A Multi-Theory Analysis of Long-lived Networks

David Clark[1]

Version 2.1, April 25, 2009

Introduction—the goal of longevity

In comparison to many artifacts of computing, the Internet has lived to an old age—it is over 35 years old. Opinions differ as to the extent that it is showing its age, and among some researchers, there is a hypothesis that the Internet of 15 years from now might be built on different principles. Whether the network of 15 years from now is a minor evolution from today's network, or a more radical alternative, it should be a first-order requirement that this future Internet be designed so that it also can survive the test of time.

I have used the terms 'longevity", or "long-lived", to describe this objective. The objective is easy to understand, but the principles that one would use to achieve it are less well understood. In fact, there are a number of different theories about how to design a network (or other system) that survives for a long time. This paper takes the point of view that many of these theories are relevant, and that one can achieve a long-lived network in different ways, by exploiting various combinations of these theories in different degree. While some theories are incompatible, many are consistent with one another.

The approach I take here is inspired by the book *Theories of Communication Networks*, by Peter Monge and Noshir Contractor (Belady and Lehman 1976). The topic of that book is not networks make of routers, but social networks made out of people and their relationships. They identify many theories that have been put forward to explain the formation and durability of social networks, and their thesis is that many of these theories are valid, to different degrees, in different such networks. So it is necessary to have a multitheoretical, multilevel framework in order to explain the character of any given social networks. Since there are many examples of social networks in the real world, one can do empirical research to try to determine how (for example), theories of self-interest, collective action, knowledge exchange, homophily and proximity shape a given network. While we have fewer networks as examples then these authors do, we can still attempt to catalog the theories that have been put forward to explain why a network might or might not be long-lived.

Classes of theories

With some degree of over-simplification, many of the theories of longevity can be classified into three subclasses, as follows:

---

[1] The research reported here and the preparation of this document was supported by the Office of Naval Research under contract N00014-08-1-0898, and by the National Science Foundation under agreement 0836555. The opinions contained are those of the author, and do not reflect the opinions of the supporting agencies.

Theories of change: These theories presume that over time, requirements will change, so a long-lived network must of necessity change. Theories of this sort sometimes use the word "evolvability" rather than "longevity" to describe the desired objective, since they assume that a network that cannot change to meet changing requirements will soon cease to be useful. The word "change" as used here, usually has the implication of *uncertain* change; if the future trajectory of the requirements on a system could be completely characterized, one could presumably fold these into the initial design process, if the cost were not prohibitive.

Theories of stability: in contrast to theories of change, theories of stability presume that a system remains useful over time by providing a stable platform on which other services can depend.

Theories of innovation: These theories assume that change is *beneficial*, not just (or rather than) inevitable. These theories stress the importance of change and innovation as economic drivers.

These classes of theories are not incompatible. Theories of innovation are often theories of stability, in that the stability of the network as a *platform* allows innovation on top of that platform by what innovation theory would call *complementors*. Taking an example from operating systems, it is the stability of the interfaces to the operating system that invites application designers to take the risk of developing and marketing new applications for that system.

Architecture and longevity

The term "architecture" is often used to describe the basic design concepts that underlie a system like a network: the top-level modularity, interfaces and dependencies, the assumptions that all parties must take as globally consistent, and so on. Again, with respect to architecture and change, there are two subclasses of theories:

Stable architecture that supports change: in this view, the architecture embodies those aspects of the system that do not change. It is the stability of the architecture that permits the overall evolution of the system.

Evolving architecture: in this view, the architecture itself can (and does) evolve to address changing needs. If the architecture cannot adequately evolve, this leads to *violations* of the architecture, which (according to these theories) leads to a gradual loss of function, and an increasing difficulty of further change, an *ossification* of the system that gradually erodes its utility.

The theory of ossification

The theory of ossification was perhaps first put forward with respect to operating systems by (Belady and Lehman 1976). They pose their First Law of Program Evolution Dynamics, the *Law of continuing change,* which states that a system that is used undergoes continuing change until it is judged more cost effective to freeze and recreate it. According to this point of view, systems lose the ability to evolve over time, and eventually have to be redone from scratch in order to allow continued change. So this theory is a theory of change, but an episodic theory, which predicts that systems (or

architectures) have a natural lifetime, and need to be renewed from time to time by a more revolutionary phase.

