The Next Generation Internet: *Unsafe at Any Speed?*

Kenneth P. Birman
Dept. of Computer Science, Cornell University [1]

Abstract

Will the Next Generation Internet provide an appropriate infrastructure for critical applications, such as are emerging in such settings as health care, electric power grid control, air traffic control, banking, and military command and control? This paper suggests that current trends are unlikely to yield the required platform. Drawing lessons from successful critical networking projects of the past, we propose an alternative based on *Virtual Overlay Networks* (VONs). Support for VONs would not require radical departure from existing router capabilities, because a limited mechanism of the type we propose is already available. Given a simple Overlay Network (ON) capability, we show that tools for building more sophisticated VONs are already widely available.

I. Introduction

The premise of this paper is that an emerging generation of critical uses of the Internet requires functionality that the "Next Generation Internet" (NGI) will probably lack. The paper develops this assertion by studying representative examples of critical applications expected to migrate to the NGI in the near future. We find that in the recent past, these applications often relied upon network isolation achieved by using physically dedicated network infrastructures. This isolation permitted developers to offer safety arguments without concern about adverse interactions with concurrently active but unrelated applications. When migrating critical applications to shared network infrastructures, however, the isolation is often lost, exposing the application to potentially serious interference problems and compromising safety. Many critical users of the NGI will therefore need a way to isolate their applications from others using the network.

In a shared communications environment, isolation involves providing some form of guarantee to the application concerning freedom from undesired interference. The specific requirements of different applications, however, lead to multiple and rather different forms of isolation. For example, some critical applications will require security from intrusion, a property offered by existing technologies for "virtual private networks" (VPNs). If this is all they need, such applications could easily migrate to the NGI. But when we focus on reliability, which is the main topic of this paper, the need is for "virtually private" bandwidth and latency, and for network infrastructures capable of tolerating failures. Here, there is nothing in the pipeline that seems to match the need.

[1] ken@cs.cornell.edu. This work was supported in part under DARPA/ONR grant N00014-96-1014, DARPA/RADC grant F30602-99-1-0532, and NSF grant EIA 97-03470. The views, opinions and findings expressed herein are solely those of the author, and do not reflect official positions of the funding agencies.

Accordingly, we formulate a proposal calling for a new type of isolation within shared networks. It involves a primitive capability that we refer to as an Overlay Network (ON), and a means for layering software over an ON to refine its properties, yielding what we call a Virtual Overlay Network (VON).

The problem of supporting ONs and VONs is strongly evocative of what is called the Quality of Service (QoS) problem, although QoS problems are typically expressed in terms of one-to-one, one-to-many, or many-to-many communication streams (this is because the commercial driving force behind most QoS proposals arises from Internet telephony, media transmission, and group collaborative work applications). We therefore ask whether the most widely discussed operations for supporting QoS in the Internet would be adequate to support ONs and VONs. Although researchers in the QoS community have never looked at ONs and VONs, one might wish for a serendipitous outcome whereby some QoS proposal could be generalized to deal with this new need. Unfortunately, our findings are negative. Attempts to use QoS mechanisms to support ONs and VONs incur non-scalable costs.

The problem of supporting ONs and VONs, however, seems to be quite tractable. In the last part of the paper we point to existing mechanisms available in most commercial routers that could, if generalized somewhat, provide highly efficient support for basic ON functionality. We also identify families of protocols for network management and group communication that could be used to transform these ONs into any of a variety of VONs. Doing so would involve undertaking research on a number of technical issues, but nothing about the problem seems inherently hard.

These considerations leave us with a delicate political challenge. The government has made it a high priority that the NGI provide high quality support for critical applications, and is perceived by the public as investing to achieve this end. Our perspective, however, has identified a need that will apparently not be met by existing trends. Moreover, development and deployment of a cost-effective, functional ON and VON capability would likely require a substantial research effort and a substantial commercialization effort, well beyond what any small research team could undertake on its own. One is tempted to speculate that unless the political picture shifts to encourage this type of work, the NGI will fail to meet the needs of an important category of anticipated users, setting the stage for potentially catastrophic future technology failures as a new generation of critical but unsafe applications are deployed.

There is ample precedent for such concerns. In 1994, the FAA acknowledged the failure of its Advanced Automation System [Cristian 1996], an attempt to update the air traffic control system using modern network technology. At the time of this writing, air traffic control in the United States remains gravely challenged. The displacement of critical applications onto the NGI has irreversible momentum, and until the fundamental technical problems are solved in a satisfying manner, we face a period during which large sectors of society and the economy will be forced to rely on applications layered over an inadequate infrastructure, subjecting them to erratic performance and availability, weak security, or other flaws.

II. The Next Generation Internet

The term NGI refers to the future Internet, resulting from technology upgrades that will increase typical performance levels by factors of 10 to 100, widespread deployment of broad-band technologies that will enable typical users to tap into this improved bandwidth, and steady enlargement of the user community [Clark 1999]. The NGI has become a public policy concern in part because the Internet is expected to play a critical role in future economic growth, but also because of reports of a "crisis" in software, notably with respect to the security and reliability of network software [Gibbs 1994]. Studies such as the Presidential Commission on Critical Infrastructure Protection [PCCIP 1997] and the National Academy of Sciences study on Trust in Cyberspace [Schneider 1998] cite grave concerns about the implications of the wave of life, safety, and mission-critical computing applications being displaced onto either the Internet per-se, or dedicated *intranet*s built using commercial, off the shelf (COTS) components, such as computers, routers, network technology and software. The author participated in an earlier study of Survivability of the Information Infrastructure, commissioned by DARPA as part of its Information Systems Advisory Taskforce (ISAT) studies in 1995.

These studies reached similar conclusions. They confirm that COTS networking technologies make it feasible to deploy some very complex applications on networks, and find that the improving performance of the Internet has removed a major barrier to doing so. They make the observation that the same technologies used to control the public Internet are also used in dedicated intranet settings, hence the two share the same benefits and the same flaws. They identify profound cost pressures on enterprises in medical, banking, power distribution, air traffic control and other sectors, and note that these pressures are compelling a move to networked software architectures for applications of all kinds, including the most sensitive, critical ones. And they express anxiety that the infrastructure is not ready for such uses. The network itself remains prone to reliability, security[2] and performance problems, and we lack a widely accepted commercial technology base for supporting mission-critical application development over this base.

Underlying the trends are two basic observations:

1. Market pressures compel the use of commercially popular technologies. Over time, essentially all forms of proprietary technology are displaced by the most successful commercial products.
2. Many transitions of critical technologies to network platforms have been successful, although there are exceptions. For example, there are huge numbers of electronic banking and stock brokerage systems, and this author played a direct role in

[2] For example, Freedman and Mann report on a massive Internet break-in that ultimately penetrated thousands of computers [Friedman and Mann 1997]. The perpetrator was found to be a somewhat retarded adolescent, working from his room on an old computer, with endless patience and no real understanding of the cracker tools he downloaded from the network. Yet he broke into a vast array of computers, including the ones used by the Bureau of Land Management (BLM) to control floodgates and dams throughout the West. With a few keystrokes, this hacker could have unleashed massive floods, killing hundreds and destroying billions of dollars worth of housing and industry in that region. The BLM system exemplifies the type of critical computing application of interest to us.

developing networked, fault-tolerant systems for use in the New York Stock Exchange (to support the overhead displays), the Swiss Exchange [Piantoni and Stancescu 1997] and the French Air Traffic Control console clustering technology underlying a system called PHIDIAS [Birman 1999]. There have also been some striking failures, such as the collapse of the FAA's project, but there are more stories of the former sort than of the latter sort. Moreover, the economic benefits associated with the successes have been dramatic, creating further pressure in favor of migration.

One can characterize the broad picture as embodying elements of a "bet" that the Internet will evolve in ways that would overcome fragility and insecurity both of the network infrastructure, and the application-development tools and methodologies used by most developers. Anticipating that this trend will yield a network suitable for mission-critical networked computing, large numbers of mission-critical applications are migrating to the network. But will the network, in fact, develop in the desired manner?

