ability to specify a variable-length path to the destination in a shim protocol called WRAP, allowing the directory service to control the path a packet takes. This path addressing also provides extensible addressability between address realms, efficient virtual private networking and scalable anti-source spoofing. The simplicity of WRAP makes it feasible to implement in hardware in the next generation of switch/routers, allowing wire speed relaying, even at the highest performance levels. Moreover, intra-realm communication can optimize out WRAP, incurring the same packet overhead in size and processing as IPv4.

[10] One of the original motivations for MPLS, efficient IP forwarding, has been eliminated by the advent of wire-speed hardware IPv4 forwarding engines.

Our current work is focused on designing and implementing the detailed protocols required by TRIAD and performing further evaluation and study to support this direction. Specifically, we are performing much more comprehensive studies of the large-scale behavior of name-based TCP and routing through simulation. The ability of name rebinding to handle routing topology changes and the effects of route aggregation need to be clearly demonstrated before TRIAD can achieve wide-scale deployment. However, we have confidence in TRIAD's scalability, since the dynamics of naming and routing are similar to what already exists in the IPv4 Internet.

We believe the primary competition to TRIAD at this stage is the continued *ad hoc* deployment short-term fixes and specialized mechanisms, including the proprietary approaches that have arisen for dealing with content distribution. Continued growth of the Internet without a guiding architecture risks increasing entropy as a result of these fixes, detracting overall from its future reliability, availability and security. We see TRIAD as an alternative to this unfortunate direction.

References

[1] David R. Cheriton, Sirpent, Proc. SigComm'89, 1989.

[2] David R. Cheriton and Chetan Rai, Wide-area Relay Addressing Protocol (WRAP), in preparation. 1999.

[3] Mark Gritter and David Cheriton, Name-based Routing Protocol Specification, in progress, 1999

[4] Hugh Holbrook and David R. Cheriton, IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications, Proc. ACM SigComm'99 Cambridge, MA Sept. 1999.

[5] Anonymous (to allow for blind review), Name-enabled Routing and Directory Services in TRIAD, in progress, 2000.

[6] P. Mockapetris, Domain Names - Concepts and Facilities, RFC 882, November, 1983 (obsoleted by RFC 1034 and 1035).

[7] D. Eastlake, Domain Name Security Extensions, RFC 2535, March 1999.

[8] P. Ferguson, H. Berkowitz, Renumbering: What is it and why do I want it anyway, RFC 2071, January 1997.

[9] S. Deering and R. Hinden, IP Version 6 Addressing Architecture, RFC 2372, July 1998.

[10] D.R. Cheriton, Local Networking and Internetworking in the V-System, Proc. 8th Data Communication Symposium, IEEE/ACM, 1983

[11] V. Jacobson, LNAT — Large-scale IP via Network Address Translation, working draft, Lawrence Berkeley Labs, Jan. 1992.

[12] E. Egevang and P. Francis, The IP Network Address Translator (NAT) RFC 1631, May 1994.

[13] Y. Rekhter, B. Moskowitz, D. Karrenberg and G. de Groot, Address Allocation for Private Internets, RFC 1597, March 1994.

[14] E. Rosen, A. Viswanathan and R. Callon, Multi-protocol Label Switching Architecture, work-in-progress, Internet draft, 1999.

[15] M. Borella, D. Grabelsky, J. Lo, Realm Specific IP: Protocol Specification, draft-ieft-nat-rsip-protocol-03.txt, Oct. 1999 (work in progress).

[16] M. Kaat, Overview of IAB 1999 Network Layer Workshop, draft-ietf-iab-ntwlyrws-over-01.txt, work in progress, Oct. 1999

[17] NAT Working Group, P. Srisuresh and Matt Holdrege, IP Network Address Translator (NAT) Terminology and Considerations draft-ietf-nat-terminology-00.txt, work in progress, July 1998

[18] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)", to appear in IEEE Journal on Selected Areas in Communication, 1999.

[19] Internet Domain Survey, July 1999, `http://www.isc.org/ds/`.

[20] Internet Performance Measurement and Analysis project, `http://www.merit.edu/ipma/`.

[21] D. R. Cheriton and T.P. Mann, Decentralizing a Global Naming Service for Improved Performance and Fault Tolerance. ACM TOCS, May 1989

[22] Wireless Application Protocol Forum `http://www.wapforum.org`.

[23] Xerox Network Services Specification, Xerox Corporation, 1980.