PROPOSAL

submitted to the

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

in response to BAA 97-04

by MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Proposal Title: ROBUST, MULTI-SCALABLE NETWORKS

Technical Topic Areas: Primary: 4. Network Protocols, 8. Autonomous Network Management, 10. Managing Systems of Systems. Secondary: 1. Enhanced Network Services

Technical Points of Contact:

David D. Clark, Principal Investigator

Telephone:

(617) 253-6002

Email:

ddc@lcs.mit.edu

Karen R. Sollins, Co-Principal Investigator

Telephone:

(617) 253-6006

Email:

sollins@lcs.mit.edu

John T. Wrocławski, Co-Principal Investigator

Telephone:

(617) 253-7885

Email:

itw@lcs.mit.edu

Massachusetts Institute of Technology Laboratory for Computer Science 545 Technology Square

Cambridge, Massachusetts 02139

Fax: (617) 253-2673

A. Innovative Claims

A robust network is one which provides high confidence and adaptability to changing circumstances as a fundamental, architectural property, rather than by layering additional mechanism above a weak core. A Multi-Scalable network is one which is able to scale in several dimensions, including size, heterogeneity of technology, number of administrative entities, and categories of end-nodes. We propose a new network architecture, the Metanet, that addresses these requirements. We propose to develop, implement, and test three complementary technologies we believe crucial to the construction of the Metanet and other robust, multi-scalable networks. To motivate and validate these technologies we propose a novel self-organizing application framework supporting highly distributed and ephemeral applications. Innovative claims of our work include:

- 1) The Metanet Architecture. The core of the Metanet architecture is the region, a partition of the network which mirrors an externally defined or imposed organization in some dimension. The existence of regions as first-class objects supports a number of capabilities, such as intermediate-point routing and efficient indexing and searching of network elements. Region boundaries are populated by waypoints, locations of translation or exchange from one set of consistency rules or conditions to another. By explicitly managing waypoints and regions, heterogeneity and multiscalability are encompassed and supported in the architecture.
- 2) A new system-level architecture for quality of service control in an inter-network infrastructure, based on the concept of QoS building blocks. We cleanly separate the *mechanism* of QoS control, implemented as a single, extremely simple component at each point within the network, from the selection and enforcement of a particular QoS control *policy*, implemented by modules located (only) at crucial control points such as end-nodes and administrative boundaries. This architecture allows QoS policies to be deployed rapidly and adjusted dynamically as circumstances require. To provide a robust service, each building block must police its use and detect abuse. Traffic sampling is used under normal condition to limit the use of resources, providing high scalability. Under abnormal conditions, the sampling is refined as a series of trial filters to identify and isolate a source of offending traffic. Thus, in contrast with current QoS architectures, ours will provide efficiency and scalability in normal circumstances, without working against the goal of robustness in the presence of failure or threat.
- 3) Formalism and mechanism for addressing location within the network; the "Problem of Where". Reasoning about absolute and relative location of objects within the network will be a fundamental part of next-generation communications architecture, where higher level services are located within the network. The problem is difficult because there is no single definition of "near"; an object near in geographic terms may be simultaneously be far in latency terms. We propose a new structure, the abstract ion map, to provide an efficient basis for using the notion of location in higher-level protocols. We use this structure to support location-aware routing and addressing for the Metanet.
- 4) The SORCS (Self-Organizing Recursive Systems) framework, a tool kit which supports applications that organize themselves "on demand" from lower-level components. The central elements of the framework are the *environment* providing background tools, *objects* or system components, and *catalysts* which cause new synthetic objects to compose themselves from other objects. The framework relies on Metanet capabilities to support effective utilization of the environment and to support composition of heterogeneous application components. Specific technical innovations needed to enable SORCS include: interface language (e.g. IDL) extensions for describing and reflecting on components; region-based search strategies supporting our multi-dimensional model of location and locality; mechanisms for self-indexing, the construction of search indices from the "inside out"; and authentication technology for decomposed and cascaded modules, including the concept of the "synthetic principal".

B. Technical Rationale

We begin by describing the Metanet, a new network architecture based on the concept of the region as a first-class element in the network. Regions and boundaries within the Metanet are one example of a network mechanism concerned explicitly manipulate position and location information within the network; we consider this class of mechanism in more detail. We turn next to a discussion of providing network resource management and quality-of service control in a robust, scalable way. The rationale concludes with a description of our strategy for constructing self-organizing "transient applications" using future large-scale networks, based on a framework closely integrated with the capabilities of our network architecture.

Regions

Today's Internet is conceptually built on the model of uniform end-to-end connectivity, with routing providing direct paths between any set of end-nodes. In reality, the global network consists of many distinct regions interconnected with varying degrees of trust and cooperation. This difference between theory and practice has led to a number of ad hoc creations such as firewalls and NAT boxes. This is an unfortunate situation. The very elements that match the Internet architecture to real world administrative, security, and policy enforcement requirements, and thus make the net actually useful, are add-ons rather than part of the architecture.

We argue that a new architectural component, the *region*, should form a central building block of the next generation network. In this model, the network is fundamentally modeled as a composed set of regions. As a common element of the architecture, all regions share a set of basic properties, operations and responsibilities. The basic operations of the network - data transmission, resource management and control, routing and addressing, security, management and configuration, and pricing - take place in this context. Our challenges are to identify and justify the core properties of the region, to recast the successful architectural principles of today's Internet to this new network architecture, and to show that this new architecture supports networks which are more robust, scale more gracefully, and are easier to manage and deploy than the existing Internet.

A region is a partition of the network which models an externally defined or imposed structure. There can be many sorts of regions at the same time - regions of shared trust, regions of physical proximity (the floor of a building or a community), payment zones and administrative regions are examples. Within a region, some particular condition is assumed to hold, and algorithms and protocols may make use of that assumption. The region structure captures requirements and limitations placed on the network by the real world. The architectural principle that regions are a fundamental part of the network is captured in the basic functional specification of a region. Beneath this model, a wide variety of implementation techniques will be required.

The adoption of regions as a structuring principle implies that network operations and protocols know and make use of location information in the network - the "question of where". This point, and the mechanisms that support it, are discussed at length in Section [QoW]. We touch on some basic aspects here before moving on.

¹This does not imply that the assumption is always correct. Where secure and correct operation of the network depends on the validity of the assumption, it may sometimes be necessary to verify this validity before proceeding. This does not negate the value of the region, just as the fact that cache entries sometimes become stale does not negate the value of a cache.

Two concepts are closely associated with regions; boundary crossing and membership or containment. The first is tied to connectivity and routing. Even in a network of regions, the goal is end-to-end communication. Paths of communications must thus be established in a mesh of regions, which implies passing through points of connection, called *waypoints*, between the regions. Concrete examples of waypoints include firewalls, proxy servers, and accounting meters. In order to incorporate waypoints into the architecture, the routing algorithms must now guide data through the necessary waypoints, and insure, to the trust level required by the customer, that the transmitted traffic has in fact done so. As an example, the crucial issue with a firewall is to insure that it has not been bypassed. This is done with ad-hoc methods today. Explicit network-layer support for waypoint location and traffic routing enables algorithms and protocols based on trusted action at region boundaries to become an integral part of the system. One such class of algorithms is discussed in this proposal's Section [QoS], covering a new model for QoS support.