Before trying to answer this question, we make an additional observation. It is a mistake to understand the NGI as if it were a single technical project, with a specific technical goal owned by a development team that will deliver a solution and deploy it on some date in the future. On the contrary, the NGI is really a catch-all term for the evolutionary path being followed by the current Internet, as new generations of faster routers and switches are rolled out and become widely deployed, gigabit fiber links become more common, and the size of the network as a whole continues to increase. A substantial community is actively researching and offering products for this future network, and while there are organizations to promote standardization and coherency of vision, such as the Internet Engineering Task Force (IETF) and the Internet-2 consortium, none has any real control over this larger process.

One can anticipate that the NGI will see rollouts of new generations of infrastructure components and connection options, some of which are already available but not universally deployed. Candidates include:

- *IPSec:* A recently introduced security architecture for the infrastructure, with protocols to secure the Domain Naming Service (DNS) and routing protocols.
- *DNSec:* A secured version of the Domain Naming Service (DNS), extended into a general-purpose directory architecture.
- *IPv6:* An extension of the IP packet formats that provides room for security headers, payload signatures or encryption, and support for long IP addresses. IPv6 is available today, but used only in a small number of experimental settings.
- *QoS:* A number of proposals would extend the Internet to provide "quality of service", such as guarantees of bandwidth and latency for voice connections.

Were the user community not a moving target, these proposals could be evaluated against a quantifiable demand. But even as these and other new protocols or services are deployed, the profile of use of the NGI is expected to evolve dramatically. For example, whereas access to the current generation of the Internet tends to be from immobile machines over dedicated lines and low-speed modems, access to the NGI will be through a mixture of high-bandwidth connections and much lower speed wireless mobile connections. This change in access pattern is likely to result in major changes in the application profile and communication patterns seen on the network.

Under such conditions, vendors are loath to make expensive bets and tend to operate conservatively, in response to well-defined market opportunities with immediate payoffs. Accordingly, our premise is that the NGI will look much like the current Internet, but operating at higher speeds and with infrastructure security mechanisms to protect routing and address resolution. We assume that some sort of connection-oriented QoS mechanism adequate to support voice communication over Internet connections will also emerge. But it seems unlikely that the NGI will incorporate any sort of systematic response to those faced with developing critical networked applications meeting stringent application-level reliability and security goals.

III. Mission Critical Computing on Networks: Two Case Studies

To better understand the nature of mission critical computing on networks, we now focus on two domains widely cited as exemplifying the trends and embodying past successes: medical computing and avionics. We do not pretend that these are completely representative, but brevity prevents us from exploring a larger set of critical networking systems, and these two applications share several attributes seen in many critical settings.

II.1 Intensive Care Computing

The development of computing systems and networks for critical-care settings (Figure 1) began soon after networking technologies first became widely prevalent at the beginning of the 1980's. In contrast to what are called *clinical database systems*, medical *critical care systems* support devices that practice medicine. For example, these systems support telemetry in cardiac care units: If the nursing station is unable to monitor a patient's heart rhythms correctly, that patient's life is endangered. The same can be said for systems that set off alarms when the patient's vital signs fall outside of acceptable thresholds. Over time, more and more such applications have been developed, to the point that today, there are even devices for infusing drugs that can be controlled over a network. In fact, this capability is rarely exploited, but the feasibility of remotely controlling the dosage of various drugs over a local network or even over some form of dedicated network link from the health care organization to the home is well established.

To market a critical care product, a vendor must obtain FDA certification before the product is offered for sale. This is a process requiring stringent verification of the safety of every aspect of the technology, including the software used within it. When devices are connected to networks, the network itself plays a critical role, hence the FDA worked with the IEEE to develop a standard (IEEE-1073) for medical information buses. Given a communication device compliant with this standard, the vendor who attaches a computing system or display to the bus is justified in making strong assumptions about bandwidth, loss and error rates, and the real-time characteristics of the network.

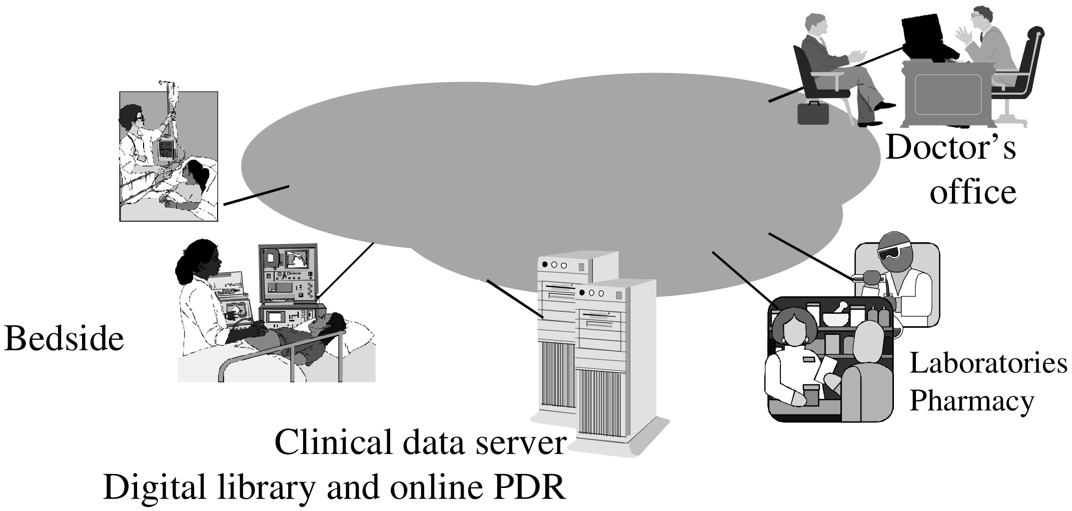

Figure 1: Although one thinks of a hospital network as being a single system, such a network actually superimposes one or more critical care networks on a clinical database network.

Thus, the HP Careview system, an intensive care unit (ICU) monitoring application developed by Hewlett Packard Corporation, incorporates a timing mechanism into its display devices. When a Careview display device senses that telemetry is not being received within the parameters of the communication bus and the data collection server, it triggers an alarm, warning the user that the telemetry displayed on the screen may no longer be accurate. HP's engineers took advantage of the IEEE-1073 specification to calibrate the associated timers so as to avoid false alarms while still ensuring that real outages will be promptly reported.

This approach to quality assurance does not extend to the clinical database (CDB) side of the hospital system. A CDB is a (legally binding) record of actions taken by the hospital accompanied by patient vital signs or laboratory results on which those actions were based, and each entry to such a database must first be reviewed and approved by a human care provider. Whereas the ICU is concerned with instantaneous display of data as it is gathered, the CDB is understood to lag reality, perhaps significantly. Until recently, a physician would not have been justified in basing a treatment decision purely upon the clinical record (without seeing the patient) because of the danger that after the CDB was last updated, something important changed. But this same reasoning makes the CDB less critical in the eyes both of the hospital and the FDA. Thus, an ICU monitoring system must be entirely validated for safety before it can be used, but a CDB would normally be constructed from off the shelf technologies and database software, using conventional application development tools and running on standard PC's or workstations.

Recent trends, however, are blurring the distinction just drawn between the ICU and CDB systems. Increasingly, data from the ICU system is imported directly into the CDB, enabling the care provider to base decisions on the clinical profile of the patient without visiting the patient at his or her bedside. This makes particularly good sense as we look to the future, because there is a growing trend to use networks in support of home care for patients who might traditionally have been admitted to the hospital, but who can actually be treated from their homes provided that a small amount of monitoring capability is available, and that devices such as insulin pumps are remotely programmable. Indeed, the entire ICU system is increasingly being migrated onto the same platforms and networks that host the CDB system. For example, customers insisted that HP migrate its product onto standard workstations running standard versions of PC operating systems or Unix, and supporting standard tools such as web access to the medical library, billing systems, etc.

This new configuration brought some surprises. HP's Careview developers describe denial of service problems that emerged as an issue in the shared configuration of their application. The network connecting the telemetry system to the displays was no longer dedicated, and during periods of heavy load the telemetry data traffic was sometimes starved of bandwidth. Yet the current generation of Internet protocols lacks any means whereby Careview could specify its requirements or be sure that they will be enforced [HP 1997]. And the problem is hardly unique to HP. EMTEK, a company manufacturing clinical database products, found that its users were running the system on networks connected to the public Internet – against their advice, but for good reasons, such as to support access to drug company web sites. Even with industry-standard security in place, such as firewalls, such a system is potentially subject to attack, and in some configurations intruders succeeded in connecting to the EMTEK servers over the Internet. At least one EMTEK database administrator found it amusing to connect to the server sites of his colleagues, leaving messages warning them to improve their security precautions [EMTEK 1997].

