Some Internet Architectural Guidelines and Philosophy

Status of this Memo

Copyright Notice

Abstract

   This document extends RFC 1958 by outlining some of the philosophical
   guidelines to which architects and designers of Internet backbone
   networks should adhere. We describe the Simplicity Principle, which
   states that complexity is the primary mechanism that impedes
   efficient scaling, and discuss its implications on the architecture,
   design and engineering issues found in large scale Internet
   backbones.

Table of Contents

1. Introduction

   RFC 1958 [RFC1958] describes the underlying principles of the
   Internet architecture. This note extends that work by outlining some
   of the philosophical guidelines to which architects and designers of
   Internet backbone networks should adhere. While many of the areas
   outlined in this document may be controversial, the unifying
   principle described here, controlling complexity as a mechanism to
   control costs and reliability, should not be. Complexity in carrier
   networks can derive from many sources. However, as stated in
   [DOYLE2002], "Complexity in most systems is driven by the need for
   robustness to uncertainty in their environments and component parts
   far more than by basic functionality". The major thrust of this
   document, then, is to raise awareness about the complexity of some of
   our current architectures, and to examine the effect such complexity
   will almost certainly have on the IP carrier industry's ability to
   succeed.

   The rest of this document is organized as follows: The first section
   describes the Simplicity Principle and its implications for the
   design of very large systems. The remainder of the document outlines
   the high-level consequences of the Simplicity Principle and how it
   should guide large scale network architecture and design approaches.

2. Large Systems and The Simplicity Principle

   The Simplicity Principle, which was perhaps first articulated by Mike
   O'Dell, former Chief Architect at UUNET, states that complexity is
   the primary mechanism which impedes efficient scaling, and as a
   result is the primary driver of increases in both capital
   expenditures (CAPEX) and operational expenditures (OPEX). The
   implication for carrier IP networks then, is that to be successful we
   must drive our architectures and designs toward the simplest possible
   solutions.

2.1. The End-to-End Argument and Simplicity

   The end-to-end argument, which is described in [SALTZER] (as well as
   in RFC 1958 [RFC1958]), contends that "end-to-end protocol design
   should not rely on the maintenance of state (i.e., information about
   the state of the end-to-end communication) inside the network. Such
   state should be maintained only in the end points, in such a way that
   the state can only be destroyed when the end point itself breaks."
   This property has also been related to Clark's "fate-sharing" concept
   [CLARK]. We can see that the end-to-end principle leads directly to
   the Simplicity Principle by examining the so-called "hourglass"
   formulation of the Internet architecture [WILLINGER2002]. In this
   model, the thin waist of the hourglass is envisioned as the
   (minimalist) IP layer, and any additional complexity is added above
   the IP layer. In short, the complexity of the Internet belongs at
   the edges, and the IP layer of the Internet should remain as simple
   as possible.

   Finally, note that the End-to-End Argument does not imply that the
   core of the Internet will not contain and maintain state. In fact, a
   huge amount coarse grained state is maintained in the Internet's core
   (e.g., routing state). However, the important point here is that
   this (coarse grained) state is almost orthogonal to the state
   maintained by the end-points (e.g., hosts). It is this minimization
   of interaction that contributes to simplicity. As a result,
   consideration of "core vs. end-point" state interaction is crucial
   when analyzing protocols such as Network Address Translation (NAT),
   which reduce the transparency between network and hosts.

2.2. Non-linearity and Network Complexity

   Complex architectures and designs have been (and continue to be)
   among the most significant and challenging barriers to building cost-
   effective large scale IP networks. Consider, for example, the task
   of building a large scale packet network. Industry experience has
   shown that building such a network is a different activity (and hence
   requires a different skill set) than building a small to medium scale

network, and as such doesn't have the same properties. In
particular, the largest networks exhibit, both in theory and in
practice, architecture, design, and engineering non-linearities which
are not exhibited at smaller scale. We call this Architecture,
Design, and Engineering (ADE) non-linearity. That is, systems such
as the Internet could be described as highly self-dissimilar, with
extremely different scales and levels of abstraction [CARLSON]. The
ADE non-linearity property is based upon two well-known principles
from non-linear systems theory [THOMPSON]:

2.2.1. The Amplification Principle

The Amplification Principle states that there are non-linearities
which occur at large scale which do not occur at small to medium
scale.

COROLLARY: In many large networks, even small things can and do cause
huge events. In system-theoretic terms, in large systems such as
these, even small perturbations on the input to a process can
destabilize the system's output.

An important example of the Amplification Principle is non-linear
resonant amplification, which is a powerful process that can
transform dynamic systems, such as large networks, in surprising ways
with seemingly small fluctuations. These small fluctuations may
slowly accumulate, and if they are synchronized with other cycles,
may produce major changes. Resonant phenomena are examples of non-
linear behavior where small fluctuations may be amplified and have
influences far exceeding their initial sizes. The natural world is
filled with examples of resonant behavior that can produce system-
wide changes, such as the destruction of the Tacoma Narrows bridge
(due to the resonant amplification of small gusts of wind). Other
examples include the gaps in the asteroid belts and rings of Saturn
which are created by non-linear resonant amplification. Some
features of human behavior and most pilgrimage systems are influenced
by resonant phenomena involving the dynamics of the solar system,
such as solar days, the 27.3 day (sidereal) and 29.5 day (synodic)
cycles of the moon or the 365.25 day cycle of the sun.

In the Internet domain, it has been shown that increased inter-
connectivity results in more complex and often slower BGP routing
convergence [AHUJA]. A related result is that a small amount of
inter-connectivity causes the output of a routing mesh to be
significantly more complex than its input [GRIFFIN]. An important
method for reducing amplification is ensure that local changes have
only local effect (this is as opposed to systems in which local
changes have global effect). Finally, ATM provides an excellent
example of an amplification effect: if you lose one cell, you destroy

the entire packet (and it gets worse, as in the absence of mechanisms such as Early Packet Discard [ROMANOV], you will continue to carry the already damaged packet).

Another interesting example of amplification comes from the engineering domain, and is described in [CARLSON]. They consider the Boeing 777, which is a "fly-by-wire" aircraft, containing as many as 150,000 subsystems and approximately 1000 CPUs. What they observe is that while the 777 is robust to large-scale atmospheric disturbances, turbulence boundaries, and variations in cargo loads (to name a few), it could be catastrophically disabled my microscopic alterations in a very few large CPUs (as the point out, fortunately this is a very rare occurrence). This example illustrates the issue "that complexity can amplify small perturbations, and the design engineer must ensure such perturbations are extremely rare." [CARLSON]

2.2.2. The Coupling Principle

   The Coupling Principle states that as things get larger, they often exhibit increased interdependence between components.

