ON THE SUITABILITY OF COMPOSABLE SERVICES FOR THE ASSURABLE FUTURE INTERNET

Daniel Stevenson
RTI, RTP, NC
dstevenson@rti.org

Rudra Dutta, George Rouskas, Douglas Reeves
NCSU, Raleigh, NC
{rdutta, rouskas, reeves}@ncsu.edu

Ilia Baldine
RENCI, Chapel Hill, NC
ibaldin@renci.org

ABSTRACT

Our SILO architecture for the future Internet consists of composable fine-grain protocol elements called "services", and explicitly enables cross-layer interaction and optimization. While information assurance was not the only goal of SILO, we recognize that a critical need for the global network is a degree of assurability. In this paper, we present our view of the consequences of the SILO architecture with respect to information assurability. We also present some examples of information assurance services that could be easier enabled by SILO than the current architecture.

I. INTRODUCTION

The Internet has changed every aspect of our lives in the past few decades, and has itself changed nearly beyond recognition in the same time. Despite the remarkable ongoing effects of the Internet, there is a widespread perception in the networking community that key limitations of its design might be bringing it close to a breakdown point and a sea-change is necessary in the next decade or so. Recently, the National Science Foundation issued a call for proposals for "clean-slate Internet design". The authors of this position paper include a multi-organization collaborative research team that has been working on such a clean-slate approach to future Internet design called "SILO", funded by a grant from the NSF Future InterNet Design (FIND) program, and security researchers collaborating to articulate the information assurance related strengths and weaknesses of this architecture. In this paper, we discuss our position with respect to some fundamental issues in Internet information assurance, and specifically articulate them with respect to our SILO architecture.

Fundamentally, the SILO architecture generalizes the concept of layering. The building block is a *service*, which takes the place of a protocol layer. Like a protocol layer, it presents a data interface to a served (upper) and serving (lower) service (layer), but in addition, it provides (i) a control interface, which communicates with a unified *control agent*, and (ii) a set of rules for composability, which states what other services this service may be composed with, in what relation. The control agent provides a unique point of security certification and unified security policies. Because the framework does not in itself limit the services which may be presented to the control agent for composition, incorporating new security services reflecting an evolving security policy is seamlessly supported by the architecture. We have previously published details of the SILO architecture [5], [10]. Further information about the SILO project can be found at the SILO website [26], which also contains technical documentation archiving the ongoing activities of our group.

We do not claim that our clean-slate architecture solves all current or future security problems, far from it. Indeed, it is possible and quite likely that the additional flexibility afforded by composable protocols may create new security vulnerabilities. We would venture to suggest that any attempt at clean-slate design cannot guarantee to foresee all the security implications of a new proposed architecture. What we do assert is something more modest and yet at the same time more realistic and practically valuable: that our proposed architecture has unique features that provides a systematic approach to enforcing integrated security policies, that it supports smooth evolution of security features, and accommodates within itself the means to identify new security threats and respond to it. As such, we view the network that we are envisaged not as being "assured", but as being "assurable"; in what follows, we thus refer to the envisioned network as the Assurable Future Internet (AFI).

A. "Design Criteria 2.0" - Designing the AFI

In a now classic paper, David Clark articulated the original prioritization of the design philosophies behind Internet architecture [8]. As pointed out in that paper, if the design goals had been different, or even merely the prioritization, the design of the Internet would likely have taken very different pathways. In the same spirit, we present the following prioritized list; it is meant to focus on pertinent issues rather than be a comprehensive list. We realize that such principles must and will be debated in the community for years to come. This process will refine them; nevertheless, we believe we have captured several key points in this list that will survive the test.

High Availability Information Delivery. Given the central role of the network in the current network-cenrtic warfare

doctrine of DoD, this must be considered the prime requirement.