New theories of design suggest that it may be possible to derive an architecture from a set of requirements by a rigorous and formal process. It is an open question how such an architecture will deal with change. If one changes the requirements and then derives a new architecture, the differences may be pervasive—essentially a new design rather than a modification of the old one. But if one takes an architecture derived in this way and modifies it after the fact, all of the theory that applied to the original design process no longer applies. This sort of action is like taking the output of a compiler and patching the machine code. It is thus possible that architectures that have been algorithmically derived from requirements will be brittle with respect to change, or (in term of these theories) easily ossified.

The theory of utility

All discussion of longevity must occur in the context of a network that is used. A network that is long-lived is a network that continues to be used over time. So it is a precondition of a long-lived network that it be useful in the first place. So any theory of longevity must have inside it some theory of utility, which explains why the network is useful. The first theory of longevity I identify is based on a specific theory of utility.

The theory of the general network

According to this theory, a fully general system, which could meet all needs, would not need to evolve, and would thus be long-lived. The theory of the general network is thus a theory of stability.

The challenge, of course, is to define exactly what a general network might be.

The theory of the ideal network and impairments: according to this theory, networks provide a very simple service that can be described in its ideal (if unrealizable) form. One statement is as follows:

> *An ideal data network will reliably deliver any amount of data to (and only to) any set of intended and willing recipients in zero time for zero cost and zero consumption of energy.*

Of course, such a network cannot be realized. Some limits, such as the speed of light, are physical limits that cannot be violated. Others, such as cost, seem to improve over time as a consequence of innovation. Taken together, these limits, sometimes called *impairments*, define how far any practical network diverges from the ideal. In this theory, a maximally general network minimizes the various impairments, and to the extent possible, allows each set of users to trade off among the impairments to the maximum extent possible. Thus, *queuing theory* seems to capture a fundamental set of tradeoffs among speed, cost (utilization) and delay. A network that (for a given class of traffic) does as well as queuing theory would predict, and allows the users to move along the performance frontier defined by queuing theory, would be seen as a maximally general network.

According to this theory, if a network is maximally general with respect to the fundamental impairments (a theory of stability) and is open to change with respects to impairments that change over time (a theory of innovation), then such a network will be long-lived.

The Internet as a practical example

Many have seen the Internet as a good, if pragmatic example of a general network, and see its longevity as a consequence of that fact. The use of packets as a multiplexing mechanism has proved to be a very general and flexible mechanism. Packets support a wide range of applications, and allow for the introduction of new technology as it evolves. [More examples and elaboration?]

The theory of real options

Real option theory captures the idea (in common with options as a financial instrument) that one can attempt to quantify the cost-benefit of investing now to "keep options open" or in other words to deal with uncertainty. It is thus a theory of change, to the extent that change equates to uncertainty. It is also a theory of the general network, but in economic terms, in that it suggests that one can spend money now to purchase flexibility later to respond to uncertain change. It does not describe what the resulting general network is (in contrast to the offered definition above), but just postulates that generally is often to be had, but at a price.

Real option theory is perhaps more often applied to the construction of a network (e.g. how much spare capacity to purchase now) than to the architecting of a network. But the theory none the less reminds us that generality may come at a price, and that price is one of the impairments to the definition of the ideal network postulated above.

The theory of tussle and points of control

The discussion of the ideal network does not fully capture what happens inside networks, because the ideal is stated from the perspective of only one class of actors—the parties desiring to communicate. The statement of the ideal does not afford any attention to other actors, such as governments that want to carry out lawful intercept on traffic, to employers and others who want to limit what can be carried over their networks, and so on. The list of stake-holders that can be identified in the current Internet is substantial, and each of these stake-holders tries to put forward their interests, perhaps at the expense of other stake-holders.

This ongoing process has been called *tussle* (Clark, Wroclawski et al. 2005), and seems to be a fundamental aspect of any system (like the Internet) that is deeply embedded in the larger social, economic and regulatory context. According to the theory of tussle, systems will prove to be long-lived if they are designed to minimize the disruptive consequence of tussles, and in particular so that tussle does not lead to violations of the architecture of the network.

Various aphorisms have been used to describe how a system should be designed to tolerate tussle.

- The tree that bends in the wind does not break.
- You are not designing the outcome, but the playing field.

The idea behind these aphorisms is to design your systems so that they do not attempt to resist tussle and impose a fixed outcome, but to be flexible in the face of the inevitable. However, they give little practical guidance as to how one might do it, other to hint that one can tilt the playing field to bias the resulting tussle consistent with the values of the designer.

Tussle isolation: One design principle that emerges from the consideration of tussle is a new modularity principle called *tussle isolation*. Computer science has a number of theories of modularity, such as layering (e.g. the avoidance of mutual dependency). The idea behind tussle isolation is that if the designer can identify in advance an area where there is likely to be persistent tussle, then the design should isolate that area so that the resulting tussle does not "spill over" into other aspects of the network.