   COROLLARY: The more events that simultaneously occur, the larger the likelihood that two or more will interact. This phenomenon has also been termed "unforeseen feature interaction" [WILLINGER2002].

   Much of the non-linearity observed large systems is largely due to coupling. This coupling has both horizontal and vertical components. In the context of networking, horizontal coupling is exhibited between the same protocol layer, while vertical coupling occurs between layers.

   Coupling is exhibited by a wide variety of natural systems, including plasma macro-instabilities (hydro-magnetic, e.g., kink, fire-hose, mirror, ballooning, tearing, trapped-particle effects) [NAVE], as well as various kinds of electrochemical systems (consider the custom fluorescent nucleotide synthesis/nucleic acid labeling problem [WARD]). Coupling of clock physical periodicity has also been observed [JACOBSON], as well as coupling of various types of biological cycles.

   Several canonical examples also exist in well known network systems. Examples include the synchronization of various control loops, such as routing update synchronization and TCP Slow Start synchronization [FLOYD,JACOBSON]. An important result of these observations is that coupling is intimately related to synchronization. Injecting randomness into these systems is one way to reduce coupling.

Interestingly, in analyzing risk factors for the Public Switched
Telephone Network (PSTN), Charles Perrow decomposes the complexity
problem along two related axes, which he terms "interactions" and
"coupling" [PERROW]. Perrow cites interactions and coupling as
significant factors in determining the reliability of a complex
system (and in particular, the PSTN). In this model, interactions
refer to the dependencies between components (linear or non-linear),
while coupling refers to the flexibility in a system. Systems with
simple, linear interactions have components that affect only other
components that are functionally downstream. Complex system
components interact with many other components in different and
possibly distant parts of the system. Loosely coupled systems are
said to have more flexibility in time constraints, sequencing, and
environmental assumptions than do tightly coupled systems. In
addition, systems with complex interactions and tight coupling are
likely to have unforeseen failure states (of course, complex
interactions permit more complications to develop and make the system
hard to understand and predict); this behavior is also described in
[WILLINGER2002]. Tight coupling also means that the system has less
flexibility in recovering from failure states.

The PSTN's SS7 control network provides an interesting example of
what can go wrong with a tightly coupled complex system. Outages
such as the well publicized 1991 outage of AT&T's SS7 demonstrates
the phenomenon: the outage was caused by software bugs in the
switches' crash recovery code. In this case, one switch crashed due
to a hardware glitch. When this switch came back up, it (plus a
reasonably probable timing event) caused its neighbors to crash When
the neighboring switches came back up, they caused their neighbors to
crash, and so on [NEUMANN] (the root cause turned out to be a
misplaced 'break' statement; this is an excellent example of cross-
layer coupling). This phenomenon is similar to the phase-locking of
weakly coupled oscillators, in which random variations in sequence
times plays an important role in system stability [THOMPSON].

2.3. Complexity lesson from voice

In the 1970s and 1980s, the voice carriers competed by adding
features which drove substantial increases in the complexity of the
PSTN, especially in the Class 5 switching infrastructure. This
complexity was typically software-based, not hardware driven, and
therefore had cost curves worse than Moore's Law. In summary, poor
margins on voice products today are due to OPEX and CAPEX costs not
dropping as we might expect from simple hardware-bound
implementations.

2.4. Upgrade cost of complexity

   Consider the cost of providing new features in a complex network.
   The traditional voice network has little intelligence in its edge
   devices (phone instruments), and a very smart core. The Internet has
   smart edges, computers with operating systems, applications, etc.,
   and a simple core, which consists of a control plane and packet
   forwarding engines. Adding an new Internet service is just a matter
   of distributing an application to the a few consenting desktops who
   wish to use it. Compare this to adding a service to voice, where one
   has to upgrade the entire core.

3. Layering Considered Harmful

   There are several generic properties of layering, or vertical
   integration as applied to networking. In general, a layer as defined
   in our context implements one or more of

      Error Control: The layer makes the "channel" more reliable
                          (e.g., reliable transport layer)

      Flow Control: The layer avoids flooding slower peer (e.g.,
                          ATM flow control)

      Fragmentation: Dividing large data chunks into smaller
                          pieces, and subsequent reassembly (e.g., TCP
                          MSS fragmentation/reassembly)

      Multiplexing: Allow several higher level sessions share
                          single lower level "connection" (e.g., ATM PVC)

      Connection Setup: Handshaking with peer (e.g., TCP three-way
                          handshake, ATM ILMI)

      Addressing/Naming: Locating, managing identifiers associated
                          with entities (e.g., GOSSIP 2 NSAP Structure
                          [RFC1629])

   Layering of this type does have various conceptual and structuring
   advantages. However, in the data networking context structured
   layering implies that the functions of each layer are carried out
   completely before the protocol data unit is passed to the next layer.
   This means that the optimization of each layer has to be done
   separately. Such ordering constraints are in conflict with efficient
   implementation of data manipulation functions. One could accuse the
   layered model (e.g., TCP/IP and ISO OSI) of causing this conflict.
   In fact, the operations of multiplexing and segmentation both hide
   vital information that lower layers may need to optimize their

performance. For example, layer N may duplicate lower level
functionality, e.g., error recovery hop-hop versus end-to-end error
recovery. In addition, different layers may need the same
information (e.g., time stamp): layer N may need layer N-2
information (e.g., lower layer packet sizes), and the like [WAKEMAN].
A related and even more ironic statement comes from Tennenhouse's
classic paper, "Layered Multiplexing Considered Harmful"
[TENNENHOUSE]: "The ATM approach to broadband networking is presently
being pursued within the CCITT (and elsewhere) as the unifying
mechanism for the support of service integration, rate adaptation,
and jitter control within the lower layers of the network
architecture. This position paper is specifically concerned with the
jitter arising from the design of the "middle" and "upper" layers
that operate within the end systems and relays of multi-service
networks (MSNs)."