These examples illustrate a broader phenomenon. The "old generation" of medical systems for critical tasks gained an important benefit from the strict isolation of the IEEE-1073 standard. In effect, the bus made it possible to reason about the safety of an application without considering other applications running in the same setting. In the new world of the NGI, however, the networks have been combined in a way that violates this implicit assumption of isolation. Critical applications share components with non-critical ones, and the dedicated platforms used in the earlier systems have been displaced by standard PC's running Windows, supporting Web access, and employing the same security mechanisms used in offices. It is no longer clear where a safety certification process should stop. A critical application might now be influenced by some other non-critical one sharing the same platform or network.

Moreover, if trends in favor of *Community Health Information Networks* (CHINs) continue, applications will increasingly span the public Internet. One can already anticipate that it will be common to monitor the health of patients over the network and to control the administration of at least some kinds of medications and therapies remotely. As any user of the web will confirm, such configurations will confront denial of service problems far more serious than anything HP Careview faces in the hospital intranet settings.

Wishful thinking would have us imagine the following capability. Suppose that the NGI offered a ubiquitous infrastructure *capable of mimicking the older split between critical and non-critical functions.* The hospital might then configure the NGI to emulate a configuration within which each of the various critical care units in the hospital would have its own dedicated network, perhaps with properties matching the IEEE-1073 standard, perhaps updated to take advantage of faster hardware, wireless connectivity, or other features of modern network devices. Additional dedicated networks could be configured to reach out into the community, for purposes of linking together community health providing organizations (such as hospitals and laboratories and specialized treatment centers), or to connect into the homes of patients receiving outpatient therapy. Still further networks would be configured for dedicated use by the CDB system; these would be isolated to prevent any sort of intrusion from the outside.

Our vision, then, is one of a number of virtual networks having a variety of real-time, bandwidth, jitter, or other performance properties, superimposed on a shared infrastructure (Figure 2), and somehow isolated so that the abstraction would be robust against some class of "threats[3]." Ideally, given a threat profile, the network could be configured to protect against it in the cheapest manner appropriate for responding to that category of events. An idealized architecture, then, would be one in which each application could configure a virtualized network isolated from external phenomena and matched to its needs, and then point to the properties of the network in certifying the safety of the solution.

Our medical scenarios yield examples of several kinds of isolation. The CDB system might need to ensure that privacy of patient records is adequately protected – isolation from unauthorized access. ICU systems need to guarantee that telemetry data arrives in a timely and secure way – a bandwidth isolation requirement. The system will need to eliminate risk of configuration errors whereby the telemetry for patient A might be displayed as if it was data for patient B (believe it or not, at the time of this writing, such misconfigurations are possible, although obviously only as a result of gross error on the part of the user). A remotely controlled insulin drug pump might need the guarantee that alarms will be correctly reported in the physician's office and that any reprogramming of insulin dosage ordered by the office will correctly and promptly be communicated to the pump, with failures in the overall process guaranteed to trigger alarms at both ends.

[3] Here, we use the term "threat" loosely. Whereas the security community uses threat to refer to a person seeking to disrupt a system [Schneider 1999], we could imagine anything from an overt attack by a virus or an intruder to an accidental disruption resulting from a configuration error or a crash.

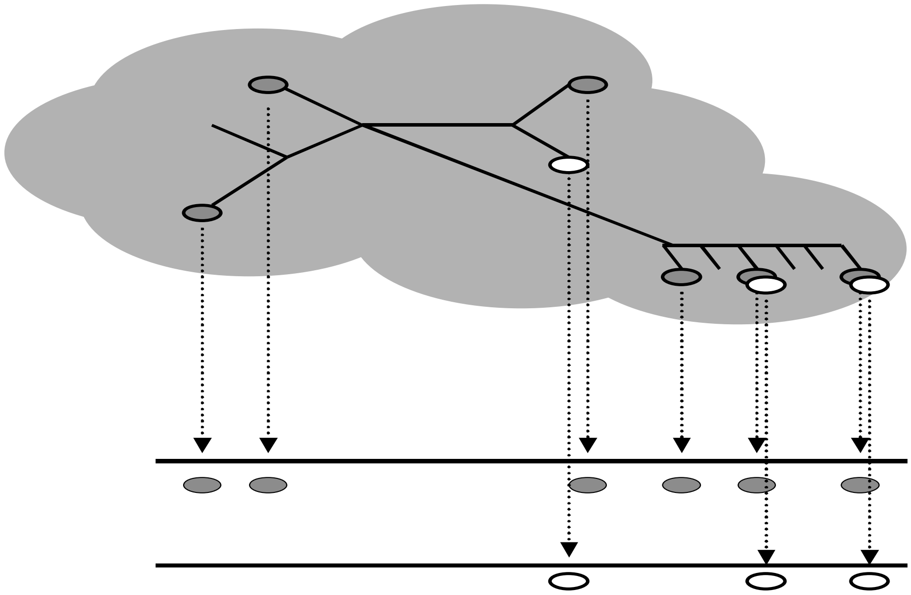

Figure 2: Two virtual networks overlayed upon one physical one. The overlay hides the structure of the physical network and may offer guarantees, such as isolation from interference or minimum bandwidth. The network shown would be incapable of offering fault-tolerance to router or link failures, because the underlying physical infrastructure lacks redundancy.

Each of these examples involves some form of isolation at the most basic level: ensuring that only legitimate users can connect to the network, that users can be authenticated, that some minimal level of bandwidth is available, and so forth. The latter illustrate functionality that might be layered over such a basic capability. As we move forward in this paper, we'll want to evaluate the various proposals for evolving the Internet with respect to these sorts of goals. To the degree that our goals are unmet by existing options, we'll also want to ask whether other, unexplored, alternatives could offer a better path forward.

II.2 Integrated Modular Avionics: The Boeing 777 Safebus

To support our argument that the needs just identified are broad ones, we briefly survey a second domain within which critical uses of computers have been well known and understood for many decades: avionics. This term, as used here, covers the electronics and software packages that operate onboard an airplane to control everything from the wing flaps to the air conditioning and computer games. Avionics systems thus span a range from life- and safety-critical to mundane. The FAA requires that critical avionics components be subjected to the most precisely defined levels of safety verification, while verification for non-critical components is limited to a demonstration that these cannot in any way interfere with the critical ones.

Developments in the avionics field parallel those in biomedical engineering. Until recently the FAA adopted a holistic approach to safety verification: the manufacturer presented the entire aircraft to the FAA with a safety certification proof, and the job of the FAA was to function as an informed skeptic, challenging the safety proof until the FAA had satisfied itself as to the methodology used and the quality of components. Avionics networks tended to be dedicated to the task and redundant: the flap control network had its own computer and its own wiring, distinct from the network used to control the engine or the one used to control the cabin ventilation system.

Cost pressures and the desire to standardize components, however, have led to a dramatic evolution in the industry, namely the emergence of *integrated modular avionics* (IMA) systems. Under this approach, the avionics system is described in two dimensions. One dimension is concerned with the various subsystems, such as the ones just mentioned. The second concerns modularity: the manner whereby components can be composed to build a larger system, and the extent to which they can share the same network or processors. Using such an approach, the team that developed the Boeing 777 proposed a shared network architecture called the *Safebus* together with a fault-tolerant software architecture used to develop critical technology components [Safebus 1994]. The Safebus provides a very strong form of isolation and predictability and the component developers draw upon these guarantees to simplify the verification argument presented to the FAA. For example, the flap controller software needs to be isolated from interactions with other subsystems, hence the Safebus runs it in time slices during which only state declared as persistent is preserved. All other aspects of the processor state are reinitialized each time the controller is executed. The processor cache and communications bus are initialized to a known, idle states, so as to minimize variance in cache hit rates and bus contention. The effect of this is to eliminate any risk of interaction between, say, the flap controller and the engine controls (or even a non-critical application, such as a video game), unless the Safebus itself is incorrect. Boeing can then certify the system by convincing the FAA that the Safebus correctly isolates the flap controller, that the controller itself is correct in isolation, and that its interactions with other subsystems (through well-defined interfaces) are appropriate and safe.