Verifiably Secure Information Delivery. The next priority for the AFI is information assurance, and includes various aspects such as authentication, accountability, confidentiality, etc. Note that we have chosen to subjugate the security requirement to that of availability of the communication function, but this comes with a goal of informing the user of what security features are available at any time or place.

Support for Mobility. For the AFI, this goal stems from the DoD NCW doctrine that relies heavily on networking and mobility. We note that this is complemented by a similar requirement in the private sector, motivated by the increasing use of ubiquitous computing.

Interworking Flexibility and Extensibility. The perfect right answer of one day is a limiting liability of the next. We believe that the AFI must not be saddled with a rigid framework that hampers growth. However, some framework must be created in which the future architecture can exist, it cannot exist in a vacuum. The design principle that emerges from this is the requirement for a minimal core framework, and a *meta*-framework to support experimentation and innovation.

Support for a Scalable, Unified Network. Currently, we are witnessing a growing gap between commodity applications running in today's Internet on the one hand, and other novel uses of the network that impose very dissimilar requirements on the other, such as E-science applications with their extreme bandwidth demands, or wireless ubiquitous and sensing applications with extreme mobility support requirements. The network of the future must provide an architecture where such diverse requirements can be expressed as different options of interacting with the same unified network, otherwise a fragmentation will occur resulting in several parallel networks.

Explicit Facilitation of Cross-Layer Interactions. Existing protocol stacks lack well-defined control interfaces for cross-layer interactions, hence the latter have to be engineered in a piecemeal and ad-hoc fashion, which are often detrimental to security. The future architecture must have explicit built-in ability for functional blocks to interact with each other to optimize the behavior of the network, as required by the specific communication task at hand.

Smooth Integration of Evolving Security Policies. Carefully crafted network security features, based on cryptography or other software-intensive techniques, often get bypassed for a variety of reasons, especially due to ignorance, and configuration or performance frustrations. This is primarily a result of the fact that security has been largely considered an add-on, and not designed integrally with the architecture. In one of the earliest design principles above, we have already required the network to understand security semantics, and integrate it with the very service of communication. However, as we already remarked, the network must allow growth and change; this applies also to the security policy.

Distribution of Data and Control. The previous two principles lead us to conclude that data and control must be separated within the architecture at all points. Separate channels must be used in communication, where high volumes of data flow must not throttle transport of control information - this is a lesson that has been learned many times over in various contexts. Data and control must also be separated at network end nodes and intermediate switching/routing nodes, with separate guaranteed processor and buffer allocations for control functions.

II. THE NEED FOR A NEW ARCHITECTURE

The Internet as it stands today can hardly be called "broken"; it continues to serve surprisingly well considering that the uses it is put to today are far removed from the original visions that led to its inception. Yet the NSF has seen fit to trigger an effort for "clean-slate design" of the future Internet, and this has been widely acknowledged as a timely step. What is perceived as being wrong with the Internet?

This can be partly answered by the old saying "nothing wrong - just not enough right". Many functionalities that would be highly desirable by the increasingly pervasive and diversified nature of network applications are not supported by the network architecture, as are challenges posed by the changing nature of the physical network itself. Solutions to such problems are either absent, or implemented as a workaround for specific contexts. Consider, for example, the recent research on TCP variants for high bandwidth-delay product networks [12], [15], [18], [22], [33], earlier work on TCP over wireless networks [4], [6], [7], [34], and current efforts towards cross-layer optimization [19], [21], [24], [28], [30], [32]. The network also does not extend easily to power-bandwidth-cpu costrained sensor networks; at this time the most popular approach is the use of a middleman base station, which fragments the networks. Delay and disconnection tolerance requirements force the same kind of mechanisms.