As a result of inter-layer dependencies, increased layering can
quickly lead to violation of the Simplicity Principle. Industry
experience has taught us that increased layering frequently increases
complexity and hence leads to increases in OPEX, as is predicted by
the Simplicity Principle. A corollary is stated in RFC 1925
[RFC1925], section 2(5):

    "It is always possible to agglutinate multiple separate problems
    into a single complex interdependent solution. In most cases
    this is a bad idea."

The first order conclusion then, is that horizontal (as opposed to
vertical) separation may be more cost-effective and reliable in the
long term.

3.1. Optimization Considered Harmful

A corollary of the layering arguments above is that optimization can
also be considered harmful. In particular, optimization introduces
complexity, and as well as introducing tighter coupling between
components and layers.

An important and related effect of optimization is described by the
Law of Diminishing Returns, which states that if one factor of
production is increased while the others remain constant, the overall
returns will relatively decrease after a certain point [SPILLMAN].
The implication here is that trying to squeeze out efficiency past
that point only adds complexity, and hence leads to less reliable
systems.

3.2. Feature Richness Considered Harmful

   While adding any new feature may be considered a gain (and in fact
   frequently differentiates vendors of various types of equipment), but
   there is a danger. The danger is in increased system complexity.

3.3. Evolution of Transport Efficiency for IP

   The evolution of transport infrastructures for IP offers a good
   example of how decreasing vertical integration has lead to various
   efficiencies. In particular,

    | IP over ATM over SONET -->
    | IP over SONET over WDM -->
    | IP over WDM
    |
   \|/
   Decreasing complexity, CAPEX, OPEX

   The key point here is that layers are removed resulting in CAPEX and
   OPEX efficiencies.

3.4. Convergence Layering

   Convergence is related to the layering concepts described above in
   that convergence is achieved via a "convergence layer". The end
   state of the convergence argument is the concept of Everything Over
   Some Layer (EOSL). Conduit, DWDM, fiber, ATM, MPLS, and even IP have
   all been proposed as convergence layers. It is important to note
   that since layering typically drives OPEX up, we expect convergence
   will as well. This observation is again consistent with industry
   experience.

   There are many notable examples of convergence layer failure.
   Perhaps the most germane example is IP over ATM. The immediate and
   most obvious consequence of ATM layering is the so-called cell tax:
   First, note that the complete answer on ATM efficiency is that it
   depends upon packet size distributions. Let's assume that typical
   Internet type traffic patterns, which tend to have high percentages
   of packets at 40, 44, and 552 bytes. Recent data [CAIDA] shows that
   about 95% of WAN bytes and 85% of packets are TCP. Much of this
   traffic is composed of 40/44 byte packets.

   Now, consider the case of a a DS3 backbone with PLCP turned on. Then
   the maximum cell rate is 96,000 cells/sec. If you multiply this
   value by the number of bits in the payload, you get: 96000 cells/sec
   * 48 bytes/cell * 8 = 36.864 Mbps. This, however, is unrealistic
   since it

assumes perfect payload packing. There are two other things that
contribute to the ATM overhead (cell tax): The wasted padding and the
8 byte SNAP header.

It is the SNAP header which causes most of the problems (and you
can't do anything about this), forcing most small packets to consume
two cells, with the second cell to be mostly empty padding (this
interacts really poorly with the data quoted above, e.g., that most
packets are 40-44 byte TCP Ack packets). This causes a loss of about
another 16% from the 36.8 Mbps ideal throughput.

So the total throughput ends up being (for a DS3):

```
            DS3 Line Rate:            44.736
            PLCP Overhead           - 4.032
            Per Cell Header:        - 3.840
            SNAP Header & Padding:  - 5.900
                                    =========
                                     30.960 Mbps
```

Result: With a DS3 line rate of 44.736 Mbps, the total overhead is
about 31%.

Another way to look at this is that since a large fraction of WAN
traffic is comprised of TCP ACKs, one can make a different but
related calculation. IP over ATM requires:

```
            IP data (40 bytes in this case)
            8 bytes SNAP
            8 bytes AAL5 stuff
            5 bytes for each cell
            + as much more as it takes to fill out the last cell
```

On ATM, this becomes two cells - 106 bytes to convey 40 bytes of
information. The next most common size seems to be one of several
sizes in the 504-556 byte range - 636 bytes to carry IP, TCP, and a
512 byte TCP payload - with messages larger than 1000 bytes running
third.

One would imagine that 87% payload (556 byte message size) is better
than 37% payload (TCP Ack size), but it's not the 95-98% that
customers are used to, and the predominance of TCP Acks skews the
average.

3.4.1. Note on Transport Protocol Layering

   Protocol layering models are frequently cast as "X over Y" models.
   In these cases, protocol Y carries protocol X's protocol data units
   (and possibly control data) over Y's data plane, i.e., Y is a
   "convergence layer". Examples include Frame Relay over ATM, IP over
   ATM, and IP over MPLS. While X over Y layering has met with only
   marginal success [TENNENHOUSE,WAKEMAN], there have been a few notable
   instances where efficiency can be and is gained. In particular, "X
   over Y efficiencies" can be realized when there is a kind of
   "isomorphism" between the X and Y (i.e., there is a small convergence
   layer). In these cases X's data, and possibly control traffic, are
   "encapsulated" and transported over Y. Examples include Frame Relay
   over ATM, and Frame Relay, AAL5 ATM and Ethernet over L2TPv3
   [L2TPV3]; the simplifying factors here are that there is no
   requirement that a shared clock be recovered by the communicating end
   points, and that control-plane interworking is minimized. An
   alternative is to interwork the X and Y's control and data planes;
   control-plane interworking is discussed below.

3.5. Second Order Effects

   IP over ATM provides an excellent example of unanticipated second
   order effects. In particular, Romanov and Floyd's classic study on
   TCP good-put [ROMANOV] on ATM showed that large UBR buffers (larger
   than one TCP window size) are required to achieve reasonable
   performance, that packet discard mechanisms (such as Early Packet
   Discard, or EPD) improve the effective usage of the bandwidth and
   that more elaborate service and drop strategies than FIFO+EPD, such
   as per VC queuing and accounting, might be required at the bottleneck
   to ensure both high efficiency and fairness. Though all studies
   clearly indicate that a buffer size not less than one TCP window size
   is required, the amount of extra buffer required naturally depends on
   the packet discard mechanism used and is still an open issue.