With IMA, Boeing and other aircraft manufacturers hope to achieve a situation in which new airplanes are cheaper to certify because components reused from one plane in a new one will need much less stringent revalidation. When a component is retired in favor of an upgraded version, the hope is that only the changed component and other subsystems with which it has direct interactions will need revalidation. Moreover, by permitting components to share some forms of hardware (and by demonstrating separately that the solution is fault-tolerant), cost savings can be achieved.

Meanwhile, other trends are introducing significant new challenges into the overall picture. With the dramatic failure of the FAA's Advanced Automation project, the ground air traffic control systems face extreme stress and overload. Current thinking is to reduce this load by taking controllers out of the loop for some types of long-distance flight, replacing them with a more autonomous style of decision making in which the pilot and navigator of each aircraft are authorized to direct their aircraft with ground intervention only in the event of exceptional circumstances. They make these decisions with automated support from ground-based database and planning systems. In effect, "free flight" (also called "4D navigation") injects the ground-based air traffic control database system into the navigation platform of the plane, raising a completely new kind of safety certification issue. Since the ground system will run on COTS-based networks and databases, one must envision a world within which the avionics system on the plane is almost an extension of the network on the ground, and where safety issues arise in a continuum reaching from plane to plane through the ground.

Is it reasonable to assert that the Boeing 777 Safebus, coupled to an air-traffic control system on the ground, constitutes an instance of an NGI problem? For the purposes of this paper we will argue that it does. Obviously, the applications running within the plane on the Safebus and the ones that comprise the air traffic control system are specialized, but both the Safebus and the air traffic control system are forms of networks. The attached system components would be hardened to provide the isolation properties just described, but having done so they could support rather conventional software – one could easily imagine running a Windows CE operating system and application in a Safebus partition. Indeed, if we step back and consider the large-scale architecture, it seems implausible that the overall network could be controlled other than by some form of NGI-based technology. The problem is that to avoid using NGI technologies, the FAA would need to build its own proprietary networking technology, and doing this on a large scale would be prohibitively costly.

In the past, there was a perception that cost was no object for projects as critical as air traffic control. Even today, this remains true on a small scale – the computers comprising the Safebus, for example, may well run a special purpose operating system or employ specialized pieces of hardware conforming to well-accepted industry standards. But as we scale such a system to larger and larger challenges –the ground-based air traffic control system contains thousands of computers and hundreds of database servers – it makes less and less sense to imagine that a specialized solution built by some single vendor could possibly be employed. Whether or not the NGI is designed with flight management applications in mind, this class of systems is another likely application for the NGI. Small wonder that the PCCIP and NAS studies expressed such alarm!

As we abstract from the broader FAA picture, it begins to look more and more like the medical one, or other emerging critical applications, such as for control of the restructured electric power grid, for environmental and weather monitoring, and emergency communications. The same displacement from specialized solutions to COTS-based solutions is occurring in all of these settings, and the same sorts of challenges are created by the trend. These may not be the majority users of the NGI, but these kinds of applications will surely migrate to the NGI technology base, and the degree to which they find a good match may have an important impact on the quality of life in this country in coming decades.

II.3 Characteristics of Critical Networking Successes

Our review of critical applications for the NGI, in conjunction with the prior history of successes in each field, reveals fundamental requirements for the NGI itself. To give some sense of the diversity of possible goals, we enumerate just the most obvious ones that arise from the examples reviewed above:

- Support for certain communication primitives, such as the IP suite, including IP multicast. (One could imagine other sets of primitives).
- The capability of isolating applications from one-another, so that a critical application (or sub-application) can be considered and validated with limited attention to other applications. Applications will need the ability to interact through well-defined interfaces, but the NGI should offer the building blocks needed to preclude undesired interactions or interference.
- The ability to guarantee certain quality of service properties. These might include some of the following (but in general, only a subset of these would be desired):
  - ❑ Minimum bandwidth
  - ❑ Maximum rate of packet loss
  - ❑ Maximum jitter (instability of delivery rate)
  - ❑ Minimum and maximum delay from sending to delivery of a message
  - ❑ Infrastructure robustness against mundane events, such as router failure, the failure of a line, congestion, and so forth.
  - ❑ Infrastructure robustness against attack (for example, the guarantee that intruders cannot disrupt routing)
- Management functionality
  - ❑ Infrastructure support for application and resource location (e.g. a DNS capability)
  - ❑ Infrastructure support for connection establishment and routing, in point to point and multicast cases.
  - ❑ Infrastructure management capabilities
- Methodology support:
  - ❑ The ability to provide a rigorous (mathematical) characterization of properties guaranteed by the infrastructure
  - ❑ A methodology for certifying components for use in the infrastructure.
- Over this basic infrastructure it should be possible to construct stronger application-level properties, such as:
  - ❑ Application-level security against attack (for example, data encryption, authentication, and a key infrastructure)
  - ❑ Automated system management mechanisms
  - ❑ Application-level fault-tolerance features
  - ❑ Data replication, for high performance and consistent handling of events
  - ❑ Support for management of persistent data and database support, which often require network-level standards for transactional operations
  - ❑ The ability to undertake some form of mathematically rigorous verification of proposed solutions.

Obviously, any single application would need only a subset of these properties, but each one of them might be critical in the kinds of critical applications we have reviewed up to the present. One could also imagine even stronger properties, such as the ability of the ON to resist attack by compromised participating endpoints, but for the purposes of this paper we will limit ourselves to a more benign concern, namely that of providing ON and VON functionality in settings where the "problems" are mostly external.

In the remaining parts of this paper, our "agenda" will be as follows. First, we introduce some terminology that will be used to characterize hypothetical NGI support for these sorts of guarantees and properties. Then, we measure the current proposals for the NGI against these needs. Finally, we offer a proposal of our own. The intention of this paper is not to "prove" that the NGI, on its current path, will be inadequate, nor is it really to propose a different technical approach: the NGI is evolving in response to perceived markets, and its capabilities will surely be useful for many purposes even if it is not able to support critical applications. Rather, the hope is that by posing the challenge as it does, this paper will invite others to offer concrete thoughts on how their proposals for NGI features respond to the real needs of critical applications.

IV. Overlay Networks and Virtual Overlay Networks

The term *Overlay Network* is sometimes used to describe a configuration within which a base network is used to support some second network, which is "layered" upon the underlying infrastructure. For our purposes in this paper, this loose definition of overlay network matches well with the requirements identified in the previous section. For example, suppose that Gotham Hospital has decided to build a CHIN. The hospital might go about this by purchasing some number of overlay networks, each isolated from the others and from the public network, and each having properties and bandwidth matched to the use. Presumably, Gotham Hospital will pay more for stronger guarantees, so it will be motivated to configure each ON to the needs of the corresponding application.

Each ON would be characterized by:

- An identifier, with which traffic within that ON can be tagged. Routers will need this to distinguish between traffic on different ONs.
- Some set of access points. One can imagine the ON as a sort of virtual Ethernet, in the sense that it offers service at multiple locations, and the resulting applications share what is logically a single infrastructure. In adopting this design point, we draw on the success of the biomedical engineering community and avionics community in building critical applications over multipoint communications architectures.
- Some set of guaranteed properties. Notice that these properties will usually relate to the aggregated traffic on the ON, not to point-to-point paths within the ON. For example, if an ON is configured to mimic a dedicated 10MBit Ethernet, we know that except for packet loss due to undetected collisions, the network should be largely reliable when the presented load is less than 10Mbits per second. But this load may originate at a single sender, or it may originate at multiple senders. HP Careview is designed so that when it runs on a dedicated network, it will not exceed the capabilities of that network. Our goal is that by purchasing the appropriate ON, Gotham Hospital can run HP Careview and even extend its monitoring ability right into the home, without changing the Careview application (except to the extent that it includes latency-dependent code).