The second concept defining our region architecture is *membership*. Because all of the members of a region share a common property, it is frequently useful to know whether a node is a member of some region. A trivial example of this is the packet transmission process in an IP host, which proceeds differently depending on whether the packet's destination is within or outside of the region defined by the host's local subnet. A more powerful example is found in the relation between regions and search operations. In a global network, most search operations are bounded or <u>scoped</u> to limit the field of search to relevant material. Because a region defines a set of objects with a like property, a region may be used to scope a search. Frequently, a region is an ideal candidate to scope a search, because the region reflects an external structuring that creates a natural constraint boundary for the search.

This suggests that the action of performing a search with the scope specified by a region should be supported efficiently within the network. However, the management of scope and search may become quite difficult in general. For example, regions that are contiguous in network-connectivity space are often efficient to search, but regions that are not (e.g.; locate all the computers in Cambridge, Mass.) require much more effort or pre-planning to carry out. Further, searches are often carried out looking at multiple constraints at once, which (in principle) leads to computations that are very complex and costly. But as a practical matter, scope is often expressed in ways that can naturally lead to efficient parallel strategies such as broadcast or multicast search.¹

The Metanet: a network of networks with qualitative heterogeneity

This section describes the Metanet; a new architecture for providing communications services in a regionalized environment. Three ideas form the core of the Metanet: a routing and addressing system designed for a region-based network, end-to-end communication semantics based on logical, rather than physical data formats, and the incorporation of data format translators at region boundaries in the network; controlled and accessed by applications through the routing architecture.

Today's networks interconnect by sharing a coherent low-level transport model - the Internet is defined by the universality of the IP protocol layer. But the uniformity of the IP layer is not a critical issue to the user constructing a distributed application. What the user cares about is more basic - how the communicating parts of his application name and locate each other, the level or quality of service he can expect to obtain, and the relationship between his application's content as it enters the network and as (perhaps transformed by format conversion or computation within the network) it leaves.

¹It is interesting to note that the concept of <u>administrative scope</u> has recently been added to the IP multicast protocols. Administrative scope is an early example of network protocol support for abstract, externally imposed regions.

In fact, data need not be carried in the same way in different parts of the network - any infrastructure which meets the user's requirements with high confidence can be used to construct a coherent application. Packets, virtual circuits, analog signals, or other modes, provided they fit into a basic service model, are equally suitable. The overall network may contain several regions, each defined by the use of a specific transmission format. For example, a voice call might be carried over a telephone infrastructure in one region and a packet infrastructure in another.

This use of heterogeneous lower layers in different parts of the network may offer powerful benefits in efficiency, deployability, adaptation to local conditions, and cost. The diversity enables the use of highly-tuned special purpose physical infrastructure in regions where performance (speed, low power, weight) or economic requirements warrant, supports the graceful integration of legacy infrastructure into an evolving system, and simplifies the use of any available technology in time of need or crisis.

We call a network which builds coherent user level semantics from a regionalized infrastructure and potentially employing qualitatively heterogeneous low level communication technologies a <u>Metanet</u>. We describe three essential aspects of the Metanet, addressing and routing, the use of a logical rather than a physical data unit, and the implications for statistical multiplexing and congestion control.

Position of addressing and routing in the architecture.

Metanet addressing, routing, and resource management architecture explicitly recognizes that the network is a collection of <u>regions</u>, rather than a single whole. It supports location and routing across specific intermediate points in the network, as well as between end-points. This implies that regions themselves, as well as hosts and routers, are first class objects within the Metanet routing framework. The concrete result of this is that Metanet regions have addresses and names. Routing explicitly manages the traversal of data format converters, security boundaries, metering and accounting units, policy modules, and other interior devices, as well as supporting end-to-end connectivity. The specifics of an addressing format are not of great concern to this proposal, as long as it is suitable for use with the abstraction map mechanism described in Section [QoW]. Our development work will likely be based either on IPv6 or the Nimrod address format, to be determined after an initial evaluation.

Requirements for a logical internetworking unit

In the Metanet, data may take different physical form (packets, cells, radio signals, analog voltages) in different regions of the network. Conversion between physical formats occurs at region boundaries. Our previous work on Application Level Framing demonstrated the value of an internetworking unit based on application semantics. We now must extend the ADU concept, which was a primitive form of logical internetworking unit, to encompass non- packet nets. A possible approach is to consider the "shape" of application communication behavior, without concern for the exact semantics. Virtually all application behavior can be expressed as either streaming, call-response, or message semantics. Of these, call-response is a stylized form of message, and not fundamental. This suggests that the Metanet logical data types should be messages and streams.¹

The internetworking data unit serves two functions; it is the fundamental unit of data transmission for the application, and it is the unit on which core networking functions of network resource management and congestion control operate. Our task is to identify logical data units which meet application and network requirements, and to express the core network functions in terms of these units.

¹Note that this question is separate from the communication pattern, such as unicast, multicast, anycast, incast, broadcast, etc.

Congestion control

Congestion avoidance and control is a fundamental requirement of any network architecture¹. A congestion control algorithm must predict or detect points of congestion, and feed information back to data sources sufficient to allow them to adapt to network capacity.

When congestion is predicted or detected, it must be managed by a combination of allocating resources to traffic, slowing the generators of traffic, and shedding excess load. A variety of congestion prediction and detection techniques known today can be applied to the metanet. However, a form of congestion detection based on statistical analysis of data is a natural outcome of our proposal for robust scalable QoS management, described in section [QoS]. We expect to utilize this approach our instantiation of the Metanet.

The principal remaining issue is that of providing feedback to generators of traffic. Although today's Internet protocols equate packet loss with congestion, a similar approach will not work in the metanet, because there is no sufficiently common unit of loss across multiple technologies and logical data types. Instead, our likely solution is to develop an abstract model for congestion feedback notification and traffic source response, which is then mapped into concrete behavior as required by a particular technology. This allows the same abstract mechanism to control, for example, packet generation in a PC and channel allocation in a spread-spectrum radio.

The Metanet Framework

The Metanet unifies and motivates the technology development described in this proposal. Support for scalable location information and intermediate point routing is discussed in Section [QoW]. Section [QoS] describes a scalable, robust approach to providing QoS control in a regionalized network architecture. A key aspect of our work will be to identify design principles that give the metanet flexibility and robustness. These principles are analogs of the Internet's structuring model of soft state and "fate sharing"; two highly successful design principles derived as consequences of the well-known end-to-end argument. Our goal here is to identify design principles for this new network architecture which are equally grounded in a proven architectural framework.