   Examples of this kind of problem with layering abound in practical
   networking. Consider, for example, the effect of IP transport's
   implicit assumptions of lower layers. In particular:

   o Packet loss: TCP assumes that packet losses are indications of
     congestion, but sometimes losses are from corruption on a wireless
     link [RFC3115].

   o Reordered packets: TCP assumes that significantly reordered
     packets are indications of congestion. This is not always the
     case [FLOYD2001].

         o Round-trip times: TCP measures round-trip times, and assumes that
           the lack of an acknowledgment within a period of time based on the
           measured round-trip time is a packet loss, and therefore an
           indication of congestion [KARN].

         o Congestion control: TCP congestion control implicitly assumes that
           all the packets in a flow are treated the same by the network, but
           this is not always the case [HANDLEY].

3.6.   Instantiating the EOSL Model with IP

   While IP is being proposed as a transport for almost everything, the
   base assumption, that Everything over IP (EOIP) will result in OPEX
   and CAPEX efficiencies, requires critical examination. In
   particular, while it is the case that many protocols can be
   efficiently transported over an IP network (specifically, those
   protocols that do not need to recover synchronization between the
   communication end points, such as Frame Relay, Ethernet, and AAL5
   ATM), the Simplicity and Layering Principles suggest that EOIP may
   not represent the most efficient convergence strategy for arbitrary
   services. Rather, a more CAPEX and OPEX efficient convergence layer
   might be much lower (again, this behavior is predicted by the
   Simplicity Principle).

   An example of where EOIP would not be the most OPEX and CAPEX
   efficient transport would be in those cases where a service or
   protocol needed SONET-like restoration times (e.g., 50ms). It is not
   hard to imagine that it would cost more to build and operate an IP
   network with this kind of restoration and convergence property (if
   that were even possible) than it would to build the SONET network in
   the first place.

4.   Avoid the Universal Interworking Function

   While there have been many implementations of Universal Interworking
   unction (UIWF), IWF approaches have been problematic at large scale.
   his concern is codified in the Principle of Minimum Intervention
   BRYANT]:

   "To minimise the scope of information, and to improve the efficiency
   of data flow through the Encapsulation Layer, the payload should,
   where possible, be transported as received without modification."

4.1. Avoid Control Plane Interworking

   This corollary is best understood in the context of the integrated
   solutions space. In this case, the architecture and design
   frequently achieves the worst of all possible worlds. This is due to
   the fact that such integrated solutions perform poorly at both ends
   of the performance/CAPEX/OPEX spectrum: the protocols with the least
   switching demand may have to bear the cost of the most expensive,
   while the protocols with the most stringent requirements often must
   make concessions to those with different requirements. Add to this
   the various control plane interworking issues and you have a large
   opportunity for failure. In summary, interworking functions should
   be restricted to data plane interworking and encapsulations, and
   these functions should be carried out at the edge of the network.

   As described above, interworking models have been successful in those
   cases where there is a kind of "isomorphism" between the layers being
   interworked. The trade-off here, frequently described as the
   "Integrated vs. Ships In the Night trade-off" has been examined at
   various times and at various protocol layers. In general, there are
   few cases in which such integrated solutions have proven efficient.
   Multi-protocol BGP [RFC2283] is a subtly different but notable
   exception. In this case, the control plane is independent of the
   format of the control data. That is, no control plane data
   conversion is required, in contrast with control plane interworking
   models such as the ATM/IP interworking envisioned by some soft-switch
   manufacturers, and the so-called "PNNI-MPLS SIN" interworking
   [ATMMPLS].

5. Packet versus Circuit Switching: Fundamental Differences

   Conventional wisdom holds that packet switching (PS) is inherently
   more efficient than circuit switching (CS), primarily because of the
   efficiencies that can be gained by statistical multiplexing and the
   fact that routing and forwarding decisions are made independently in
   a hop-by-hop fashion [[MOLINERO2002]. Further, it is widely assumed
   that IP is simpler that circuit switching, and hence should be more
   economical to deploy and manage [MCK2002]. However, if one examines
   these and related assumptions, a different picture emerges (see for
   example [ODLYZKO98]). The following sections discuss these
   assumptions.

5.1. Is PS is inherently more efficient than CS?

   It is well known that packet switches make efficient use of scarce
   bandwidth [BARAN]. This efficiency is based on the statistical
   multiplexing inherent in packet switching. However, we continue to
   be puzzled by what is generally believed to be the low utilization of

Internet backbones. The first question we might ask is what is the
current average utilization of Internet backbones, and how does that
relate to the utilization of long distance voice networks? Odlyzko
and Coffman [ODLYZKO,COFFMAN] report that the average utilization of
links in the IP networks was in the range between 3% and 20%
(corporate intranets run in the 3% range, while commercial Internet
backbones run in the 15-20% range). On the other hand, the average
utilization of long haul voice lines is about 33%. In addition, for
2002, the average utilization of optical networks (all services)
appears to be hovering at about 11%, while the historical average is
approximately 15% [ML2002]. The question then becomes why we see
such utilization levels, especially in light of the assumption that
PS is inherently more efficient than CS. The reasons cited by
Odlyzko and Coffman include:

> (i).   Internet traffic is extremely asymmetric and bursty, but
>   links are symmetric and of fixed capacity (i.e., don't know
>   the traffic matrix, or required link capacities);
>
> (ii). It is difficult to predict traffic growth on a link, so
>   operators tend to add bandwidth aggressively;
>
> (iii). Falling prices for coarser bandwidth granularity make it
>   appear more economical to add capacity in large increments.

Other static factors include protocol overhead, other kinds of
equipment granularity, restoration capacity, and provisioning lag
time all contribute to the need to "over-provision" [MC2001].

5.2. Is PS simpler than CS?

The end-to-end principle can be interpreted as stating that the
complexity of the Internet belongs at the edges. However, today's
Internet backbone routers are extremely complex. Further, this
complexity scales with line rate. Since the relative complexity of
circuit and packet switching seems to have resisted direct analysis,
we instead examine several artifacts of packet and circuit switching
as complexity metrics. Among the metrics we might look at are
software complexity, macro operation complexity, hardware complexity,
power consumption, and density. Each of these metrics is considered
below.

5.2.1. Software/Firmware Complexity

   One measure of software/firmware complexity is the number of
   instructions required to program the device. The typical software
   image for an Internet router requires between eight and ten million
   instructions (including firmware), whereas a typical transport switch
   requires on average about three million instructions [MCK2002].