As we move forward in this paper, an argument will emerge that the ON need not offer every possible application-level property. Given an underlying ON with a sufficient set of raw or base properties, it will often be possible to build layers of software over the base ON that introduce new abstract properties, such as the application-level security properties enumerated earlier. We will call such a network a *Virtual Overlay Network* because it extends the base properties of an ON with new properties that appear to hold for the underlying network, but are actually implemented on behalf of the user by means of some sort of protocol or algorithm. In general, our goal should be to identify the weakest possible ON that would still support the widest possible set of VONs, although this paper will not actually solve that problem. Instead, we simply observe that once an ON provides some basic guarantees of minimum bandwidth and packet loss, and secures its own infrastructure against failure and attack, it should be possible to build almost any desired property over this, provided only that the basic speed of the ON is adequate to support the desired speed of the VON even with this intermediary software in use.

For example, suppose that the ON guarantees 99% reliability (with respect to packet loss), but that the application requires 99.99% reliability. If the packet losses are independent, a VON providing the latter property could be implemented by sending each packet twice. Of course, a real ON might not be able to guarantee absolute freedom from correlated packet loss, hence the conditions under which the VON can provide the desired reliability may not be as simple as this implementation would suggest: perhaps, the VON only guarantees high reliability at lower speeds than the ON, or requires a redundant underlying ON infrastructure so that each packet can be sent on two independent paths. Our goal and model, then, is of an ON rich enough to support such solutions, and a collection of VON implementations layered over the basic ON options, paying (for example, by redundant communication and perhaps even by purchasing redundant communication paths) to achieve a stronger quality of service guarantee.

At the risk of digressing momentarily, notice that even a task as basic as formalizing "99% reliability" is surprisingly hard. Suppose that one is given a network that works properly for the 990,000 packets sent, then drops 10,000 packets in a row. Was this network 100% reliable, and then 0% reliable, or is this a 99% reliable network? Clearly, reliability of this sort has a time-scale. Pursuing the idea, one might say that during any 1-second period, no more than 1% of packets are dropped. But now suppose that a 1-second period occurs during which only 50 packets are sent. Must this network deliver all 50 to satisfy the 99% goal, or can it drop 1 packet, which seems more like a 98% reliability property. If we somehow settle on a definition, we encounter a new difficult question. Given a pair of reliability goals, under what conditions can one be strengthened to achieve the other, using redundant transmission or other means? Such questions are deep, and much work will need to be done before even simple statements can be reduced to implementable mathematical formalisms.

V. The Next Generation Internet

We now return to the NGI itself, in order to better understand how it will differ from the current Internet in the dimensions relevant to our discussion.

IV.1 The NGI Vision

The NGI is a work in progress, and to make matters worse, there is much debate about the best way to achieve even the widely accepted goals. Broadly, evolution of the NGI occurs in several ways.

- The dominant form of evolution reflects competition between the major router and switch vendors. The market – consisting of ISPs and telephony providers – expresses its preference by favoring the better product, and to the extent that a feature becomes widely used and widely successful, similar features soon appear in competing products. The cost of new features is a very important aspect of this competition.
- The second important driver is the end-user. Users pay for the Internet, and generate the traffic vendors carry. So the Internet evolves as ISPs respond to changing patterns of use and competitive pricing pressures to offer packages of services and pricing appealing to their customers.
- The third driver is the research community and IETF. This community is a steady source of ideas, but only some of them transition into widespread use, and any large-scale acceptance tends to be slow.

The remainder of this section speculates about the evolution of the NGI in light of what is known about the trends in these three respects. We focus on performance, security and quality of service.

IV.2 Achieving Higher Performance

It is certain that the NGI will be faster. Widespread deployment of broadband technologies and optical fiber is expected to yield a 10- to 100-fold improvement in the performance of the Internet. This benefit reflects the steady improvements of backbone segments of the network and the rollout of broadband technologies such as cable and DSL modems to the end-user.

IV.3 Improving Security

The NGI will also have much better security properties than does the current public Internet. ISPs have a strong motivation for improvement of the security and maintainability of the network infrastructure. At present, a clever hacker could severely disrupt the Internet, for example by flooding it with phony routing table updates or DNS updates. Accordingly, one can expect rapid deployment of the IP security architecture, IPsec, and the secured versions of routing and DNS protocols. As noted earlier, the DNS architecture itself is likely to evolve in conjunction with this rollout.

Application-level security is another matter. Earlier, we mentioned *virtual private networks*, which are well-matched with one aspect of the security requirements enunciated above. These are software abstractions overlaid on a shared network, in which communication between end-points belonging to the VPN is signed (for authentication), encrypted (for secrecy), or both. Given a key infrastructure, and these are increasingly standard, a VPN offers a way to implement one form of VON – but only one form, and only one at a time: a given machine can only belong to a single VPN. Indeed, a VPN is similar to a firewall, except that whereas a firewall acts only at the periphery (where packets are filtered), a VPN acts at the network interface of each attached computer.

In our own work, we have explored the extension of VPNs into a form of VON focused exclusively on security issues [Rodeh et. al. 1988]. Called a Dynamic Virtual Private Network (DVPN), our solution permits a single machine to belong to multiple VPNs, and provides fault-tolerance. We also provide protocols for rapidly changing security keys when the set of participating computers or applications changes. DVPNs are limited to the case where no services are shared between VONs (having a single file server or database that lives in multiple VONs becomes problematic if the VONs are nominally secured with respect to one-another, since the server can easily leak data from one security domain to another). Nonetheless, it seems reasonable to assert that by extending the VPN concept most security issues arising in ONs and VONs can be solved. This said, it also seems reasonable to conclude that absent changes in the security marketplace, DVPN solutions are not likely to emerge anytime soon.

The medical and avionics critical systems point to other types of security issues too, which require responses at levels other than within the infrastructure. A lengthy digression would be required to do justice to these here. Very briefly, the issue is that the application-level security problem of greatest interest to current security solution providers, beyond the firewall-like problems just discussed, concerns the security of simple financial transactions over the network – secured access to web sites, and secured electronic purchases. Such a model is soon seen to be inadequate when we consider the security needs of a hospital. In such a setting, one wants to permit unrestricted access to a patent's records from the bedside, but to restrict or at least create audit trails for access to the same records by the same individual when he or she is working in some other unit.

Suppose that renowned actress Marilyn Monsail is admitted to Gotham Hospital. When Dr. Kildare is at her bedside, he should have the information needed to make treatment decisions. But, if Dr. Kildare is not one of Marilyn's physicians, the same access might violate her privacy. Such issues become even more critical when we consider remote control of devices, such as insulin pumps, where the patient might be seriously hurt or killed by incorrect dosage levels: Dr. Kildare should be able to treat his own patients, but should be prevented from interfering (whether accidentally or intentionally) with the treatment of other patients.

Problems such as this demand a mixture of security mechanisms, such as key infrastructures and authentication, and *policy rules*. Our example illustrates what might be called a "role-based" security policy (ones that distinguish between the abilities of the same individual in different situations), and a form of location-aware policy. These are examples of problems not treated by the current generation of security solutions, hence it is unclear what forms of lower-level infrastructure support will be needed to solve them.

The National Academy of Sciences study cited earlier discusses such problems in some depth [Schneider 1998], and spells out a research agenda that responds to the need.

IV.4 Options for Supporting Isolation

Although expressed in terms of isolation, one can also understand an ON or a VON as offering forms of many-to-many quality of service. This creates an apparent match between trends in the Internet QoS domain and the needs of critical applications such as the ones we surveyed. Internet QoS research seeks to overcome the erratic performance of the current Internet, which respresents an obstacle to migration of telephony from the current dedicated infrastructure onto a packet-based one.

In reviewing this work, it is important to keep in mind that the prevailing QoS proposals were formulated without consideration for supporting ONs and VONs. Moreover, most QoS proposals focus on point-to-point communication links, and their extension to multipoint communication is an active topic for research. Nonetheless, if there is a way to support an ON or a VON using technology already on the NGI table, QoS proposals are the obvious place to look. Our challenge is thus to seek an unrecognized opportunity to reuse one of these mechanisms in support of a VON.

a) RSVP.