Location within the net: the "problem of where"

The core function of a network is data delivery. Beyond this lies the possibility of higher-level services that the user wishes to invoke inside the network, as with advanced service networks and certain active network designs. Placing function "inside" the network necessarily implies a concern about location within the net, because if the computation could effectively be positioned at an arbitrary location, it might as well be located at the user's end-point. Research described in this section addresses the problem of creating, managing, and using location information.

Establishing concepts of location

The meaning of location in network space is complex, because the definition has many dimensions. An application looking for a "nearby" resource may mean nearby physically, administratively, or in terms of available bandwidth or low latency. Finding the "shortest path" to another node may mean solving a complex multi-parameter optimization problem, rather than a simple lookup. Methods for recording, storing, or utilizing locational information in the net must address this difficulty. Traditionally this is done either by precomputing enough information to meet requirements, as with routing protocols, or by a multicast-search strategy, as with ethernet ARP. In a complex multiparameter environment these techniques are frequently inadequate. We propose a new technique; the <u>abstraction map</u>.

¹Other than one which handles only constant-bit-rate traffic using pre-allocated resources. This network can be considered to use null congestion-control, since it can be proven to never be required.

The abstraction map descends from map-based routing architectures such as Nimrod. The important difference is that maps are partial and generated on demand, based on information provided by the querier. This *late binding* of user requirements to a map allows us to efficiently support location in multi-parameter spaces without unmanageable overhead. A realistic system will support optimizations to further reduce overhead and improve robustness.

Intermediate-point routing

Traditional network routing has been concerned with computing paths between end-points. New architectures such as the Metanet require the routing function to address transit points as well. Two additional issues arise.

First, the low-level data forwarding protocol must be able to cope with waypoints; there must be a way, using either data packets themselves or a flow setup protocol, to control the path as well as the destination of transmitted data. Today's networks typically claim to support this capability but some impose a significant performance cost when it is used. An effective engineering solution is required. A design well-matched to our abstraction-map routing might be based on an efficient source routing engine, or on an encapsulation or labeling strategy. We expect to explore both approaches, using existing standards and capabilities when possible.

Second, the network infrastructure must *certify* to the customer's satisfaction that the required intermediate waypoints have been traversed. The choice of mechanism depends strongly on the trust relationships within the system. If the trust relationship is between the waypoint and the end-node, then a *wrapping* approach, in which the waypoint signs the packet, may be used. If the trust relationship is between the end-node and the routing infrastructure, the end-node must be assured that the routing infrastructure is functioning correctly and not compromised, and the routing infrastructure must assure itself that nodes and regions it controls are identified correctly and not supplying corrupt information. This is <u>decentralized provably verifiable routing</u>. We are at present able to identify some of the characteristics of DPVR, but believe that significant further research is required before the capability can be deployed in any practical way.

Parameterizing the network characteristics

Today, routing protocols select routes based on hop count, perhaps qualified by a single weighting metric often called *cost*. Cost can be used to model any single measure of link quality, either actual cost, or link loading, delay, capacity or whatever. Routing based on a single metric is well understood, but once the network starts to offer a more complex range of service qualities, it is possible that more than one metric is needed to describe the relevant link characteristics. Finding optimal routes in a multi-dimensional space is a complex problem, both theoretically and practically. Thus, in solving the question of where, we need to find useful abstractions and approximations that let algorithms understand the characteristics of the network over which it is operating.

Optimizing from two perspectives

Traditionally, one thinks of optimizing the operation of an application, by configuring it to use the networks resources efficiently. However, there are actually two optimizations involved. From the perspective of the application, the goal is to obtain best practical performance for a set of end-to-end data flows. From the perspective of the network provider, the goal is to allocate the resources at a particular point in the network among all users in a rational way. The final allocation of resources in the network must balance these two potentially competing requirements. If our abstraction map is to be successful, it must thus capture both perspectives -- the application and the network provider.

In a commercial network, the network provider may encourage the efficient use of its resources by charging for them. The application then selects what resources to use, taking into account both the cost and the resulting performance. However, it should be recognized that cost, as an expression of resource consumption, is an approximation, and sometimes a very imperfect one. Further, in

some networks, such as military networks, there may not actually be a charge levied for the use of the net. None the less, it must be a goal of an overall design to permit (and encourage, if not force) applications to use resources efficiently, as well as optimize their own performance.

Way-points in the metanet

The placement of waypoints in the Metanet provides a test environment for our development of technology for utilizing location information in the network. Way-points serve a number of purposes. One purpose is to define a "meta-routing" point, where the representation is transformed as data flows from one part of the metanet to another. If we assume that a route has been computed by some means, the question of where reduces to finding locations along that route at which special actions need to be taken. The location of this way-point is, to some extent, determined naturally by the physical characteristics of the metanet topology.

In other cases, the way-point provides a location where some application-level service can be installed, either to deal with the need for a different data representation in different parts of the network, or to deal with the need for data caching or replication to enhance performance and robustness of the application. The point to note is that desirable locations for installing the way-point can depend strongly on the purpose for which the way-point is being installed. Choosing a correct location for the way-point may depend on access to a significant amount of detail about aspects of the network. This access can be provided by the abstraction map, and the way-point located and activated through intermediate-point routing.

Finally, the waypoint may be providing an administrative or security function at the boundary of a region. Requirements imposed on these waypoints stress the certification and verification components of the intermediate-point routing architecture.

System-level architecture for quality of service control

Quality of Service control is the task of managing shared network resources to deliver service commitments to end users. It is a fundamental function of any network element.

The traditional model of QoS control is that resources at each switching node are allocated to a connection using a setup protocol, and the packet scheduler uses these reserved resources to assure the desired QoS. Recently, this model has been extended (by the DARPA-funded Integrated Services and RSVP work) to incorporate multicast, receiver-oriented reservation, and soft state, which significantly improve the scalability and robustness of the basic model.

We observe, though, that this model has limitations. Use of a setup protocol is appropriate when allocation is required for short-lived flows, but long-lived resource allocations are better expressed through network management. At the same time, the setup protocol may not be dynamic *enough*. When very short sessions are to be supported, the overhead of a session setup is too high; the only practical mechanism is to put the QoS information in the data packets themselves.

A second and more distressing observation is that current QoS control technologies may work against network trustworthiness and robustness, because today's mechanisms create an unfortunate tradeoff between scalability and efficiency on the one hand and strong assurance of performance on the other. Technologies such as aggregation that are more efficient (and thus cheaper) under normal circumstances are less resistant to failure under extraordinary circumstances, such as failure of physical resources or attacks on the network. This is because of a basic tradeoff between aggregating information in the network, which is necessary for scalability, and treating each application session separately, which allows the network to monitor and defend correctly behaving sessions against rogues.

We propose to address these concerns with a new system model for QoS control. The model is based on two technical innovations. First, we cleanly separate QoS control mechanism from policy through the use of QoS Building Blocks (QoSBBs) at each switch node, controlled by policy modules located (only) at crucial control points such as end-nodes and administrative boundaries. Second, we suggest that a class of resource management algorithms based on

statistical analysis and adaptive behavior can simultaneously benefit from aggregation in the normal case and defend against failures and malicious attacks. When such algorithms are used to implement the QoS building blocks, the overall system is responsive to users needs and changing conditions with extremely high confidence.