   This difference in software complexity has tended to make Internet
   routers unreliable, and has notable other second order effects (e.g.,
   it may take a long time to reboot such a router). As another point
   of comparison, consider that the AT&T (Lucent) 5ESS class 5 switch,
   which has a huge number of calling features, requires only about
   twice the number of lines of code as an Internet core router [EICK].

   Finally, since routers are as much or more software than hardware
   devices, another result of the code complexity is that the cost of
   routers benefits less from Moore's Law than less software-intensive
   devices. This causes a bandwidth/device trade-off that favors
   bandwidth more than less software-intensive devices.

5.2.2. Macro Operation Complexity

   An Internet router's line card must perform many complex operations,
   including processing the packet header, longest prefix match,
   generating ICMP error messages, processing IP header options, and
   buffering the packet so that TCP congestion control will be effective
   (this typically requires a buffer of size proportional to the line
   rate times the RTT, so a buffer will hold around 250 ms of packet
   data). This doesn't include route and packet filtering, or any QoS
   or VPN filtering.

   On the other hand, a transport switch need only to map ingress time-
   slots to egress time-slots and interfaces, and therefore can be
   considerably less complex.

5.2.3. Hardware Complexity

   One measure of hardware complexity is the number of logic gates on a
   line card [MOLINERO2002]. Consider the case of a high-speed Internet
   router line card: An OC192 POS router line card contains at least 30
   million gates in ASICs, at least one CPU, 300 Mbytes of packet
   buffers, 2 Mbytes of forwarding table, and 10 Mbytes of other

state memory. On the other hand, a comparable transport switch line
card has 7.5 million logic gates, no CPU, no packet buffer, no
forwarding table, and an on-chip state memory. Rather, the line-card
of an electronic transport switch typically contains a SONET framer,
a chip to map ingress time-slots to egress time-slots, and an
interface to the switch fabric.

5.2.4. Power

Since transport switches have traditionally been built from simpler
hardware components, they also consume less power [PMC].

5.2.5. Density

The highest capacity transport switches have about four times the
capacity of an IP router [CISCO,CIENA], and sell for about one-third
as much per Gigabit/sec. Optical (OOO) technology pushes this
complexity difference further (e.g., tunable lasers, MEMs switches.
e.g., [CALIENT]), and DWDM multiplexers provide technology to build
extremely high capacity, low power transport switches.

A related metric is physical footprint. In general, by virtue of
their higher density, transport switches have a smaller "per-gigabit"
physical footprint.

5.2.6. Fixed versus variable costs

Packet switching would seem to have high variable cost, meaning that
it costs more to send the n-th piece of information using packet
switching than it might in a circuit switched network. Much of this
advantage is due to the relatively static nature of circuit
switching, e.g., circuit switching can take advantage of of pre-
scheduled arrival of information to eliminate operations to be
performed on incoming information. For example, in the circuit
switched case, there is no need to buffer incoming information,
perform loop detection, resolve next hops, modify fields in the
packet header, and the like. Finally, many circuit switched networks
combine relatively static configuration with out-of-band control
planes (e.g., SS7), which greatly simplifies data-plane switching.
The bottom line is that as data rates get large, it becomes more and
more complex to switch packets, while circuit switching scales more
or less linearly.

5.2.7. QoS

   While the components of a complete solution for Internet QoS,
   including call admission control, efficient packet classification,
   and scheduling algorithms, have been the subject of extensive
   research and standardization for more than 10 years, end-to-end
   signaled QoS for the Internet has not become a reality.
   Alternatively, QoS has been part of the circuit switched
   infrastructure almost from its inception. On the other hand, QoS is
   usually deployed to determine queuing disciplines to be used when
   there is insufficient bandwidth to support traffic. But unlike voice
   traffic, packet drop or severe delay may have a much more serious
   effect on TCP traffic due to its congestion-aware feedback loop (in
   particular, TCP backoff/slow start).

5.2.8. Flexibility

   A somewhat harder to quantify metric is the inherent flexibility of
   the Internet. While the Internet's flexibility has led to its rapid
   growth, this flexibility comes with a relatively high cost at the
   edge: the need for highly trained support personnel. A standard rule
   of thumb is that in an enterprise setting, a single support person
   suffices to provide telephone service for a group, while you need ten
   computer networking experts to serve the networking requirements of
   the same group [ODLYZKO98A]. This phenomenon is also described in
   [PERROW].

5.3. Relative Complexity

   The relative computational complexity of circuit switching as
   compared to packet switching has been difficult to describe in formal
   terms [PARK]. As such, the sections above seek to describe the
   complexity in terms of observable artifacts. With this in mind, it
   is clear that the fundamental driver producing the increased
   complexities outlined above is the hop-by-hop independence (HBHI)
   inherent in the IP architecture. This is in contrast to the end to
   end architectures such as ATM or Frame Relay.

   [WILLINGER2002] describes this phenomenon in terms of the robustness
   requirement of the original Internet design, and how this requirement
   has the driven complexity of the network. In particular, they
   describe a "complexity/robustness" spiral, in which increases in
   complexity create further and more serious sensitivities, which then
   requires additional robustness (hence the spiral).

The important lesson of this section is that the Simplicity
Principle, while applicable to circuit switching as well as packet
switching, is crucial in controlling the complexity (and hence OPEX
and CAPEX properties) of packet networks. This idea is reinforced by
the observation that while packet switching is a younger, less mature
discipline than circuit switching, the trend in packet switches is
toward more complex line cards, while the complexity of circuit
switches appears to be scaling linearly with line rates and aggregate
capacity.

5.3.1. HBHI and the OPEX Challenge

As a result of HBHI, we need to approach IP networks in a
fundamentally different way than we do circuit based networks. In
particular, the major OPEX challenge faced by the IP network is that
debugging of a large-scale IP network still requires a large degree
of expertise and understanding, again due to the hop-by-hop
independence inherent in a packet architecture (again, note that this
hop-by-hop independence is not present in virtual circuit networks
such as ATM or Frame Relay). For example, you may have to visit a
large set of your routers only to discover that the problem is
external to your own network. Further, the debugging tools used to
diagnose problems are also complex and somewhat primitive. Finally,
IP has to deal with people having problems with their DNS or their
mail or news or some new application, whereas this is usually not the
case for TDM/ATM/etc. In the case of IP, this can be eased by
improving automation (note that much of what we mention is customer
facing). In general, there are many variables external to the
network that effect OPEX.