More seriously, such uneasy and ad-hoc solutions violate original design principles, and obliterate architectural unity. In other cases, addressing broader needs such as IP address shortage or evolving security requirements creates creates the same problems. Like a hydra, multiple new problems arise after every new solution. Network Address Translation is not consistent with the end-to-end principle,

and this poses an accountability problem, even though NAT is itself a component of firewall security. IPSec and fragmentation do not co-exit comfortably [35], and IPSec violates the layering principle, being introduced at layer 2.5, conceptually. Similarly, transport layer security solutions have been introduced at layer 3.5. We are forced to conclude that what is wrong with the Internet is *this very approach of incremental ad-hoc fixes.*

Increasing capabilities of technology also create problems. Recent increase of Internet penetration and the rise of P2P communication models have created vulnerability to demand saturation, as pointed out by measurements of BitTorrent traffic [11] and the case of the mass download of the Starr Report [1]. IP address spoofing was never considered a serious threat because of the "prohibitive" amount of computation required, until Kevin Mitnick accomplished it, more than a decade ago [14]. Port scanning was considered a rare oddity, but is now one of the most common vulnerabilities of hosts. Distributed DOS attacks remain a vulnerability for the large part, basic Internet workhorses such as OSPF have well-known vulnerabilities [23], [27], [31], and it is generally acknowledged that spam-fighting and Human Interaction Proof techniques keep barely one-step ahead of the attackers [13].

We are led back inescapably to our design principles of Flexibility and Extensibility, and Smooth Integrability of Evolving Security. This is the fundamental job of the software architecture of the future AFI: support an evolving network without breaking down, enable agile design responses to emerging threats and practices without fragmenting; in short : *design for change*.

A. What to Keep?

Clearly, there is much that is good in the current architecture of the Internet, and would be worth retaining in the AFI, or a commodity network. We mention only a few salient ones. The original Internet design goal of robustness in the face of outages needs to remain, and indeed needs to be strengthened, especially for the AFI. In particular, continued operation in the face of multiple failures, not just one, is clearly required for the AFI. However, we must recognize that once again a matter of choice is raised - the network architecture must not mandate recovery from multiple failures for all users of the network, but provide the service for those users who require it.

One of the original design criteria which has continued to be represented in the architecture is the requirement to present a low barrier to entry. In general, the ability to execute the protocol has been the single requirement to connect to the network and to obtain all privileges. We recognize the unifying power of the low-barrier principle, and believe that this principle is more important than ever in the face of the highly diverse heterogeneous network of the future, containing as they will extremely simple devices such as sensor/actuators or embedded systems. However, again this principle must be refined - we comment extensively on this in the next section.

The Internet architecture has also posed a comparatively low barrier to being able to implement innovative applications in this architecture. In turn, such applications have served as an agent to changing the very social dynamic of our society; we need only consider the effects of email, web, you-tube, myspace to find examples. The Internet has done more to foster freedom of thought and mass participation in critical thinking than any development since the printing press. Changes paralleling these have occurred within the specific context of the DoD. Examples include the use of the Internet and web-based interfaces to bridge the technical gaps in integrating information flows across legacy systems and in accelerating decision processes [2], [3], [25], [29]. This element must remain in the AFI.

B. ...And What Must Go?

There are many concepts in use in the Internet and many aspects of its current architecture that can be identified as the cause of some limitation or problem. However, as we have seen above, sometimes the source of the problem was an intended fix, and hardly something that can be blamed. Many disagreements can exist regarding the details of which specific concepts or mechanisms should be kept and which abandoned. We have attempted to identify more general and basic concepts; we believe these identify crucial large changes which must be made.

The first one relates to the issue of security, and is particularly pertinent from the AFI perspective. Currently, security is not a concept that is represented in the network at all; this must change. Security semantics must be understood by the network architecture, in the same way that semantics of endpoint, forwarding, routes, address, etc. are. Security solutions must not be considered an add-on, something that is imposed only by higher layers. In the AFI, semantics relating to trust will be embedded into other architectural functions. A step in a routing algorithm might read: "When an LSA is received from a currently trusted router about a link to a currently untrusted router ..." which is different from the all-trusted model that is implemented today by failing to include security semantics in the architecture. As part of this, the operationally default mode of no accountability will also change.