The RSVP proposal was introduced as a response to the QoS requirement for the NGI, and advanced to the RFC stage, only to falter when ISPs proved unwilling to accept the high costs of the solution [Zhang 1993]. Basically, the proposal operates by sending a special kind of resource reservation packet from the connection initiator to the remote end-point. At each router along the way, a resource reservation is attempted. If the request reaches the remote endpoint successfully, a confirmation packet locks down the reservation and the connection is guaranteed some service profile (bandwidth, latency, priority) at the routers on the route. Periodically, the reservation must be refreshed or it automatically lapses. An extension of the basic methodology permits reservations on behalf of multicast sessions.

There are two common criticisms of RSVP, which we cite because our own ON and VON proposals will have enough similarity to RSVP to elicit similar concerns. First, when using RSVP, routers must classify each incoming packet with respect to the source and destination in order to match packets to the corresponding reservation (the “flow” to which the packet belongs must be located). Resources must be allocated on a per-flow basis. Thus, as a network scales and the number of potential point-to-point connections rises, the load on the router can be expected to rise at least linearly in the number of endpoints, and perhaps even quadratically. These costs are seen as prohibitive, and while fast classification algorithms have been proposed, it seems likely that the classification problem is hard to solve on a large scale at gigabit speeds.

A second and more administrative issue concerns the correct way to bill the user for the cost of the call: if ISP A must route the call through more of its routers than ISP B, how should the billing be split between them? If ISP B has legal responsibility for the quality of service guarantees offered to the user, how can ISP B monitor the behavior of ISP A to protect itself in the event that the guarantee is violated? And suppose that one ISP or the other experiences an outright failure. How should RSVP behave when a major routing change occurs?

b) RED and RIO

Random Early Detection (RED) is a scheme for signaling congestion from routers to TCP endpoints by starting to drop packets before congestion actually occurs. The reasoning is that because TCP needs some time to back off, signaling congestion only after a problem has occurred is "too late". Instead, as a router first detects a possible load surge, it immediately begins to select packets at random and drop them, in this manner eliciting a back-off response from the TCP protocol. *Random Early Detection with In and Out of Profile Bits* (RIO) extends this idea to provide a form of quality of service for connections [Floyd and Jacobson 1993] [Clark and Fang 1997]. Users, when they establish connections, provide a profile of expected use. On this basis, resource availability is checked – this can be done using a scheme like RSVP, or through some form of centralized resource scheduling service. The RED policy is then applied with a bias: given a choice, the router sensing overload selectively targets packets that are either not associated with resource reservations at all, or that are out of profile. Notice that the classification problem occurs at the time a packet enters the network, not in the inner nodes. This eliminates the potentially quadratic costs mentioned earlier. On the other hand, resource reservation is still required, and if an inner router becomes oversubscribed, reservation requests must be denied. Finally, it should be stressed that the reservation policy is based on a notion of point-to-point connections. The significance of this will become clear shortly.

c) Diffserv

Differential service mechanisms generalize some aspects of the architecture just described to allow providers to allocate different levels of service to different users of the Internet. Broadly speaking, any traffic management or bandwidth control mechanism that treats different users differently – ranging from simple Weighted Fair Queueing to RSVP and per-session traffic scheduling – falls under this definition. However, in common Internet usage, the term is coming to mean any relatively simple, lightweight mechanism that does not depend entirely on per-flow resource reservation. In particular, the Diffserv community has been increasingly focused on a class of mechanisms that operate by classifying packets at the borders of the network on the basis of the user's usage profile, recent access patterns, and willingness to pay for higher quality of service. Higher priority is given to packets from users paying for better service and communicating lightly; lower priority to packets from users paying for best-effort performance or who have exceeded their predicted traffic profile. Routers weight their decisions so as to favor high priority packets over lower priority ones. Clark and others have shown that these sorts of policies can be administered in a manner that frees the routers within the network from any need to classify packets beyond the prioritization already performed at the edge of the network, with very good probability that the user's desires will be achieved [Clark 1995].

IV.5 The High Cost of ONs and VONs Using Proposed QoS Mechanisms

None of the options listed above are particularly good choices for implementing ONs: each yields an expensive solution. The fundamental problems arises from the need for an ON to emulate a dedicated network resource that may have many users. Imagine a medical monitoring system configured to use a 10Mbit ON accessible at 100 endpoints, for example. As we saw earlier, such an application would need to be designed so as to ensure that that overload will never occur, or so that any overloads which might occur (causing packet loss) are tolerable, but the problem is a familiar one, and in the absence of contention from unpredictable sources, one can solve this problem in any of a number of ways: by analysis of the communication load associated with the application, scheduling communication, etc. We now wish to move this application to run on an ON implemented using a QoS mechanism as the underlying building-block.

It is quite possible that the burst load associated with any particular pair of endpoints (a source sending to some destination) could reach the performance limit of the ON. Thus, in our example, the ON must guarantee an aggregated bandwidth of 10Mbits/second, but be capable of providing peak bandwidth to any pair of users that happen to generate the full load at a time when the ON is otherwise idle. Each pair of endpoints is a potential generator of 10Mbits/second. A connection-based resource reservation protocol might therefore be forced to reserve O(100^2*10Mbits), or 100Gbits, of capacity at a central router. Clearly, such a naïve implementation of resource reservation would be unsatisfactory.

One can now speculate about various options for improving our naïve solution. For example, we could dynamically allocate resources to endpoint-pairs, so that at any point in time, the aggregated allocation doesn't exceed 10Mbits/second, but with the allocation shifting around as needed. But shifting the allocation of bandwidth will take time, and requires a form of distributed coordination protocol that is hard to support with fault-tolerance guarantees. We rule out this solution as overly complex.

Similarly, one can imagine periodically multicasting the activity profile of the endpoints so that all endpoints have a reasonably current picture of the pattern of usage within the ON as a whole. But this solution suffers from an n^2 growth in overhead messages: as we scale the size of the ON up, the number of such messages will grow at least linearly in the number of endpoints, and each is a *1-n* multicast. Here, the multicast could be implemented using a simple scheme because reliability is probably not critical, but the background load looks intolerable. Moreover, the higher the quality of data needed, the more overhead we pay. With any lower quality of data, the quality of profile marking or prioritization may become unacceptably poor.

It would be speculative to continue along this line of analysis. But it seems likely that one could prove that solutions of this sort, based on point-to-point reservation models or quality of service mechanisms, are intrinsically costly and scale poorly. The NGI, then, is unlikely to represent a very friendly environment for applications seeking ON mechanisms. More sophisticated VON solutions are even less likely to succeed.

VI. Alternatives for ON and VON Support

The considerations just cited suggest that the emerging NGI is unlikely to yield a technology suitable for the kinds of critical uses reviewed previously. The NGI will certainly be faster and more secure, and will probably support a variety of telephony and media transmission options. Yet it will not be "safer", if the safety of critical applications is an important objective. However, such an outcome could be avoided. We now present an informal proposal for an alternative way of building VONs that seems to be well within the reach of current technology, and capable of responding to the need.

V.1 Using Router Partitioning to Implement ONs

Existing routers support a mechanism by which one ISP can supply networking connectivity to another, with guarantees of minimum throughput and priority. The technical feature with which this is accomplished is termed a "router partitioning" and involves tagging packets with a "flow identifier".

To use router partitioning, a provider configures the router to set aside some percentage of its resources (Figure 2), dedicating these to a particular flow. Each incoming packet is classified by means of its flow identifier and treated as if it resides entirely within a virtual router defined by the resource subset allocated on behalf of that flow. For example, if Global Domination Networks leases one-half the capacity of some router to Gotham Network Solutions, then each GNS packet would be labeled with the same flow identifier and the GNS routers on the network routes associated with the lease would set aside half of their resources on behalf of GNS. This is done in a way that virtualizes the router on behalf of GNS: one can even imagine a scenario in which GNS packets are dropped (due to congestion) although the GDN "side" of the same routers is idle.