OoS Building Blocks

In the QoSBB approach, primary switching elements within the network replace their fixed-function traffic scheduling algorithms with a lower-level set of carefully defined composable primitives; the QoSBBs. These QoSBBs do not themselves implement a particular packet scheduling discipline, but instead are composed to do so based on information contained in the arriving packets. The QoSBBs at a node thus form the primitive "instructions" of a somewhat programmable packet scheduler. The same QoSBBs, implemented in highly efficient hardware within the network switching core, can provide a wide variety of QoS control services.

The behavior of the QoSBBs at routers within a region of the network is determined by control bits in packets that are set by boundary elements that understand the resource management policy used within that region of the network. These elements may in turn may be following the instructions of a setup protocol, network management, or a sufficiently aware application. This choice is transparent to the QoSBB. Since the QoSBB takes its "instructions" only from the arriving data packets, rather than from locally stored state, it is perfectly scalable. Further, the "programming" can be changed as the packets cross regional boundaries within the net, implementing different policies in different administrative regions.

Examples of OoSBBs

An obvious example of a QoSBB is a collection of queues that are scheduled and managed in different ways. For example, a queue the use of which is governed by admission control provides a form of *real time* service similar to the Controlled Load service proposed for the Internet by our group. Such a queue might also have an aggressive dropping policy to insure a short maximum queue length. In contrast, a queue that is governed by admission control, but which accepts a much larger average and maximum queue length is suitable to build a service similar to the CIR capacity service of Frame Relay.

Another sort of QoSBB is a dropping policy, in which each individual packet carries a flag indicating whether this packet, in times of congestion, should be preferentially dropped or preserved. Simple dropping of packets from a real time stream can allow the less important parts of a hierarchically coded data stream to be stripped off when encountering a point of reduced bandwidth in the network.

The most interesting QoSBBs may look quite different from traditional packet scheduling functions. Our Expected Capacity scheme uses a QoSBB based on statistical sampling of packet load and adaptive controlled dropping to allocate different amounts of capacity (bandwidth) to different traffic streams based on policy profiles set at the edges of the network. The dropping policy QoSBB used here can be used as well for selective real time dropping. As this example illustrates, one QoSBB (dropping preference) can be used to build up different services with very different overall qualities. The relationship between how individual packets are treated and the resulting overall service is sometimes not obvious. A critical task within our proposed research is to identify approaches for computing the required policy module behavior ("packet programming") within a system starting from the available QoSBBs and the desired user QoS.

An additional challenge when creating QoSBBs is to define them in such a way that they do not demand exactly one implementation approach. As an example, consider scheduling in a serial link and a shared access LAN or wireless link. In the first, the queue management, dropping preference, and admission control functions may be implemented in a centralized way in the serial link driver. In the shared access model, the algorithms that determine access to the medium are distributed at each of the access points, and these must work in concert to achieve the same effect. We believe, specifically, that the two QoSBBs described above can be implemented in either

manner. For example, our approach to drop preference is a probabilistic algorithm driven by queue length, but queue length can be approximated by observing channel loading, so by observing the total traffic on the shared medium, each access point can implement its part of the dropping algorithm.

Boundary Tagging

In the scheme as described, the packet carries in its header the indication as to what QoS processing it requires. Mechanically, this is coded in a few bits that instruct the various QoSBBs how to process the packet. This approach leads to very efficient packet forwarding engines, since the router need not keep track of specific packet flows, but instead can just process each arriving packet according to its contained QoSBB control bits. However, these request bits in the header must be set in a way that can be trusted, and users who request a premium service will need to be charged for this service in some way.

We assume that the Internet is composed of regions of shared trust -- often regions under one administration such as corporate campuses, networks operated by single service providers, and so on. Not every router inside such a region needs to validate the service request that the packet is requesting. Instead, modules at the edge of such a region can examine the packet flows, validate that the service requested is legitimate, and generally enforce policy concerns of various sorts. Our architecture thus has policy modules at the boundaries of trust, and efficient forwarders in the interior of regions. We call this architecture boundary tagging, because the policy module can both validate the QoS tags in the packet, as well as modify these tags to adjust the service that a given flow receives inside any particular region of the network.

Aggregation and robustness

To make the system robust against attack, the collection of QoSBBs at each node must monitor their own behavior and take defensive action when they detect a problem. The behavior monitoring function can be performed by adaptive random sampling of the traffic flow. Under normal conditions, sampling behavior is minimal, in order to limit the use of resources. If a problem such as unexpected congestion is detected, the sampling is progressively refined by creating trial filters until the source of the traffic is localized. One obvious trial filter is to separate the traffic arriving on each physical input link, which permits the router to detect if one or another physical link is a source of overload. If one link is misbehaving, this information can be passed upstream along that link to the next router, and this process can recursively repeat itself until the actual source of overload is isolated. If, somewhere in the network, an incoming link is isolated that cannot in turn regulate its offered traffic, the router may then simply discard all the traffic from that link.

Other sorts of trial filters can be imagined, which separate the traffic into different partitions in an attempt to isolate the flow of packets that is overloading the router. If a trial filter is identified that yields a well-behaved subset of the traffic, the subset can then be forwarded separately while the source of the overload is further refined. At the same time, information can be sent towards a policy module, which can take more intelligent defensive action. This "isolation on demand" technology provides the scalability of aggregate approaches with the trustworthiness and robustness of strict per-flow resource management.

The interrelation of congestion control and OoS management

Congestion, overload due to the shortage of a shared network resource¹, is an unavoidable consequence of statistical sharing. Its control is thus a fundamental part of the both the Internet and the Metanet. Any congestion avoidance and control mechanism has several components. First, there must be a signaling mechanism used to push back on a flow that must reduce its rate. There is

¹Congestion is usually viewed as occurring when offered traffic exceeds available link bandwidth. However, undersupply of other network resources such as router CPU capacity are sources of congestion as well.

much current work in this area, including our recent redesign of the TCP "fast retransmit" algorithm and various proposals to add explicit congestion notification to the internet protocols.

However, the more difficult aspect of congestion control is the algorithm that determines which flows must slow down. We view this aspect of congestion control as inseparable from the problem of maintaining a particular quality of service. This suggests that congestion control and avoidance be tightly integrated with QoS control and management, perhaps using the same overall signaling scheme.

Our current work on expected capacity allocation takes this approach, utilizing a control algorithm which permits different flows to achieve different utilizations in a controlled way during congestion. ATM's ABR service has a similar high-level goal, though the details and technical approach differ widely. We will continue to merge our work on QoS together with congestion control mechanisms to create a single, unified congestion and QoS management architecture supporting the operation of QoSBBs in the Metanet.

Self-Organizing Recursive Systems

A network exists to support its applications. We argue that it is necessary to evaluate a network architecture in light of the applications it can support, and that the capabilities of the Metanet regions, waypoints, and abstraction maps - make it easier to construct distributed applications and enable new classes of applications that cannot reasonably be constructed in today's networks. We propose to validate this claim by tightly coupling our network research to a careful reconsideration of traditional models for creating distributed applications.