The second change we see as inevitable can be described as *network striation* or *stratification*, and is a combined

response to several architectural needs. In the first place, we shall argue in Section III-C that the network must be an enabler, and must not take up decisive positions on either sides of "tussles in cyberspace". This implies that different users will require, and receive, diffrent views of the network. Simple sensor devices of low power and low intelligence will see a network that can be operated simply, without heavy protocols. However, other endpoints that are less constrained will see a more complicated network with many more value-added services and options. A unified architecture can only support both by presenting different semantics to the two. The set of services offered to the sophisticated device may well be a superset of those offered to the simplistic device. Thus, paradoxically, the network must become stratified in order to avoid being fragmented. This is also motivated by the issue of security; the low barrier to entry must be accompanied by several *levels* of entry. Endpoints that are unable to provide secure connection protocols can still connect to the network, but will see only a small subset of the services and resources offered by the network; endpoints and applications that can establish more trust will see a richer network. This concept is related to other emerging models of secure networking, such as the "default-off" model advanced by McKeown.

The presentation of different views of the network to different applications may appear to be worrisome because it implies loss of complete transparency, thus violating the end-to-end principle. Rather than being daunted by this, we embrace this as a required change. We contend that the principle in question is no longer valid, it is a mechanism that has been confused with a goal, and the correct goal is not complete transparency but flexibility and extensibility. Indeed, part of the motivation for the principle was the desire to push complexity away from the network core and to the edges - creating the "smart endpoint, dumb core" model that was a reversal of the telephony model. However, the AFI we envision and the stratified network we postulate cannot afford to be dumb, and we advocate a "any endpoint, smart core" model.

Finally, we believe that the concept of protocol layering has ossified in the current architecture, and must be rejuvenated. While layering remains useful as a concept, the rigidity and lack of variety in the layering model as currently used poses essential limitations, with particular consequences for security. Rejuvenating layering is also at the heart of our proposed clean-slate architecture.

III. LAYERING AND COMPOSABILITY

Layering was one of the earliest design principles identified for communication software, and it has been one of the most influential. The paradigm has been questioned many times, but it has in general withstood all challenges and the test of time. There has also been a certain degree of reluctance on the part of the networking community to seriously consider alternatives. We feel that this stems from two basic reasons, and the practical success of layered software (a highly rational reason) is only one of them. The other is a reason related to legacy - layering was not imposed by design decision in the earliest networking software, but instead was observed as an emergent quality; powerful evidence of the suitability of this paradigm.

The two basic roles played by layering in networking are (1) modularization, and (2) abstraction. The first allows development of the software and hardware in functional modules, which is further drastically constrained by only allowing a module to interact functionally with a single upper and a single lower layer. The second, abstraction, is primarily useful in creating maintainability, because diverse implementations of different functionalities can be most easily plugged in or out when no layer is aware of how any other layer performs its functions. This goal is helped along by the constrained modularity offered by layering.

In rethinking the role of layering, we have come to the conclusion that the apparent shortcoming of layering is not a characteristic of layering itself, but the rigidity that the concept has acquired over the decades; a rigidity, we believe, that formed no part of the original vision of layering. The proliferation of custom cross-layer control and optimization strategies in the literature show that the maintainability benefits of a rigid adherence to the principle of abstraction is more than offset by the opportunity cost in the inability to optimize across layers. The proliferation of half-layer solutions and custom solutions involving layer inversions show that all functionality may not be positioned at the same place in the layers for all uses of the network. These considerations have culminated in the conception of the SILO framework.

A. The SILO Architecture

In essence, the SILO architecture generalizes the concept of layering. Next, we briefly describe the essential points of SILO; for a more complete description, see [10] or [5].