Finally, it is important to note that the quantitative relationship
between CAPEX, OPEX, and a network's inherent complexity is not well
understood. In fact, there are no agreed upon and quantitative
metrics for describing a network's complexity, so a precise
relationship between CAPEX, OPEX, and complexity remains elusive.

6. The Myth of Over-Provisioning

As noted in [MC2001] and elsewhere, much of the complexity we observe
in today's Internet is directed at increasing bandwidth utilization.
As a result, the desire of network engineers to keep network
utilization below 50% has been termed "over-provisioning". However,
this use of the term over-provisioning is a misnomer. Rather, in
modern Internet backbones the unused capacity is actually protection
capacity. In particular, one might view this as "1:1 protection at
the IP layer". Viewed in this way, we see that an IP network
provisioned to run at 50% utilization is no more over-provisioned
than the typical SONET network. However, the important advantages

that accrue to an IP network provisioned in this way include close to
speed of light delay and close to zero packet loss [FRALEIGH]. These
benefits can been seen as a "side-effect" of 1:1 protection
provisioning.

There are also other, system-theoretic reasons for providing 1:1-like
protection provisioning. Most notable among these reasons is that
packet-switched networks with in-band control loops can become
unstable and can experience oscillations and synchronization when
congested. Complex and non-linear dynamic interaction of traffic
means that congestion in one part of the network will spread to other
parts of the network. When routing protocol packets are lost due to
congestion or route-processor overload, it causes inconsistent
routing state, and this may result in traffic loops, black holes, and
lost connectivity. Thus, while statistical multiplexing can in
theory yield higher network utilization, in practice, to maintain
consistent performance and a reasonably stable network, the dynamics
of the Internet backbones favor 1:1 provisioning and its side effects
to keep the network stable and delay low.

7. The Myth of Five Nines

Paul Baran, in his classic paper, "SOME PERSPECTIVES ON NETWORKS--
PAST, PRESENT AND FUTURE", stated that "The tradeoff curves between
cost and system reliability suggest that the most reliable systems
might be built of relatively unreliable and hence low cost elements,
if it is system reliability at the lowest overall system cost that is
at issue" [BARAN77].

Today we refer to this phenomenon as "the myth of five nines".
Specifically, so-called five nines reliability in packet network
elements is consider a myth for the following reasons: First, since
80% of unscheduled outages are caused by people or process errors
[SCOTT], there is only a 20% window in which to optimize. Thus, in
order to increase component reliability, we add complexity
(optimization frequently leads to complexity), which is the root
cause of 80% of the unplanned outages. This effectively narrows the
20% window (i.e., you increase the likelihood of people and process
failure). This phenomenon is also characterized as a
"complexity/robustness" spiral [WILLINGER2002], in which increases in
complexity create further and more serious sensitivities, which then
requires additional robustness, and so on (hence the spiral).

The conclusion, then is that while a system like the Internet can
reach five-nines-like reliability, it is undesirable (and likely
impossible) to try to make any individual component, especially the
most complex ones, reach that reliability standard.

8. Architectural Component Proportionality Law

   As noted in the previous section, the computational complexity of
   packet switched networks such as the Internet has proven difficult to
   describe in formal terms. However, an intuitive, high level
   definition of architectural complexity might be that the complexity
   of an architecture is proportional to its number of components, and
   that the probability of achieving a stable implementation of an
   architecture is inversely proportional to its number of components.
   As described above, components include discrete elements such as
   hardware elements, space and power requirements, as well as software,
   firmware, and the protocols they implement.

   Stated more abstractly:

      Let

         A   be a representation of architecture A,

         $|A|$ be number of distinct components in the service
              delivery path of architecture A,

         w   be a monotonically increasing function,

         P   be the probability of a stable implementation of an
              architecture, and let

      Then

         Complexity(A) = $O(w(|A|))$
         P(A)   = $O(1/w(|A|))$

      where

      $O(f) = \{g: N \to R \mid$ there exists c > 0 and n such that g(n)
      < c*f(n)$\}$

      [That is, O(f) comprises the set of functions g for which
      there exists a constant c and a number n, such that g(n) is
      smaller or equal to c*f(n) for all n. That is, O(f) is the
      set of all functions that do not grow faster than f,
      disregarding constant factors]

   Interestingly, the Highly Optimized Tolerance (HOT) model [HOT]
   attempts to characterize complexity in general terms (HOT is one
   recent attempt to develop a general framework for the study of
   complexity, and is a member of a family of abstractions generally
   termed "the new science of complexity" or "complex adaptive

systems"). Tolerance, in HOT semantics, means that "robustness in
complex systems is a constrained and limited quantity that must be
carefully managed and protected." One focus of the HOT model is to
characterize heavy-tailed distributions such as Complexity(A) in the
above example (other examples include forest fires, power outages,
and Internet traffic distributions). In particular, Complexity(A)
attempts to map the extreme heterogeneity of the parts of the system
(Internet), and the effect of their organization into highly
structured networks, with hierarchies and multiple scales.

8.1. Service Delivery Paths

The Architectural Component Proportionality Law (ACPL) states that
the complexity of an architecture is proportional to its number of
components.

COROLLARY: Minimize the number of components in a service delivery
path, where the service delivery path can be a protocol path, a
software path, or a physical path.

This corollary is an important consequence of the ACPL, as the path
between a customer and the desired service is particularly sensitive
to the number and complexity of elements in the path. This is due to
the fact that the complexity "smoothing" that we find at high levels
of aggregation [ZHANG] is missing as you move closer to the edge, as
well as having complex interactions with backoffice and CRM systems.
Examples of architectures that haven't found a market due to this
effect include TINA-based CRM systems, CORBA/TINA based service
architectures. The basic lesson here was that the only possibilities
for deploying these systems were "Limited scale deployments (such) as
in Starvision can avoid coping with major unproven scalability
issues", or "Otherwise need massive investments (like the carrier-
grade ORB built almost from scratch)" [TINA]. In other words, these
systems had complex service delivery paths, and were too complex to
be feasibly deployed.