To build robust, multi-scalable systems, one must reconsider the central problems of building and supporting distributed applications: publication (the problem of making known any potential components of an application), discovery (the problem of finding the appropriate components), composition (the problem of merging the components into a new application or higher-level component), and management (the problem of making sure that the composed application is performing appropriately). We see four trends that will have significant impact on the next generation of such applications: growth in the number of potential components; an increase in the range of components that will be available; the mobility and volatility of potential components such as applets, agents, and avatars; and a potential decrease in the functionality or capabilities of lowest-level building blocks, such as sensors and other small, inexpensive embedded devices. These trends challenge the scalability and flexibility of existing networks. We argue that the Metanet's emphasis on regional coherence and dynamic adaptation to a changing environment overcomes these challenges. In this context we propose the Self-Organizing Recursive Systems or SORCS toolkit.

We describe the applications addressed by SORCS with two examples. The first is taken from Computing and Communications in the Extreme: Research for Crisis Management and Other Applications, the DARPA funded report by the Computer Science and Telecommunications Board of the National Research Council. A number of the participants in that study emphasized the need, whether in a military or civilian crisis for an electronic "anchor desk" or control center as an assistant to the humans organizing and orchestrating the response to a crisis. Such a desk would present to the human a developing and mutating suite of informational and predictive tools. These might include the results of intensified sensor collection and data analysis of specific aspects of the situation, resource (human and non-human) requisition, deployment and tracking, simulation prediction, etc. Each of these sorts of facilities may itself have been created independently in response to the crisis, and be corralled together to participate in the anchor desk.

The second example is much simpler, but has enough of the same characteristics that it exemplifies a potential demonstration of the SORCS architecture. In this case, we postulate a small set of composable elements, the lights and light switches in a house. The homeowner buys a new lamp and switch, introduces them into the home environment and then links them to each other, so that the switch is the control interface to the light. The home also includes a power management service

that monitors all electrical appliances in the house. The homeowner also carries a PDA with wireless network connectivity. This allows the homeowner to turn on and off lights either locally or remotely, by creating a light control application that includes the PDA as its user interface. The power management service may also turn certain lights (for example external high wattage spot lights) off as needed, as well as other appliances such as the hot water heater.

The central features in both of these examples are: an <u>environment</u> that provides a core set of services, specific <u>components</u> being brought together, a <u>catalyst</u> that is the mover and shaker, dynamically causing a new object to come into existence, and possibly a <u>dynamic monitor</u> that provides a trouble-shooting function. These are the key concepts of SORCS.

The environment captures the information and tools available to the catalyst. These tools may include services such as computing, name translation, storage, caching, printing, authentication, etc. Together with these tools is location and usage information for each, which may be mapped into the Metanet through the use of regions and abstraction maps. The environment serves as the "primordial soup" when composing the application.

The catalyst is the second element of our facility. It is not the new resource being created, but rather a "trigger" that will cause elements from the environment to come together and the new resource to be born. The catalyst uses both search indices and active discovery to discover the components to be brought together. In some cases, the components may not provide exactly the necessary interface; to solve this problem, the catalyst may require from the environment a wrapper or some glue to match interfaces. The most intellectually challenging functions of the catalyst are the synthesis of a new principal and passing along the "breath of life".

The concept of component objects is not new, but objects in the SORCS model will need to provide certain novel functions. Of these the most important are self-description and what is called retrospection or introspection by different communities. The function needed is the ability to expose and know one's own role or functionality. This is central for two reasons. First, we have postulated that in order to avoid scaling problems in indexing, self-indexing will be necessary, and in order to self-index an object must have some self-knowledge. Second, because we must assume that the functionality of objects may evolve with time, once a catalyst has determined the subjects of its composition, it must be able to ask them about their interfaces, in order to requisition any necessary wrappers or glue. Primitive retrospection interfaces are beginning to appear in today's component software models, but most are targeted towards direct use by humans. We require that the information be computer-oriented, to permit effective self-organization.

Several research problems are raised by the SORCS application model. Two which we expect to address in our work are the need for efficient indexing and searching and methods for dynamic monitoring of composed components.

Indexing and searching play an important role in any distributed composition problem, but problems arise with scaling. In particular, indexing will need to be automated and searching will need to be limited by scope. The Metanet region allows SORCS-level requirements to be reflected cleanly into the network infrastructure. SORCS relies on the postulated capability of the Metanet to optimize searches conducted with regional scope.

Recall that a region is a partition of the network based on a set of externally defined conditions or characteristics. It reflects not only end points, but also may reflect qualities of the "connections" among them. For example, in a region in which a certain set of security constraints is defined, it may be assumed that those constraints apply among the members of the region. An index reflects characteristics. It is not necessarily the case that because two nodes have the same security attributes in an index that the communication between them does also. In order to control the time and cost of searches, scopes for searching will be critical. A scope may be as simple as restriction to one or a set of orthogonal regions, or may be more algorithmic. We expect to begin our work with simple scoping. In terms of implementation, because regions will often define policy boundaries, indices may either be defined along those same boundaries, or if they are large, they

may be partitioned along regional boundaries. In our example, indices of interest to the anchor desk and its subsidiaries might reflect sources of information about human and non-human resources (e.g. how to find portable toilets and the likes, or databases for estimating populations and their behaviors), large-scale simulation capabilities, etc. In contrast, the light controlling application created on the PDA may need an index that translates from street address to TCP address. In both cases scoping bounds the effort and cost of the mapping, and efficient regional scoping conducts the search in a way which optimizes the use of network infrastructure.

The application function of dynamic monitor is directly analogous to the monitoring of congestion problems in our QoS research, described previously. A dynamic monitor may be closely related to a catalyst. They both have a model of the nature of an object. The catalyst attempts to cause it to come into existence, while the monitor checks on the continued correct behavior based on correct behavior of components. As in the QoS control approach, it will have a toolkit for discovery, isolation, and perhaps feedback to misbehaving components. In our example, the power manager itself is such a monitor which includes input from other sources, such as the power grid. In the anchor desk, a monitor might both check on existing synthesized objects, but might also monitor human behavior as an indicator. If the human is making too many requests for additional information, the objects may not be providing enough or the right information for the situation and correctional behavior might be indicated.