Service and Methods: Fine-grained protocols. The fundamental building blocks in the SILO framework are *services*. A service is a well-defined and self-contained function performed on application data, and addresses a separate, simple and reusable function. Hence the architecture provides a much finer granularity than current protocols which typically embed complex functionality. Each service provides an upper and lower data interface (as today), but also provides a minimal control interfaces which we also refer to as *knobs*. Finally, each service

provides a list of partial ordering constraints with respect to other services. Beyond these constraints, any set of services can be composed into a stack (a "silo" in our terms) in any order. A *method* is a realization of a service that uses a specific mechanism to carry out the functionality associated with the service. A silo structure and all related state information are associated with a specific traffic stream and persist for the duration of the connection.

Unified Control. A *control agent* is an entity residing inside a node, which is responsible for (1) composing a silo for an application stream, and (2) appropriately adjusting all the service- and method-specific knobs and facilitating cross-service interactions. Composing a silo refers to determining the subset of services it contains, their order in the stack, and the method implementing each service. The objective is to dynamically custom-build a silo for each new connection. To this end, the control agent takes into account the application's QoS requirements, current network resource availability and other conditions, the precedence constraints among services, and any policy in effect at the time.

Cross-layer optimization. The silo approach can simply be viewed as "transport in layers, control and secure across layers." As today, services (protocols) and methods are required only to provide a minimal interface, hiding internal details. However, traditional protocols are only required to provide invocation methods (APIs), whereas in the SILO framework we require them to provide a minimal *control interface* as well. Beyond this, the methods can be designed and implemented in isolation as before. However, the control agent has a unique view into the knobs of every method in the silo, and can embody all the integrated control concerns. In this way, "cross-layer" (or, more appropriately, "cross-service") is accommodated as part of the mainstream architecture.

B. Composable Services for the AFI

We do not claim that the SILO framework is in itself a solution for current or future security problems. Rather, we have argued before that the job of the network architecture is primarily to enable a wide variety of security features, and to support smooth integration of evolving security services on an ongoing basis. We now propose that the SILO framework, as described above, is such an architecture, because: (1) it enables organic integration of security features and services into the AFI architecture, (2) it allows for a consistent implementation of needed security mechanisms (i.e. regardless of their placement within a SILO stack, they only need to be implemented, and their correctness verified, once), and (3) the explicit cross-layer interactions of the SILO framework allow for

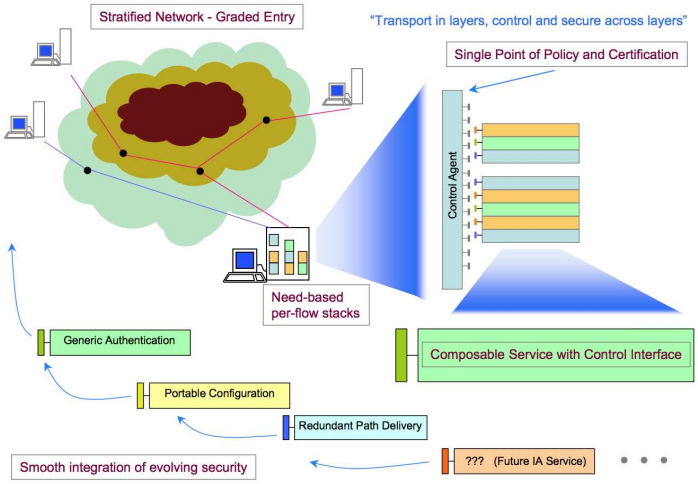

Fig. 1. Vision of Assurable Future Internet enabled by SILO

easier enforcement and verification of enacted security policies, both in end nodes and in the core of the network.

In this subsection, we discuss some specific issues related to information assurance in the light of our architecture. Figure 1 shows both a brief representation of the SILO framework and its role in the envisioned AFI, and illlustrates the issues discussed below.

Unique Certification Point. The control agent forms a unique point at which security policies may be embedded, and certifications may be held. In effect, the control agent certifies the entire communication stack, since it has a control interface to each service. In turn, it presents a unique certification interface to the operating system. The control agent embeds the unified security policy.