9. Conclusions

This document attempts to codify long-understood Internet
architectural principles. In particular, the unifying principle
described here is best expressed by the Simplicity Principle, which
states complexity must be controlled if one hopes to efficiently
scale a complex object. The idea that simplicity itself can lead to
some form of optimality has been a common theme throughout history,
and has been stated in many other ways and along many dimensions.
For example, consider the maxim known as Occam's Razor, which was
formulated by the medieval English philosopher and Franciscan monk
William of Ockham (ca. 1285-1349), and states "Pluralitas non est

ponenda sine neccesitate" or "plurality should not be posited without
necessity." (hence Occam's Razor is sometimes called "the principle
of unnecessary plurality" and " the principle of simplicity"). A
perhaps more contemporary formulation of Occam's Razor states that
the simplest explanation for a phenomenon is the one preferred by
nature. Other formulations of the same idea can be found in the
KISS (Keep It Simple Stupid) principle and the Principle of Least
Astonishment (the assertion that the most usable system is the one
that least often leaves users astonished). [WILLINGER2002] provides
a more theoretical discussion of "robustness through simplicity", and
in discussing the PSTN, [KUHN87] states that in most systems, "a
trade-off can be made between simplicity of interactions and
looseness of coupling".

When applied to packet switched network architectures, the Simplicity
Principle has implications that some may consider heresy, e.g., that
highly converged approaches are likely to be less efficient than
"less converged" solutions. Otherwise stated, the "optimal"
convergence layer may be much lower in the protocol stack that is
conventionally believed. In addition, the analysis above leads to
several conclusions that are contrary to the conventional wisdom
surrounding packet networking. Perhaps most significant is the
belief that packet switching is simpler than circuit switching. This
belief has lead to conclusions such as "since packet is simpler than
circuit, it must cost less to operate". This study finds to the
contrary. In particular, by examining the metrics described above,
we find that packet switching is more complex than circuit switching.
Interestingly, this conclusion is borne out by the fact that
normalized OPEX for data networks is typically significantly greater
than for voice networks [ML2002].

Finally, the important conclusion of this work is that for packet
networks that are of the scale of today's Internet or larger, we must
strive for the simplest possible solutions if we hope to build cost
effective infrastructures. This idea is eloquently stated in
[DOYLE2002]: "The evolution of protocols can lead to a
robustness/complexity/fragility spiral where complexity added for
robustness also adds new fragilities, which in turn leads to new and
thus spiraling complexities". This is exactly the phenomenon that
the Simplicity Principle is designed to avoid.

10. Security Considerations

   This document does not directly effect the security of any existing
   Internet protocol. However, adherence to the Simplicity Principle
   does have a direct affect on our ability to implement secure systems.
   In particular, a system's complexity grows, it becomes more
   difficult to model and analyze, and hence it becomes more difficult

   to find and understand the security implications inherent in its
   architecture, design, and implementation.

11. Acknowledgments

   Many of the ideas for comparing the complexity of circuit switched
   and packet switched networks were inspired by conversations with Nick
   McKeown. Scott Bradner, David Banister, Steve Bellovin, Steward
   Bryant, Christophe Diot, Susan Harris, Ananth Nagarajan, Andrew
   Odlyzko, Pete and Natalie Whiting, and Lixia Zhang made many helpful
   comments on early drafts of this document.

12. References

   [AHUJA]   "The Impact of Internet Policy and Topology on
                      Delayed Routing Convergence", Labovitz, et. al.
                      Infocom, 2001.

   [ATMMPLS]   "ATM-MPLS Interworking Migration Complexities Issues
                      and Preliminary Assessment", School of
                      Interdisciplinary Computing and Engineering,
                      University of Missouri-Kansas City, April 2002

   [BARAN]   "On Distributed Communications", Paul Baran, Rand
                      Corporation Memorandum RM-3420-PR,
                      http://www.rand.org/publications/RM/RM3420", August,
                      1964.

   [BARAN77]   "SOME PERSPECTIVES ON NETWORKS--PAST, PRESENT AND
                      FUTURE", Paul Baran, Information Processing 77,
                      North-Holland Publishing Company, 1977,

   [BRYANT]   "Protocol Layering in PWE3", Bryant et al, Work in
                      Progress.

   [CAIDA]   http://www.caida.org

   [CALLIENT]   http://www.calient.net/home.html

   [CARLSON]   "Complexity and Robustness", J.M. Carlson and John
                      Doyle, Proc. Natl. Acad. Sci. USA, Vol. 99, Suppl. 1,
                      2538-2545, February 19, 2002.
                      http://www.pnas.org/cgi/doi/10.1073/pnas.012582499

   [CIENA]   "CIENA Multiwave CoreDiretor",
                      http://www.ciena.com/downloads/products/
                      coredirector.pdf

   [CISCO]   http://www.cisco.com

   [CLARK]   "The Design Philosophy of the DARPA Internet
                    Protocols", D. Clark, Proc. of the ACM SIGCOMM, 1988.

   [COFFMAN]   "Internet Growth: Is there a 'Moores Law' for Data
                    Traffic", K.G. Coffman and A.M. Odlyzko, pp. 47-93,
                    Handbook of Massive Data Stes, J. Elli, P. M.
                    Pardalos, and M. G. C. Resende, Editors. Kluwer,
                    2002.

   [DOYLE2002]   "Robustness and the Internet: Theoretical
                    Foundations", John C. Doyle, et. al. Work in
                    Progress.

   [EICK]   "Visualizing Software Changes", S.G. Eick, et al,
                    National Institute of Statistical Sciences, Technical
                    Report 113, December 2000.

   [MOLINERO2002]   "TCP Switching: Exposing Circuits to IP", Pablo
                    Molinero-Fernandez and Nick McKeown, IEEE January,
                    2002.

   [FLOYD]   "The Synchronization of Periodic Routing Messages",
                    Sally Floyd and Van Jacobson, IEEE ACM Transactions
                    on Networking, 1994.

   [FLOYD2001]   "A Report on Some Recent Developments in TCP
                    Congestion Control, IEEE Communications Magazine, S.
                    Floyd, April 2001.

   [FRALEIGH]   "Provisioning IP Backbone Networks to Support Delay-
                    Based Service Level Agreements", Chuck Fraleigh,
                    Fouad Tobagi, and Christophe Diot, 2002.

   [GRIFFIN]   "What is the Sound of One Route Flapping", Timothy G.
                    Griffin, IPAM Workshop on Large-Scale Communication
                    Networks: Topology, Routing, Traffic, and Control,
                    March, 2002.