The uniqueness of this work lies in the self-organization nature of the relationships to create and manage new objects and the flexibility in recursively using objects, by allowing them to play different roles in different relationships. To this end we can identify a number of specific components to this research:

- 1. Definitions of the functions of the environment, the catalyst, component objects, and monitors. In the Information Mesh, we postulated a type model based on roles, in which objects could play multiple roles, and the set of roles that an object played might vary. By defining these functions as roles orthogonally to other functionality of an object, the object can move in and out of different relationships with other objects.
- 2. Principal synthesis. The problem here, as discussed above is the creation of a new identity for the composite object, with appropriate privileges and restrictions.
- **3. Introduction of new objects** into an environment. This is important not only for synthesized principals, but also for newly imported static elements, such as new light lamps or portable seismic sensors. How does such an element get an identity and how is useful meta-information about it captured either for inclusion into regions or for self-indexing?
- 4. Self-indexing. Although this is suggested in the previous point, there can be a separable effort to address the problems of self-indexing. One can postulate that self-indexing may occur at other times than just at introduction as a new object. In order for an object to insert itself into an index, it must make a choice about indexes. It then must learn the information required by the index, collect or generate that information and then submit it to the index.
- 5. Attaching small objects to the net. The problems here have to do with how to put orders of magnitude more, very small, very inexpensive elements on the net. If every light and light switch in every home and office is to be "on the net" in some fashion or other, it is undesirable to expect that each such object will have all the function of a large computer. We must identify what it means for them to be "on the net". Are there high level functions that they must support, functions such as answering to a global identifier, supporting introspection, self-indexing, interpreting constraints on access, etc.? Do we imagine that there may be proxies or gateways that provide those functions on their behalf? That may be adequate for other composite object s that wish to include them. But what do they need to know about themselves and the other objects with which they collaborate? What high level protocols will they require? Will they live in a private world, inhabited by their local community (all the lights and switches in my house) into which projections of a small number of external objects are injected? If so, how might this be handled.

In short, we will be examining the question of what is in the Metanet and what is above or below it.

Conclusions -- network scale and robustness

The various mechanisms and architectural principles that we have proposed here play a number of fundamental roles as we conceive the next generation network. However, we wish to stress the two factors that we identified earlier as critical motivators for our research -- multi-scale networks and robust networks.

We address the issue of scale in a number of ways.

- Our approach to Quality of Service, which permits the aggregation of flows at a router, avoiding the need for per-flow classification, and localizes policy and enforcement to specific points at the edge of regions of shared trust, is a key to building high-performance routers for very large networks.
- The concept of *region*, as articulated in our *metanet* architecture, is key to large-scale networks. Only by explicitly recognizing this architectural component can we bound the scope of network management functions, propagation of errors, global visibility of local change, and so on. Abstraction maps, and the various abstractions of physical network attributes on which they are based, are another key to scale, since they allow one to view the network at a reduced level of detail when warranted.
- Our concept of self-organization of components within a region is an example of the use of scope to bound an activity that thus becomes more practical. Precisely because we limit the scope, the action can become automatic, which in turn permits building a larger overall network. It is the network equivalent of "think globally, act locally."

We address the issue of robustness as follows.

- Our core observation is that the objectives of efficiency and scale can work to the detriment of robustness, because they encourage such actions as aggregation of traffic, which makes it harder to isolate malicious sources. We thus look specifically at the practice of traffic aggregation (which we believe is essential) and assert that any such mechanism must include isolation of sources as well as aggregation of them as needed as an intrinsic pair of tools. Our QoSBB approach provides this pair of options, with the isolation normally happening only at the region boundaries, backed up with local options as necessary.
- Our approach to answering the "question of where" provides comprehensible visibility (i.e. abstracted to a reasonable level) to the current operating conditions of the network. It permits the application (or a optimization module running on its behalf) to view the network, not as a black box, but a resource that can be utilized taking into account both the needs of the application and the network operator.
- In one respect, the issues of scale and robustness are the same. Large scale systems cannot be built at all unless they are intrinsically robust, lest they fall of their own weight. Thus, tools such as regions add to the robustness of the network, because they add the necessary isolation that limits the number of global interactions.

Finally, our *metanet* concept represents a basic advance in the articulation of network principles. While holding on to those architectural features that are truly fundamental, such as global addresses and statistical multiplexing at a fine grain, we abstract more cleanly than previous architectures such mechanisms as congestion control, control formats, and administrative (and other) regions within the network. We preserve ideas such as the end-to-end principle, but recast it in terms of the abstracted application entity, rather than in terms of a physical end node. In doing so, we permit a greater range of realization of function, ranging from end-node composition to network representation of data.

C. Results, Products and Transferable Technology

Results

This section presents the specific results we expect to produce as the research described in this proposal advances. At the completion of the proposed contract period, we will have completed and demonstrated the key functional blocks of two significant innovations in networked systems:

- The Metanet an architecturally scalable and robust network architecture based on the **region** abstraction, **location information** as a basic network datatype, and resource management services based on **composable building blocks**.
- SORCS an architecture for self-organizing, recursively composable applications, which organize themselves from networked lower-level instances in the presence of catalysts.

For purposes of presentation, our expected results are divided into four broad categories.

Robust Region-Oriented Quality of Service Mechanisms

Our exploration of design strategies for flexible, inherently robust Quality of Service provisioning will, if successful, lead to a number of different results.

- 1. We will identify a range of QoSBBs, beyond the examples used in this document (selection of service queues and dropping preference). We imagine that the set of QoSBBs, while not large (remembering that what routers do is very simple), can reflect a creative use of QoSBBs to construct a range of services.
- 2. We will demonstrate commercially viable QoS provisioning based on an architecture containing QoSBBs at switches within the core of a network region, controlled by QoS policy modules at region boundaries.
- 3. We will demonstrate the isolation of misbehaving sources from within an aggregated scheme of traffic. We intend to develop at least two algorithms. One is based on modeling the traffic from each input port, and pushing back recursively across the net until the source is identified. The other uses probabilistic trial filters to cut the aggregated traffic into disjoint partitions, with the hope of finding the malicious flow by a series of divisions.
- 4. We will, if necessary, define new higher level service models intended to match evolving application demands and economic constraints. To this point, we have proposed real time services -- the Controlled Load service, and advanced best efforts services -- the Expected Capacity service. We anticipate that new services will be required as we better understand the performance requirements of new classes of applications.

As a part of this project, we will develop a prototype demonstration of our QoSBB scheme, using the CAIRN testbed or other suitable development environment. We will use routers at the edge of the CAIRN network as points of policy enforcement and traffic tagging, while the routers within the network classify and forward the traffic based only on traffic tags. We will demonstrate several modes of QoS assignment. In one mode the source will encode the desired service in the packet header, and this request will be policed at several points inside the net, including within his machine. In the other mode, the assignment of the QoS will be made at a boundary point, and will

be derived from a setup request made by the application or by a management protocol. Finally, we will augment the demonstration routers with the ability to monitor their own delivered service, protect themselves against offending flows, and push control information outwards to the policy modules to further isolate these flows.

While this will be implemented within the CAIRN environment, a more compelling demonstration requires a larger-scale context, which we may create through CAIRN virtual networks or by simulation. We estimate that CAIRN virtual networks give us the ability to use CAIRN as an emulator for a network of roughly four times its size. As this is still somewhat limiting, we will also develop simulation models and technology adequate to verify the behavior of our approach in a network containing thousands of nodes. As a side result, we will incorporate our simulation models into a publicly available simulator such as NS, for use by other researchers.