- DNS: if the early designers had understood that the DNS would include names over which there would be trademark disputes, that use of names could have been made a separate service, so that the scope of the trademark disputes could be minimized.
- [More]

Placement of interfaces: In addition to isolating tussle, one can "move it around" by the placement of critical interfaces within a system—another example of a non-technical principle for modularizing a system. The economist Ronald Coase received a Nobel Prize for his *theory of the firm*, which (very roughly) states that firms will contract for a service or function if the *transaction cost* is low, where transaction cost covers not the cost of the service itself, but the search cost, the bargaining cost, the costs that arise due to lack of accurate information about the other firms, etc. If transaction costs are high, the firm may incur a lower total cost by realizing the service or function internal to the firm. So inter-firm interaction (e.g. competition, bargaining, etc.) will be found where there are low transaction costs. In systems of the sort we are describing here, transaction costs will be lower at points in the architecture where there are well-defined interfaces. If an interface between modules is well-defined and easy to understand, then exploiting this interface will have a sufficiently low transaction cost that it can form the basis for competitive interaction among firms. If, on the other hand, there is no clear interface at a particular point, it is hard to "open up" that point to inter-firm action, and that point will normally remain internal to the firm. So tussle will often occur around critical interfaces. By moving the interfaces, one can "move" the tussle.

Removal of interfaces: a sub-class of the above theory is the idea that by intentionally removing interfaces and making the system less modular and more integrated, one can increase the power of the firm that owns the system, and limit competitive entry as well as other forms of tussle. This theory is an example of a theory of stability (as well as a theory of market power and hegemony—see below.)

Asymmetric struggle: Many tussles are defined by the fact that different stake-holders have access to different sorts of tools and methods. Network architects define

module interfaces and shift function around, governments pass laws and regulations, network operators make investments and configure the physical network. Each of these actions can advantage the particular stake-holder and disadvantage the adverse interests. Given this fact, it is worth some study (a full exploration is beyond the scope of this paper) as to how these various methods interact with each other. Like a game of "rock, paper, scissors", they sometimes seem to circle around each other in endless cycles. One sub-theory that characterizes some of the interactions is the **theory of the blunt instrument**, the idea that while each stake-holder has distinct powers, the design of one part of the system can blunt the tools of control that the others have, and thus render them less effective. Thus, for example, the use of encryption as part of the network design greatly limits the ability of other actors to observe (and thus to impose limits on) what the users are doing. In the extreme, the network operator is reduced to carrying all traffic, blocking all encrypted traffic, or refusing to serve the relevant customer—an example of blunting the network operator's instrument of control.

Tussle and longevity

The theory of tussle might be seen as the theory of change, but in fact it may be closer to a theory of dynamic stability. Stability need not imply a fixed system, but can also imply a system that has checks and balances, or feedback, to home it to a stable point. Tussle can be viewed as such a mechanisms—a set of forces that tend to bring a system back to a stable compromise point if some new input (e.g. a technical innovation) shifts it away from that point. Over time, the compromise point may shift (as social norms shift over time) and the stable point may be different in different cultures. So tussle can be seen as a dynamic and ongoing mechanism that contributes to system longevity, and further as a mechanism that allows the outcome to be different in different cultures, as opposed to a rigid system that attempts to impose global agreement in contexts where global agreement is not feasible. This variation in outcome, as well, is a contributor to longevity.

The theory of building blocks and composable elements.

The theory of the general network assumed that one could describe what an ideal, or fully general network would do. It was based on the concept of a network as a system with a very simple core function. Another point of view is that a network should be capable of offering a much richer set of services (perhaps not all at the same layer). The measure of a network would not be how well it does at limiting the impact of impairments, but how easy it is to incorporate new sorts of services between the communicating parties. In this point of view, if the network is built only of fixed-function routers, that is a limiting rather than a stabilizing outcome.

This point of view becomes more prevalent if one looks not just at the simple, packet-forwarding layer, but at services "above" that layer, which might do things such as convert information formats, validate identity, provide various sorts of security services and the like. In this layered view, one would then ask of the packet layer whether it was optimally suited to support the deployment and configuration of these higher-level services. For example, to insure the proper operation of security services, it

might be important to make sure that the packets cannot bypass the services as they are being forwarded. So the desire to deploy these higher layer services may change and expand the requirements at the packet level, even if these services are seen as "higher layer" services.