It may seem that if GDN has excess resources available, it should in general offer them to GNS, but upon further reflection, it becomes clear that dropping packets when GNS reaches its contracted-for traffic level is necessary if the solution is to work correctly. The problem is that protocols such as TCP are designed to back off as congestion starts to occur. Suppose, hypothetically, that GNS was allowed to expend 20% more bandwidth than it had contracted for simply because GDN has excess capacity. TCP will now be operating at a rate that represents a serious overload for the ON on which GNS is operating. Now, imagine that traffic surges on the GDN side, forcing GDN to cut GNS back to its contracted-for profile of resources. The TCP protocol, rather than seeing "early warning" of impending congestion will suddenly experience high rates of packet loss and a brief period of disruption is very likely – one that would have been avoided if GNS was limited to the profile of services it paid for, in the first place. We see, then, that for an ON to operate correctly, it needs to provide a faithful imitation of a dedicated network even with respect to the limitations of such networks!

Notice that router partitioning yields a kind of Overlay Network, implemented by one ISP on behalf of another ISP. All of the traffic flowing through GDN on behalf of GNS seems to live within a single shared ON, competing with other traffic in the same ON, but not influenced in any way at all by traffic originating in other flows.

Our proposal starts by imagining that we extend this existing mechanism into one that can support perhaps thousands or tens of thousands of flows, but otherwise behaves in the same matter. The idea is that Gotham Hospital might purchase several ONs from GNS, which reserves resources for these ONs on routers associated with routes between the Gotham Hospital communication endpoints (computers). In general, each ON will serve some large number of endpoints. For example, the hospital ICU monitoring system (perhaps extended into the homes of some outpatients) could include several hundred monitoring devices, all sharing a single ON, with a single ON flow identifier. This is in distinction to the connection-oriented perspective that one sees in existing quality of service proposals, such as the ones reviewed earlier. Gotham Hospital purchases aggregated quality of service properties when it leases an ON from GNS, but the only situation in which this translates directly to a point-to-point guarantee is when one knows that the application itself will not generate conflicting traffic. Nonetheless, as in the case of the HP Careview application, one can easily imagine designing applications that carve up the bandwidth of an ON in a careful way. Here, we use the term application to talk about what might in practice be a network configuration having a large number of participating programs at various locations. In effect, any networking application that could be imagined in a dedicated networking environment might make sense in an ON.

Gotham's need for multiple ONs would arise because the hospital has multiple applications with differing and independent needs. For example, Gotham would probably want an ON for its critical care monitoring applications and a separate ON for its clinical database uses. In general, the hospital might have several networks for monitoring, a network for use by the laboratories, a secured network for sensitive clinical data, and a less secured network for billing. But the number of networks is unlikely to be large in comparison to the number of computers connected to the network.

The important insights are two. First, an ON offers strong guarantees of isolation between applications communicating in different flows. This basic property is the key to establishing all sorts of higher level properties. Second, because each ON treats the flow as an aggregation (nothing about the proposal operates on a point-to-point basis), routers will see at worst a linear classification problem. Gotham Hospital, which may operate thousands of computers, would require at most a handful of VONs. Unlike the $O(n^2)$ problem we saw earlier, we now have one that is linear in the number of "enterprises" using the network, a very small number in comparison with the number of endpoints.

V.2 Building VONs over ONs

Gotham will want more than dedicated bandwidth from its ONs. The applications just mentioned are likely to differ in several ways: the monitoring network seeks to emulate the IEEE-1073 standard, hence it may have real-time requirements not seen in the clinical network, which seeks to emulate a dedicated ethernet. It can probably manage with a limited form of security (perhaps, authentication at the time a device is connected to the network and signatures on packets). In contrast, the clinical network may need to encrypt packet contents for reasons of secrecy, and yet derives little benefit from stringent realtime guarantees that may force the network to run slower (in general, communication networks that offer realtime properties are slower than the technology over which they are implemented).

These requirements could go much further. A network used to actually control medical devices should be reliable even if links or routers crash, suggesting that the physical configuration of the resources used to construct the ON may need to be carefully controlled (for example, we might require redundant routes between all pairs of endpoints). A clinical network is unlikely to need such a stringent reliability property, and might not wish to pay the extra cost.

Broadly, we should imagine a world within which the core properties of the ON can be strengthened by layering network management and control software over the endpoints of the ON. An example of such software would be a protocol for the ICU network that sends each packet twice, using independent routes, so that real-time properties will be preserved even if one route is disrupted by a failure. More broadly, the roles of this network infrastructure software extend to a wide range of network management, provisioning, security and control functions. For example:

- A VON might require specialized routing
- We may wish to authenticate the connection of a computer to the VON, and assign it a specialized session key for use while communicating in the VON
- We may need to refresh these session keys periodically, or in cases where a computer leaves the VON and is no longer trusted
- We may wish to coordinate the signing or encryption of data
- We may need to monitor loads and adapt to overload
- The VON may require some special protocol to ensure reliability, such as the duplicate transmission rule outlined above

Abstracting, we can say that a VON resembles what the distributed computing community call a "process group." The desired group communication properties would depend upon the needs of the application, and (in distinction to classical work on distributed computing) the network itself (the ON) would itself offer a basic level of guarantees, reflecting the dedication of resources on its behalf. This last point is important, because it offers a foundation over which much stronger properties can be guaranteed.

For example, consider the TCP protocol. In a conventional network, one cannot really assert that a TCP connection is "reliable", since such a connection can potentially break in the event of an infrastructure disruption, even if neither endpoint has failed. With a TCP connection within a VON, the identical TCP protocol might be able to offer a stronger guarantee, such as the guarantee that if no more than one network link or router fails, the TCP connection will never break unless one of its endpoints crashes. This specific guarantee would require a degree of redundancy within the ON, but one can imagine all sorts of low level guarantees, translating to all sorts of higher level properties visible through TCP and other protocols.

Generalizing from this example, we can see that one builds a VON up from an ON by successively abstracting – layering one or more software-implemented abstractions over the ON, so that each layer extends or strengthens the properties of the stack of layers and ON below it. This is a way of building distributed protocols that has become popular over the past decade: it was introduced as the "streams" architecture of the Unix system then generalized by the *x*-Kernel [Peterson 1989], and adapted to group-structured applications in our work on the Horus [Van Renesse 1996] and Ensemble [Hayden 1998] systems. Recent work has shown how to use formal methods to reason about, optimize, and prove properties of systems structured in this manner [Liu 1999][Birman 2000].

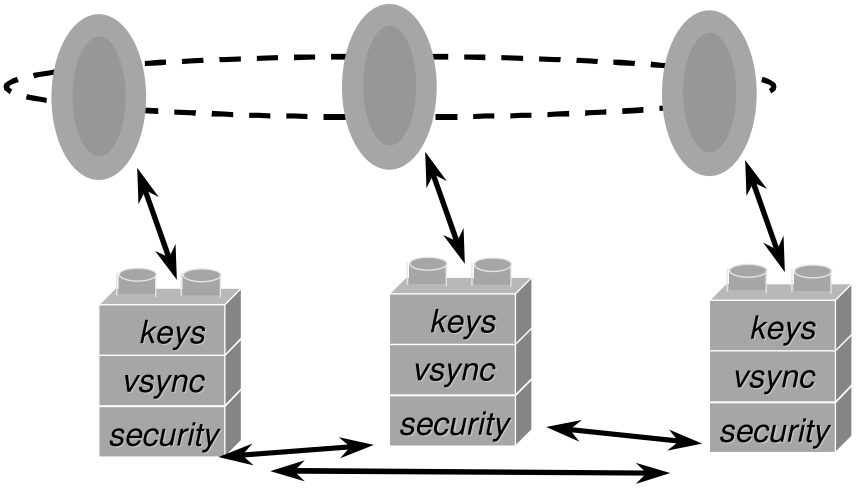

Figure 3: At each of the endpoints of a group (a VON), a stack of protocol elements assists in transforming the properties of the underlying ON into the desired VON properties. The application process would not normally be directly aware of the behavior of the protocol stack. The stack shown supports a model called "secured virtual synchrony" over which security keys might be managed on behalf of the application. VONs with different roles and properties might use different stacks.

Given this perspective, it becomes possible to import a substantial body of knowledge about group communication into the VON arena (Figure 3). The author, for example, has worked on four styles of group communication system:

- Traditional best-effort group communication environments
- Virtually synchronous group communication, with carefully managed group membership and multicast facilities (in addition to traditional point to point mechanisms of the sort normally seen in the Internet)
- Secured group communication systems, providing authenticated join and keying both for the group as a whole and for point-to-point connections, as needed.