   [HANDLEY]   "On Inter-layer Assumptions (A view from the
                    Transport Area), slides from a presentation at the
                    IAB workshop on Wireless Internetworking", M.
                    Handley, March 2000.

   [HOT]   J.M. Carlson and John Doyle, Phys. Rev. E 60, 1412-
                    1427, 1999.

   [ISO10589]   "Intermediate System to Intermediate System
                      Intradomain Routing Exchange Protocol (IS-IS)".

   [JACOBSON]   "Congestion Avoidance and Control", Van Jacobson,
                      Proceedings of ACM Sigcomm 1988, pp. 273-288.

   [KARN]   "TCP vs Link Layer Retransmission" in P. Karn et al.,
                      Advice for Internet Subnetwork Designers, Work in
                      Progress.

   [KUHN87]   "Sources of Failure in the Public Switched Telephone
                      Network", D. Richard Kuhn, EEE Computer, Vol. 30, No.
                      4, April, 1997.

   [L2TPV3]   Lan, J., et. al., "Layer Two Tunneling Protocol
                      (Version 3) -- L2TPv3", Work in Progress.

   [MC2001]   "U.S Communications Infrastructure at A Crossroads:
                      Opportunities Amid the Gloom", McKinsey&Company for
                      Goldman-Sachs, August 2001.

   [MCK2002]   Nick McKeown, personal communication, April, 2002.

   [ML2002]   "Optical Systems", Merril Lynch Technical Report,
                      April, 2002.

   [NAVE]   "The influence of mode coupling on the non-linear
                      evolution of tearing modes", M.F.F. Nave, et al, Eur.
                      Phys. J. D 8, 287-297.

   [NEUMANN]   "Cause of AT&T network failure", Peter G. Neumann,
                      http://catless.ncl.ac.uk/Risks/9.62.html#subj2

   [ODLYZKO]   "Data networks are mostly empty for good reason",
                      A.M. Odlyzko, IT Professional 1 (no. 2), pp. 67-69,
                      Mar/Apr 1999.

   [ODLYZKO98A]   "Smart and stupid networks: Why the Internet is like
                      Microsoft". A. M. Odlyzko, ACM Networker, 2(5),
                      December, 1998.

   [ODLYZKO98]   "The economics of the Internet: Utility, utilization,
                      pricing, and Quality of Service", A.M. Odlyzko, July,
                      1998.
                      http://www.dtc.umn.edu/~odlyzko/doc/networks.html

   [PARK]   "The Internet as a Complex System: Scaling,
                       Complexity and Control", Kihong Park and Walter
                       Willinger, AT&T Research, 2002.

   [PERROW]   "Normal Accidents: Living with High Risk
                       Technologies", Basic Books, C. Perrow, New York,
                       1984.

   [PMC]   "The Design of a 10 Gigabit Core Router
                       Architecture", PMC-Sierra, http://www.pmc-
                       sierra.com/products/diagrams/CoreRouter_lg.html

   [RFC1629]   Colella, R., Callon, R., Gardner, E. and Y. Rekhter,
                       "Guidelines for OSI NSAP Allocation in the Internet",
                       RFC 1629, May 1994.

   [RFC1925]   Callon, R., "The Twelve Networking Truths", RFC 1925,
                       1 April 1996.

   [RFC1958]   Carpenter, B., Ed., "Architectural principles of the
                       Internet", RFC 1958, June 1996.

   [RFC2283]   Bates, T., Chandra, R., Katz, D. and Y. Rekhter,
                       "Multiprotocol Extensions for BGP4", RFC 2283,
                       February 1998.

   [RFC3155]   Dawkins, S., Montenegro, G., Kojo, M. and N. Vaidya,
                       "End-to-end Performance Implications of Links with
                       Errors", BCP 50, RFC 3155, May 2001.

   [ROMANOV]   "Dynamics of TCP over ATM Networks", A. Romanov, S.
                       Floyd, IEEE JSAC, vol. 13, No 4, pp.633-641, May
                       1995.

   [SALTZER]   "End-To-End Arguments in System Design", J.H.
                       Saltzer, D.P. Reed, and D.D. Clark, ACM TOCS, Vol 2,
                       Number 4, November 1984, pp 277-288.

   [SCOTT]   "Making Smart Investments to Reduce Unplanned
                       Downtime", D. Scott, Tactical Guidelines, TG-07-4033,
                       Gartner Group Research Note, March 1999.

   [SPILLMAN]   "The Law of Diminishing Returns:, W. J. Spillman and
                       E. Lang, 1924.

   [STALLINGS]   "Data and Computer Communications (2nd Ed)", William
                       Stallings, Maxwell Macmillan, 1989.

   [TENNENHOUSE]   "Layered multiplexing considered harmful", D.
                        Tennenhouse, Proceedings of the IFIP Workshop on
                        Protocols for High-Speed Networks, Rudin ed., North
                        Holland Publishers, May 1989.

   [THOMPSON]   "Nonlinear Dynamics and Chaos". J.M.T. Thompson and
                        H.B. Stewart, John Wiley and Sons, 1994, ISBN
                        0471909602.

   [TINA]   "What is TINA and is it useful for the TelCos?",
                        Paolo Coppo, Carlo A. Licciardi, CSELT, EURESCOM
                        Participants in P847 (FT, IT, NT, TI)

   [WAKEMAN]   "Layering considered harmful", Ian Wakeman, Jon
                        Crowcroft, Zheng Wang, and Dejan Sirovica, IEEE
                        Network, January 1992, p. 7-16.

   [WARD]   "Custom fluorescent-nucleotide synthesis as an
                        alternative method for nucleic acid labeling",
                        Octavian Henegariu*, Patricia Bray-Ward and David C.
                        Ward, Nature Biotech 18:345-348 (2000).

   [WILLINGER2002]   "Robustness and the Internet: Design and evolution",
                        Walter Willinger and John Doyle, 2002.

   [ZHANG]   "Impact of Aggregation on Scaling Behavior of
                        Internet Backbone Traffic", Sprint ATL Technical
                        Report TR02-ATL-020157 Zhi-Li Zhang, Vinay Ribeiroj,
                        Sue Moon, Christophe Diot, February, 2002.

13. Authors' Addresses

   Randy Bush
   EMail: randy@psg.com

   David Meyer
   EMail: dmm@maoz.com

14. Full Copyright Statement

Acknowledgement