This circumvents one of the worst ills of layered software in terms of security, that it is never possible for layered software to know exactly what is layered above or below it in the overall stack. Even sophisticated security mechanisms are helpless if some of their software components are bypassed, or they are stacked over some protocol that opens up a vulnerability. With SILO, neither innocent bypassing nor malicious masquerading of parts of the security services are possible for since they are all visible to the control agent.

Striated Network - Graded Entry. The SILO framework enables network striation since security policies and control agents at endpoint and intermediate nodes can ensure that endpoints which connected with only a certain level of privilege. Also, only endpoints with a certain level of demonstrated trust would be able to access the appropriate level of the striated network. This feature also provides an answer to a growing concern about increasing integration in the global network. Separation of functions is less efficient, but comes with a certain degree of automatic

protection. However, the striation of the network into many different service and trust levels (together with the obvious step of control and data separation) removes this problem. Integration of the network is offset by graded entry into the different trust levels of the network.

Portable Configuration - "My Network". Many information assurance concerns stem ultimately from the human-computer boundary, at which there are several concerns, one of which is correctly carrying over the human's security policies and other preferences between different computers used by the human. Every user of any IT system customizes the system to some extent. However, many choices regarding the local behavior (which refers to the local end system) and service behavior (which refers to both local and remote end systems) in fact relate to the behavior of the network, even though the user perceives them as being related to the local IPD or the service. For example, consider that most personal computers enabled with 802.11 network cards provide the user with a choice to "turn interference robustness ON or OFF" or similar. However, internally this results in the use of optional parts of the networking protocol such as RTS/CTS. Naturally, such a decision cannot be supported by the 802.11 client card alone, the access point must also configure its behavior accordingly. Such a decision on the part of the user is by its very nature dynamic - when operating in a comparatively interference-free region, the user would naturally like to obtain the greater efficiency obtained by turning the mechanism OFF. Such preference sets can become complex, and it becomes a liability to maintain them separately on every IPD the user uses, leading to inconsistent security policies and possibly bypassing secuirty mechanisms.

The SILO framework provides the user with an interface to specify preferences and policies: the control agent. In fact, these settings are what the control agent translates into optimal knob settings. This provides the added advantage of portability. Currently, such user settings would be embedded in the software resident on a particular IPD, or server. Within SILO, the user could well carry around a small memory stick or similar device which is itself passive, but is equipped with a nearly universal interface such as USB. When this is attached to the IPD, the control agent can read off the settings, and immediately implement the service preference profile of the user. This concept may be called "My Network". We note that this addresses one of the important visions of the AFI, that of providing seamless and secure network experience to a mobile user.

Service Examples. We discuss two possible information assurance services to show the diverse range of services that can be integrated into the SILO framework. The first is the obvious one of *generic authentication*. Authentication is a very common function and is required at many points in communication. However, at this time, the approach is for every protocol that needs it to embed an implementation separately, resulting in needless redundancy at the transport and application layers. From the SILO perspective, these functionalities perform the same service, and can be realized only once, with a trusted and certified implementation. An instance of this implementation would be composed into the stack at every place it is required.

Another example is that of *redundant delivery on disjoint paths*. Recovery from failures or continued service under partial failures or attacks is a typical requirement of the AFI. Usually this is thought of as either a network layer function, that of implementing alternate routing in the presence of failure (protection switching), or a transport layer function, that of guaranteeing delivery by retransmission, or a combination of both. However, with the SILO framework, we can conceive of the new redundant delivery service, which attempts to transmit the data requesting this service along two physically disjoint (or risk-disjoint) paths to the same destination. While such a service could be created today, we note that it would require a completely new protocol, together with a great deal of cross-layer interaction, which would in turn create brittle interactions. The advantages of the SILO framework is obvious in this context; integrating the new service would be supported by the framework, the service would require other services which provide information about the routing as helpers, and the cross-layer interaction would be explicitly enabled.