The term "building blocks" is sometimes used to describe these sorts of service elements, and the generality of the network is then related to the generality of these building blocks, and the degree of control over how these service building blocks can be composed. If the building blocks are implemented as programs running on general-purpose processing nodes, there are few limits to the generality of the functions that can be programmed, so the limits to the expressive power will be found in the sequencing of these service elements, the ways that this sequencing can be expressed, and the information that is made available to these elements.

(Expressive power is related to the way that the service element gains access to the information being transmitted. If the service elements operate on individual packets, then the breaking the information into packets may limit the information that the elements can see. If the element re-assembles the packets into a larger data unit, the ability of the service element to see the larger data unit may offer more generality. If parts of the information are encrypted, this would imply a deliberate narrowing of what the service element can see and act on, which illustrates the use of design to influence tussle.)

The theory of expressive power

There seem to be two, perhaps contradictory, theories of expressive power—the maximal and the minimal theory.

In the maximal theory, the theory of expressive power states that a network will be long-lived if it has rich expressive power, so that new service elements can be introduced and invoked. At the packet level, expressive power would be increased by adding more powerful addressing modes (such as source addressing, which could route a packet through a series of service elements) and additional fields in the packet header that could be used to convey additional information to the service elements. If the service elements act on larger data units that are assembled out of packets at the point where the element is located, this sort of expressive power can be captured in data at the application layer. (Mail Transfer Agents are an example of higher-level, or application-level service elements. They act on and modify the header of the mail message, and schemes for mail encryption are defined to encrypt the body but leave the header visible so the mail system can function properly.)

The opposite, or minimal, theory about service elements and expressive power arises within the theory of tussle. In this point of view, any service element will be a point of contention and tussle, as different stake-holders try to control the service being provided. Thus, ISPs sometimes block access to third-party mail transfer agents in an attempt to force a customer to use their mail service; by doing so the ISP may be able to impose limitations on what the customer can do (for example what are acceptable email names). This theory would suggest that a network design might deliberately limit the

expressive power of the design (perhaps at certain of the layers in the design), to limit the points of tussle in the network, and thus bring about longevity through stability.

The theory of programmable elements (active networks)

The theory that building blocks bring beneficial flexibility has an aggressive version in which elements within the network can be programmed dynamically, perhaps even by means of programs carried within the data packets themselves. This point of view, sometimes called Active Networks, can be argued as reducing tussle rather than facilitating it, since it tilts the playing field toward the end-user, and blunts the instruments of control that belong to the stake-holders "in" the network. The programs come from the edge, selected and installed by the end-user or his agents; the stakeholders who are in the network only provide the platform for these programs. They cannot easily regulate what those programs do, except by attempts to impose limits on how they are composed. With no ability to see what the programs do, and only a "blunt instrument" capability to limit how they are composed, the result (according to this point of view) is a stable platform (see below) on which innovation can be driven from the edge.

The theory of programmable elements is a sub-case, of course, of more general **theories of in-network elements**: elements that are placed at strategic points in the network but that are not particularly programmable or flexible. Today, we see firewalls, VPN end-points and the like, as well as higher-level elements such as web caches scattered about the network. In most points of view, these are add-ons, or after-thoughts, at best harmless to the basic architectural principles and at worst violations. They are not generally viewed in positive terms as contributing to longevity. This point of view might be challenged (after all, routers are not dynamically programmable but form the foundation of the Internet as a stable platform and a general network). The suspicion of these elements is that they seem hard to deploy (not useful as a rapid response to change), and (based on insights so far) no elements other than packet forwarders have been identified that contribute to generality. So fixed (or slowly changing) elements in the network do not seem to align either with a theory of change or a theory of a general network. They may be viewed as barriers to longevity, using the theory of ossification.

The theory of the stable platform

The theory of the stable platform is a theory understood by those who study innovation. According to this theory, innovation (which represents a valuable form of change) is facilitated by a stable platform with an unchanging interface and service definition. In the language of this theory, those who innovate "on top of" the platform are called *complementors.* If the platform itself is unstable and subject to change and innovation, this increases the cost of building complementary systems (e.g. "applications") that exploit the platform (as the application must be upgraded to keep pace with the platform changes) and increases the risk (due to uncertainty about changes in the platform that might reduce the functionality of the application). This theory is an example of a theory of innovation that is a theory of stability.

The theory of the stable platform can be stated in dynamic form: to the extent that there are a number of complementors, they will use their power to argue for the stability

of the platform, which will induce more complementors to join, and so on, in a positive feedback situation. The reverse of this dynamic is also a part of the theory; if a platform is not useful, it makes no difference if it is stable.