We have briefly considered the problem of getting accurate traffic information for simulations of these algorithms. One approach would be to take a trace of traffic from a typical point in the Internet, and use our tools to isolate pre-determined parts of this stream, or to isolate malicious traffic that has been mixed in.

Location information in networked systems

This component of our work is based on two key results:

- 1. We will develop and verify abstract models of network topology and performance, designed to simplify the task of managing location information in a complex network. These models will be verified through simulation or analytical approaches, as appropriate.
- 2. We will describe and define the abstraction map, a mechanism for capturing complex details of a region in the network and making abstractions of those details available to distant clients. The abstraction map is a representation of network conditions sufficient to model a variety of questions, ranging from routing to placement of services within the network.

This baseline research, if successful, should yield a basic understanding of multidimensional abstract network representation, which can be embodied in new or existing routing protocols. Our plan is not to perform the actual integration of these ideas with currently operational routing software, but to demonstrate a proof of concept.

A specific result which we hope to obtain is to understand how the abstraction map can express the information necessary both the optimize the performance of the application and to optimize the use of network resources. Commercial networks are likely to use cost as a way to model resource usage. We wish to understand how effective a model this is, and how it can be used to model circumstances (e.g. military networks) where charges are not actually imposed for network use.

We propose to evaluate this part of our research through a number of specific test cases each of which, if successful, will lead to a result in its own right, as well as contributing to our larger understanding of the overall problem. Some test cases we have identified as promising are described below:

1. To illustrate the use of the abstraction map to guide an application in positioning services within the network, we identify a number of possible projects. An example is the use of an abstraction map to calculate efficient dynamic cache location. The SRM protocol from LBL, used in the wb tool, is an ideal candidate for this experiment, because it uses multicast for all of its protocols. This means that caches can be installed into an SRM session without any modification to the existing protocols. A new tool can be designed that requests an abstraction map of the multicast tree, performs an analysis, and installs caches inside the network as appropriate. The running

SRM session will automatically include this cache site in its multicast-based operations and see improved performance.

- 2. The use of abstraction maps for dynamic service location emphasizes the late-binding nature of the abstraction map concept. This requires the dynamic construction of an abstract map of performance between a specific client and a number of potential service sites. With the aid of a suitable abstraction map, we can develop a "smart" DNS or directory service that can use information about the performance/topology of the network to determine the best alternative to select from among a range of servers, based on the location of the requesting client.
- 3. A key use of abstraction maps is as a component of the intermediate-point routing system designed for the Metanet. This use, which couples the results described here with much of the rest of this proposal, is further discussed below.

The Metanet

A key result of the work proposed here will be the construction and demonstration of a small Metanet. The demonstration Metanet will implement the following concepts, described in the technical rationale:

- A region-based architecture, with regions as first-class objects.
- A uniform global addressing model, and a simple intermediate-point routing design based on abstraction maps.
- The ability to determine location and perform location-specific functions within the network, using on the intermediate-point routing design.
- The capability of supporting communication between end-nodes on qualitatively different network types, using data translators and intermediate-point routing.

This network will operate in a laboratory setting. Certain work items needed to achieve this result may be tested in a larger setting such as CAIRN. Achievement of this result is dependent on the successful achievement of a number of the other results described in this proposal. More detail about these intermediate results appears elsewhere in this section.

Self-Organizing Recursively Composable System framework

We expect a number of specific results to derive from the higher level result of an <u>Self-Organizing</u> Recursively Composable System (SORCS) architecture. In particular:

- 1. An architecture for recursive self-organization of network applications, based on the concepts of catalyst, environment and objects. Two aspects of this definition are:
 - Definitions of the roles of catalyst, environment, and component object. By doing this, we will be able to demonstrate that an object will sometimes be part of the environment, and perhaps at others be either a catalyst or a component used on the synthesis of new objects.
 - The definition of principal synthesis. A principal, once created, is a full participant in the network communication. A demonstration would be to permit a principal to be the end-point in a TCP connection or WWW interaction, even though it had no single physical location anchoring its address.

- 2. Interface definition for self-organizing elements. Our current strategy is to define extensions to the Java component architecture language in support of the non-human interfaces needed for self-organizing systems, in particular for self-description and introspection. This is also intended to allow for flexibility in terms of presenting either a user interface or back-end to other user interfaces. An interesting demonstration of this facility would be an object that could either provide its own user interface or "back end" itself to a WWW server, so that it can be accessed by a human using a WWW browser.
- 3. Mechanisms to optimize searching with regional scope. We expect to demonstrate self-insertion in indexes and scoping of searches on region boundaries. As a preliminary demonstration of self-indexing, we will extend the definitions of several sorts of services, so that they are capable of inserting themselves correctly, into a region-based index using the SRVLOC server location protocol [SVRLOC]. The advantage of this over many of the document indexing services is that they are based solely on textual content or at least textual descriptions, and we expect that many of the objects of interest will not be textual.¹
- 4. Dynamic monitors. Our first experiment with dynamic monitors will be the demonstration of a catalyst/dynamic monitor pair that will generate a route based on a set of abstraction maps and on some regular basis verify that the route continues to meet the criteria set for it. If not, the dynamic monitor will be designed to take appropriate action, depending on the nature of the change.
- 5. Support of very small "lightweight" hardware components on the network. We will demonstrate the catalysis of fully-functional objects from simple components not otherwise able to participate in the full protocol structure of the Metanet. Our demonstration vehicle will be based on a small collection of embedded-system microcontrollers as the lowest-level objects and one or more standard PDA's (e.g.; the Apple Newton MP2000) serving as catalysts and user interface controllers.

Transferable Technology Path

The technical outputs of our research program are academic papers, demonstration and reference implementations of the developed algorithms and technologies cast in software or constructed electronic components, and trained graduate students.

There are a number of modes for the transfer of our research results into the broader community.

We have traditionally made all successful software developments available both for research and commercial use, and have seen this approach lead to the start of two major commercial ventures. We will continue this practice.

A second path is through the introduction into the IETF and other standards bodies of proposed standards and working papers based on our work. Recent examples include our work on URNs and the Controlled Load real time service. We see a number of potential IETF contributions from the work proposed here, including new QoS definitions, new models of pricing, and specific proposals for enhanced link-state routing protocols that provide abstract parameterization of network characteristics.

¹ The goal of this research is not to study indexing and searching, but rather to understand the symbiotic relationship between regions and natural scoping as realized in indexing and searching.

Because of our well-developed direct ties into industry, we are sometimes able to influence directly the development of products with new features based on our research. This process, while unpredictable, will be exploited as appropriate.

The more general results of our research -- the overall understanding of network architecture and its implications for the broader picture of network evolution, have been incorporated into a number of widely read reports, including a number of important publications from the Computer Science and Telecommunications Board, with which we are closely involved. We believe that our involvement there has increased the direct transfer of our results and knowledge both into industry and government policy-makers.

D. Comparison with Other Research

Work related to this project falls in a number of areas.