C. Orthogonality and Composability

The question of whether network design should embed enabling of disabling specific services or types of services arises periodically, and has recently sparked off the "net-neutrality" debate. Specific questions usually involve what has been described as "tussles in cyberspace", where two opposing interests in a commercial or other space both attempt to enforce their points of view by mechanisms embedded in networks [9]. Examples include guarantees of privacy and anonymity versus support for legitimate and authorized governmental wiretaps, or the desire to offer premium services selectively and exclusively to customers who are profiled as being able to pay for them. We believe that the network should be designed not to enforce one position, but to enable both with a unified structure, in such cases. For example, for the above tussles, we argue that privacy is a network service, like key escrow. The availability of such services may be regulated or limited, but it is not the place of the designer to cripple the network by not supporting *either* the option of privacy *or* that of surveillance. A key requirement in the above is the

desire for a *unified* architecture; we stand with the opinion (expressed in a recent talk [17]) of Bob Kahn, one of the earliest architects of the Internet, that adding any number of service preferences are acceptable, as long as they do not fragment the Internet, causing the "balkanization" many have feared [20]. Many of these issues are orthogonal to network design. While basic security related semantics must be built into the architecture, all issues which are not absolutely required for network operation must be implemented as optional services, which are composed into the stack by the SILO control agent as and when required by the appropriate security policy. We note that composable services are ideally suited to realize this orthogonality.

IV. CONCLUSION - THE ROAD TO THE AFI

Together with the original SILO project, we are also developing further the concept of an AFI, realized (in part) by SILO. The current focus of the SILO effort lies in primarily automating the construction of the SILO service stacks and designing the SILO control algorithms. Adding a security dimension to our ideas and implementing one or more prototypes embodying those dimensions will present an excellent early opportunity to test the viability of our proposed approach. Our early prototype will incorporate the concepts described above, albeit in a limited or simplified form. This prototype can then be red-team reviewed by security experts in order to confirm its correctness with respect to implementing desired security services, and identify weaknesses. We believe that the AFI can be substantially realized in the same timeframe of 10-15 years proposed by NSF to put together the "clean-slate" design developed within the FIND program.

REFERENCES

[1] The Starr Report: Old media bows before the internet. http://www.webreference.com/outlook/extra3/page3.html, 1998.

[2] D.S. Alberts, J.J. Garstka, and F.P. Stein. Network centric warfare: Developing and leveraging information superiority. CCRP Publications, 2000.

[3] D.S. Alberts and R.E. Hayes. 2003.

[4] H. Balakrishnan, , V. N. Padmanabhan, S. Seshan, and R. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *ACM SIGCOMM '96* , pages 256–269, Aug. 1996.

[5] Ilia Baldine, Manoj Vellala, Anjing Wang, George Rouskas, Rudra Dutta, and Daniel Stevenson. A unified software architecture to enable cross-layer design in the future internet. *ICCCN* 2007.

[6] K. Brown and S. Singh. M-TCP: TCP for mobile cellular networks. *Computer Commun. Review*, 27(5):19–43, Oct. 1997.

[7] R. Caceres and L. Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE JSAC*, 13(5):850–857, June 1995.

[8] D. D. Clark. The design philosophy of the DARPA internet protocols. *ACM SIGCOMM*, pages 106–114, Aug. 1988.

[9] D. D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in cyberspace: Defining tomorrow's internet. *ACM SIGCOMM*, pages 347–356, Aug. 2002.

[10] Rudra Dutta, George N. Rouskas, Ilia Baldine, Arnold Bragg, and Dan Stevenson. The SILO architecture for services integration, control, and optimization for the future internet. *IEEE ICC* 2007.

[11] Leslie Ellis. BitTorrent's swarms have a deadly bite on broadband nets. http://www.multichannel.com/article/CA6332098.html, 2006.