- Probabilistic group communication, offering bimodal multicast and probabilistic membership tracking, for use in settings where scalability and stable throughput even when failures occur take precedence over absolute logical guarantees.

Each of these could be understood as the infrastructure one might use in supporting some class of VONs. Moreover, this list illustrates just some of the many data points one can imagine in the spectrum of possible VONs. Others might include support for partition tolerance, mobility, specialized realtime guarantees, other types of security architectures, specialized protocols for video or other media tranmission, infrastructures which integrate management of the VON into application-level mechanisms such as automated restart facilities, and so forth.

The key to our proposal is the idea of successively strengthening core properties by layering software over more basic VONs and ultimately over the underlying ON, with its strong isolation and resource allocation guarantees. Lacking these basic guarantees, one could layer the same sorts of protocols over the Internet, but would generally not arrive at the desired outcome, because the base case for proving many sorts of properties revolves around the properties of the underlying ON, and the isolation of the ON from external interference. Given an ON with even weak isolation properties, however, one can often find ways to strengthen them. In general, one should imagine tradeoffs, such as paying in the form of communication overhead to obtain a higher degree of reliability.

But notice that not every form of ON can be used to support every possible VON. An ON lacking a secured infrastructure may be intrinsically insecure and hence fundamentally inadequate for supporting certain types of secured VONs. An ON that can be severely disrupted by a single router failure would be unable to support a VON requiring continuous steady communication throughput even when routers crash.

V.3 Research Challenges

Our proposal leaves many questions open, and indeed seems to identify a potentially large area for research. While it is easy to lay out a vision of our a world of ONs and VONs could look, before such a network can be constructed we need to know:

- How to specify the properties of an ON or a VON, and to prove that a given protocol, when operated over a particular category of ON, yields a desired VON.
- How to dynamically administer resources on the routers implementing an ON.
- How to perform flow classification rapidly with large numbers of flows.
- How ISPs might share revenue associated with VONs that cross ISP boundaries.
- How to build ONs with certain basic properties not traditionally available in the Internet. For example, it is unclear how to use existing routers to implement an ON that can mask the effect of small numbers of link or router failures.
- How to integrate the resulting functionality into application-level programming tools and environments.
- The right set of services needed in support of VONs, such as PKI's and naming services and services for learning the topology and properties available from the VON in an automated manner.

- How to support security for applications that don't match the traditional security models, such as our telemedicine examples.

These problems will need more study, but none looks insurmountable. In contrast, overcoming the scalability issues cited earlier to implement a VON using existing quality of service proposals appears to be intractable. At best, a VON built using such mechanisms would be extremely expensive and relatively inefficient.

VII. Conclusions

This paper advances an argument that the Next Generation Internet, as presently conceived, is unlikely to provide a safe infrastructure for supporting mission-critical applications. A new but simple mechanism, which we call the Virtual Overlay Network, or VON, is proposed as a partial remedy to the problem. We suggest that VONs could be supported at acceptable cost over components similar to the ones used in the current Internet. Enough is known about building group communication systems with various kinds of properties so that, given a primitive Overlay Network mechanism, VONs could be offered with a wide range of properties and guarantees.

The financier J.P. Morgan is widely quoted as having responded to a question about the future of the stock market with the prediction that "prices will fluctuate." Unless something changes, one could advance a similar speculation with regard to the NGI: "Developers of critical applications will be severely challenged." Yet it is within our technical reach to solve this problem. It would be a great shame if the NGI fails to do so now, when the public is conscious of the need, and the government perceives itself as investing to solve the problem.

VIII. Acknowledgements

This paper was based on a colloquium talk presented at various locations over the period 1998-1999. The author is grateful to the members of the audience at these talks, who gave extremely helpful feedback, notably Peter Wegner, who urged the author to set these thoughts down in written form. Conversations with Danny Dolev and Ohad Rodeh about DVPNs set the stage for our VON proposal. Fred Schneider made many valuable suggestions about the organization of this paper. Finally, the comments of Robbert van Renesse, Farnam Jahanian and Xiaoming Liu are gratefully acknowledged.

IX. References

[Clark 1999] David Clark. High Speed Data Races Home. *Scientific American* 281:4 (Oct. 1999), 94-105.

[Gibbs 1994] Wayt Gibbs. Software's Chronic Crisis. *Scientific American* 276:3 (Sept. 1994).

[PCCIP 1997] Robert Marsh (ed). *Critical Foundations: Protecting America's Infrastructure.* The report of the Presidential Commission on Critical Infrastructure Protection. Oct. 1997

[Schneider 1998] Fred Schneider (ed). *Trust in Cyberspace.* National Academy of Sciences Press (Washington), 1998.

[Freedman and Mann 1997] David Freedman and Charles Mann. *@Large: The Strange Case of the World's Largest Internet Invasion.* Simon and Shuster (New York), 1997.

[Piantoni and Stancescu 1997] Rico Piantoni and Constantin Stancescu. Implementing the Swiss Exchange Trading System. *Proc. 27th FTCS* (Seattle, Washington), 309-313, July 1997.

[Birman 1999] Kenneth Birman. A Review of Experiences with Reliable Multicast. *Software Practice and Experience* 29(9), 741-774 (Sept. 1999).

[Cristian 1996] Flaviu Cristian, Bob Dancey, Jon Dehn. Fault-Tolerance in Air Traffic Control Systems. *ACM TOCS* 14:3 (Aug. 1996), 265-286.

[HP 1997] Hewlett Packard Corporation. Personal communication, August 1997.

[EMTEK 1997] EMTEK Corporation. Personal communication, August 1997.

[Safebus 1994] Todd Carpenter, Kevin Driscoll, Ken Hoyme, and Jim Carciofini. ARINC 659 Scheduling: Problem Definition, *Proc 1994 IEEE Real Time System Symposium*, June 1994.

[Rodeh et. al. 1998] Ohad Rodeh, Ken Birman, Mark Hayden and Danny Dolev. Dynamic Virtual Private Networks. Dept. of Computer Science, Cornell University, Technical Report TR98-1695 (August 1998).

[Zhang 1993] Lixia Zhang, Stephen Deering, Deborah Estrin, Scott Shenker, and Daniel Zappala . RSVP: A New Resource Reservation Protocol. *IEEE Network.* 7:9, pp. 8-18, Sept. 1993

[Floyd and Jacobson 1993] Sally Floyd and Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, 397-413

[Clark and Fang 1997] David Clark and Wenjia Fang, Explicit Allocation of Best Effort Packet Delivery Service. MIT Lab for Computer Science Technical Report, 1997.

[Clark 1995] David Clark and J. Wroclawski. An Approach to Service Allocation in the Internet, Internet Engineering Task Force Draft Report, July 1997.

[Peterson 1989] Larry Peterson, Norm Hutchinson, Sean O'Malley and Mark Abbott. RPC in the *x*-Kernel: Evaluating New Design Techniques. *Proc. 12th SOSP* (Litchfield Park, Az), Nov. 1989, 91-101.

[Van Renesse 1996] Robbert van Renesse, Kenneth P. Birman and Silvano Maffeis. Horus: A Flexible Group Communication System. *Commun. of the ACM* 39:4, 76-83, April 1996.

[Hayden 1998] Mark Hayden. *The Ensemble System.* Ph.D. dissertation, Cornell University Dept. of Computer Science, Dec. 1998.

[Liu 1999] Xiaoming Liu, Christoph Kreitz, Robbert van Renesse, Jason Hicky, Mark Hayden, Kenneth Birman, and Robert Constable. "Building Reliable, High-Performance Communication Systems from Components". To appear, Proc. 17th ACM SOSP, Dec. 1999.

[Birman 2000] Ken Birman, Robert Constable, Mark Hayden, Christoph Kreitz, Ohad Rodeh, Robbert van Renesse, Werner Vogels. The Horus and Ensemble Projects: Accomplishments and Limitations. Proceedings of the DARPA Information Survivability Conference & Exposition (DISCEX'00), Jan. 2000 (Hilton Head, South Carolina).