Quality of Service and resource control—There is a great deal of work reported in the literature on quality of service. Relevant topics include definitions of service classes, specifically the work in the IETF and ATM communities, work on reservation setup, such as the RSVP, ST-II and Tenet protocols, and work on packet scheduling disciplines. Most of these schemes presume some sort of reservation protocol, or per-flow state in the network. Our approach, in contrast, permits aggregation of flows in the core of the network, and uses means other than per-flow state to insure performance in the face of abuse. Reservations, if used, do not need to result in per-flow state. The idea that router behavior is composed of very simple components has been articulated before. The packet scheduler proposed by Van Jacobson at LBL, the Class Based Queuing scheduler[FL95], is based on this concept. However, the behavioral elements there are different from the ones we propose, and the range of relationship between the per-packet behavior and the resulting per-flow QoS is not a developed concept.

Congestion control -- Congestion control is a component of QoS. In some cases, congestion control has only the goal of regulating an aggregate of flows. However, many schemes have the further objective of achieving some balance among the various flows at the point of congestion, for example the work on feedback driven rate adaptation done at MIT and elsewhere and now part of the ATM standard. Random Early Detection, or RED[FJ93], is an approach to congestion control that avoids per-flow state, relying instead on a random sample of the packet stream. It establishes the concept of statistical analysis as a basis for control. A proposed enhancement to RED, informally called the Penalty Box scheme, which has been simulated but not tried in practice, attempts to use this statistical sample of the stream to detect large consumers of capacity. Our proposal derives from this approach, but carries it further in several key respects, in particular the idea of enforcing explicit QoS by statistical sampling, the idea of co-ordinated recursive trial filters to identify the offending source, and the isolation of detected anomalies through upstream propagation of results.

Service location -- With the advent of the Web, there have been a number of practical approaches to finding the "best" copy of a replicated service. Examples include the current work in the IETF on resource discovery (which is limited in scale, and assumes a simple model where links are either up or down), the anycasting proposal, and various schemes for selecting from among a list of addresses. The paper of Guyton and Schwartz[GS95] provides a summary of this work, and describes two general approaches, one based on information from the network, and one based on testing the network using end-point initiated protocols. The latter approach has been preferred in the short run, because of the difficulty of modifying the network infrastructure. We propose to explore the other path, and ask what information the network could usefully export, if it could be modified at will. Our goal is to identify a good set of performance and topology abstractions that the network can compute and export, which the user can exploit to build useful service configurations. An example of the need to relate location in virtual and real space is the Avatar system, described by Lam and Reed[LR96]. To understand the behavior of WWW traffic, they present a visualization of the flows in the context of the geographic location of the traffic sources and sinks. To implement this, they apply location heuristics to the information contained in the DNS.

<u>Abstract topology computation</u> -- Link-state routing protocols proceed by flooding the network with link-state information, and then computing a topology map from which routing decisions are then derived. As typically implemented, these schemes compute a separate routing decision at each hop, which implies that the decisions must be consistent to avoid routing loops. One exception to

this approach is Nimrod, a proposal for routing first conceived by Chiappa[CCS92], and developed at BBN. Nimrod splits the function of gathering routing information and constructing the routing map from the function of selecting the route. It permits maps to be constructed at various levels of abstraction. Our work is in this same spirit, but addresses the concept of a multidimensional routing space, and the objective of locating preferred service points within the network - points that can serve as routing way-points, meta-net transform points, or nearby sources of support services.

Meta-net abstractions -- Our previous work on Application Layer Framing provides one of the architectural principles for the meta-net. The "Touring Machine" project at Bellcore provided an alternative view of how to compose disjoint base technologies into a single managed architecture. Some early protocol research attempted to bridge different lower layer protocols into a single applications semantics, such as the NIFTP design from England in the early '80s.

SORCS -- Our previous work on the Information Mesh, especially in terms of its typing model based on roles provides one basic element of this work. The ANSA project[VDL93] has put some thought into the problem of birth of new objects, although they have not thought in terms of automated catalysis. In terms of catalysis, a Java applet is capable at present of generating a new web page which has a new Java applet in it, and thus, slightly indirectly of spawning a new active entity. The newly created applet inherits its privileges and restrictions as realized by the local security manager directly from the spawning parent. There are several candidate component definition languages, IDL which is part of CORBA from the OMG[OMG95], the DCOM/OLE model from Microsoft[Bro95, Cha96], and the JavaBeans component language from JavaSoft[Jav96]. There is significant work on index online indexing, much of which comes from the librarty community. There are also some notable services available from the computer science community some in the form of products such as AltaVista from Digital, Yahoo from Netscape, WAIS from Wais Inc, etc. These systems have an extremely simplistic model of self-indexing in that the whole object is presented (in Yahoo with some categories) and their indexers grovel over the whole text generating potential key words. There are also some collaborative efforts such as Harvest[Bow95] and the enterprise wide Service Location work of the IETF[VGP96]. Paul Francis of NTT Labs has proposed an interesting navigation system in the Ingrid prototype system.[Fra95]. This system is designed specifically to allow for navigating among similar documents, with a model of similarity based on key words chosen by the author. It is not our intention to innovate with respect to indexing services, but to take advantage of other work in order to address the problems of self-indexing.

Citations

Note: This section contains references only to specific works mentioned above and elsewhere in the proposal. See also the attached papers on our own related work. The list is not intended to be a complete list of published related work.

[Bow95] C. Bowman et al., Harvest: A Scalable, Customizable Discovery and Access System, TR CU-CS-732-94, Revised version, March 1995.

[Bro95] K. Brockschmidt, Inside OLE, 2nd ed., Microsoft Press, Redmond, WA 1995.

[Cha96] D. Chappell, ActiveX and OLE, Microsoft Press, Redmond, WA, 1996.

[CCS92] I. Castineyra, J. Chiappa, M. Steenstrup, *The Nimrod Routing Architecture*, RFC 1992, August, 1996.

[FJ93] S. Floyd and V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, 1(4), August 1993, pp. 397-413.

[FJ95] S. Floyd and V. Jacobson, Link Sharing and Resource Management Models for Packet Networks, Transactions on Networking, 3(4), August 1995, pp. 365-386.

P. Francis, Ingrid: A Self-Configuring Information Navigation Infrastructure, Proc. Fourth WWW Conference, Boston, MA 1995.

[GS95] J. Guyton and M. Schwartz, Locating Nearby Copies of Replicated Internet Servers, TR CU-CS-762-95, University of Colorado, Boulder.

[Jav96] JavaSoft, JavaBeans 1.0, Sun Microsystems, December, 1996.

[LR96] S. Lamm and D. Reed. Real-Time Geographic Visualization of World Wide Web Traffic. Fifth International WWW Conference, Paris, France, May 1996.

[OMG95] Object Management Group, The Common Object Request Broker: Architecture and Specification, Rev. 2.0, July, 1995

[VDL93] R. J. van der Linden, An Overview of ANSA, AR.000.00, Architecture Projects Management, Ltd., Cambridge, GB, May, 1993.

[VGP97] J. Veizades, E. Guttman, C. Perkins, Service Location Protocol, draft-ietf-srvloca-protocol-15.txt, January, 1997.