[12] S. Floyd. HighSpeed TCP for large congestion windows. IETF Internet Draft <draft-floyd-tcp-slowstart-01.txt>, 2003.

[13] J. Goodman, G.V. Cormack, and D Heckerman. Spam and the ongoing battle for the inbox. *CACM*, Feb 2007.

[14] K. Hafner and J. Markoff. Cyber punk - outlaws and hackers on the computer frontier. 1995. ISBN: 1-872180-94-9.

[15] C. Jin, D. X. Wei, and S. H. Low. FAST TCP: Motivation, architecture, algorithms, performance. *IEEE INFOCOM*, pages 2490–2501, March 2004.

[16] S.T.Redwine Jr. and W.E.Riddle. Software technology maturation. *8th International Conf. on Softw. Engr.*, pages 189–200, 1985.

[17] Robert Kahn. On net neutrality. Video: http://blog.sektormedia.org/2007/01/robert_kahn_on_.html, 2007.

[18] D. Katabi, M. Handley, and C. Rohrs. Tussle in cyberspace: Defining tomorrow's internet. *ACM SIGCOMM*, pages 89–102, Aug. 2002.

[19] R. Madan, S. Cui, S. Lall, and A. Goldsmith. Cross-layer design for lifetime maximization in interference-limited wireless sensor networks. *IEEE INFOCOM*, Mar. 2005.

[20] Computer Business Review Online. ITU head foresees internet balkanization, November 2005.

[21] V. T. Raisinghani and S. Iyer. Cross-layer feedback architecture for mobile device protocol stacks. *IEEE Communications*, 44(1):85-92, Jan. 2006.

[22] S. Ravot. Grid TCP. http://netlab.caltech.edu/FAST/meetings/2002july/GridTCP.ppt.

[23] S.F.Wu, H.C.Chang, F.Jou, F.Wang, F.Gong, and C.Sargor. JiNao: Design and implementation of a scalable intrusion detection system for the OSPF routing protocol. *Computer Networks*, 1999.

[24] V. Srivastava and M. Motani. Cross-layer design: A survey and the road ahead. *IEEE Communications*, 43(12):112-119, Dec. 2005.

[25] D. Stevenson. Net-centric operations issues. Personal communication with DoD personnel.

[26] SILO Design Team. Services Integration, controL and Optimization for the Future Internet. http://www.net-silos.net.

[27] Brian Vetter, Feiyi Wang, and S.Felix Wu. An experimental study of insider attacks for the ospf routing protocol. *ICNP*, Oct. 1997.

[28] J. Wang, L. Li, S. H. Low, and J. C. Doyle. Cross-layer optimization in TCP/IP networks. *IEEE/ACM Transactions on Networking*, 13(3):582–595, June 2005.

[29] Wikipedia. Network centric warfare. http://en.wikipedia.org/wiki/Network-centric_warfare.

[30] R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet. CrossTalk: Cross-layer decision support based on global knowledge. *IEEE Communications Magazine*, 44(1):93–99, January 2006.

[31] Shyhtsun F. Wu, Fei yi Wang, Brian M. Vetter, Rance Cleaveland, Y. Frank Jou, Fengmin Gong, and Chandramouli Sargor. Intrusion detection for link-state routing protocols. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1997.

[32] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S-Y. Kung. Network planning in wireless ad hoc networks: A cross-layer approach. *IEEE JSAC*, 23(1):136–150, January 2005.

[33] L. Xu, K. Harfoush, and I. Rhee. Binary increase congestion control (BIC) for fast long-distance networks. *IEEE INFOCOM*, pages 2514–2524, March 2004.

[34] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen. TCP performance issues over wireless links. *IEEE Communications Magazine*, 39(4):52–58, April 2001.

[35] G. Ziemba, D. Reed, and P. Traina. Security considerations for ip fragment filtering. Request for Comments, 1858, 1995.