Again, the Internet has been seen as a good illustration of a stable platform (the service of the IP layer) that permits innovation "on top of" that platform. The theory of the stable platform has been used to explain the longevity of the current Internet.

The theory of modularity and dependency

The theory of the stable platform is a specialization of a very general and well-understood principle of Computer Science: the use of layering to define a unidirectional dependency between two modules, the "lower" and the "upper" layer. The upper layer depends on the lower layer, but not the other way around. Much has been written about layering, and the related concept of abstraction. However, this more general form of the theory seems to have less to say about longevity than the more specific theory of the stable platform.

The theory of semantics-free service

The theory of the stable platform does not say anything about what function the platform should implement in order to be useful and general. The theory of the general network provides one answer to that question: the platform should provide a general service that is as close to the ideal (the minimum set of impairments) as can be fashioned.

The version of the theory of the general network offered above was that the network should just deliver bytes. In contrast to the theory of composable building blocks, the network should not have any model of what those bytes represent, or what the high-level objective of the application is. This version of the theory has sometimes been called the **semantics-free** network, or the **transparent** network, or (in more colloquial terms), "what comes out is what goes in". The assumption is that if the network begins to incorporate a model of what one or another application is trying to do, it will end up being specialized to those applications, at the cost of generality.

It has been argued that the longevity of the Internet is due to its semantic-free design, and the refusal of its designers to allow the protocols to be optimized to the popular application of the day. It could be argued that semantics-free service is an example of the theory of utility, but it is not clear what line of reasoning would be used to make this point in advance. However, the theory of the general network may imply the theory of semantics-free service, since (as it was stated earlier) the general network was defined as "delivering data", which seems to imply a semantics-free service.

This theory is a close relative to the **end-to-end argument**, but in the beginning that argument was about correct operation, not about generality. The interpretation of the end-to-end argument as an argument for generality can be found implicitly in the original paper (Saltzer, Reed et al. 1984), but has become more elaborated and explicit in some of the subsequent writings about the argument.

The theories of global agreement

One point of view about architecture is that it defines those aspects of the system about which there must be global agreement: architecture defines those parts of the system that "work the same way everywhere". In this context, there are actually two theories about global agreement and longevity: the minimal theory and the maximal theory.

The theory of maximal global agreement: This theory postulates that the more aspects of the system are well-defined, the more stable the platform. By providing a well-specified functional specification for the platform, the difficulty and risk to the complementor is minimized. The word "maximal" is probably an overstatement of this theory—the more careful statement would be that "up to a point", increased specification and careful definition is a good thing.

The theory of minimal global agreement: This theory is a theory of change. It states that the fewer things we all have to agree to in common, the more we will be able to accommodate a range of uses with different needs. As long as the platform remains useful, having fewer points of global agreement is beneficial, and will allow the network to evolve over time without disrupting the utility of the platform. So in contrast to the maximal or "up to a point" theory, this is a "down to a point" theory, or perhaps (to paraphrase the quote of Einstein): "Everything should be made as minimal as possible, but no less."

False agreement: Whichever version of the theory is put forward, there remains the question of when a global agreement is really an agreement, and when it is the illusion of agreement. An example from the Internet might be the initial assumption that the Internet was based on the global agreement that there was a single global address space. It was thought that this agreement was important, and one of the basic tenets of the stable IP platform, but then Network Address Translation devices were introduced, and the Internet survived. Some would say that because NAT devices impair certain classes of applications (in particular, passive servers located behind NAT devices), we should view NATs (and the loss of global addresses) as a significant violation of the stable architecture. However, if the Internet was equipped with a protocol that allowed state to be installed dynamically in NAT devices (perhaps an example of the theory of the building block), the Internet could support essentially all the applications it did in the era of global addresses.

However the reader might choose to analyze this example, the more general question is how one would test a proposed point of global agreement to see whether agreement is actually required about the point in order to have a stable platform. Clever reconceptualization may allow what was seen as a global agreement to be set aside with no loss of power.

One might pose an informal "test of time" theory—that a presumed point of global agreement should only be judged in hindsight based on whether people actually depend on it. But this seems like a poor candidate for a design principle. On the other hand, it seems difficult to take the position that we can *force* dependency to force stability. The theory of utility suggests that if a feature is not useful, it does not matter if it is stable, or if it is a point of nominal global agreement.

The theory of technology independence

The theory of technology independence is another theory of stability in the face of change. This theory states that a system will be long-lived if it allows new generations of technology to be incorporated into the system without disrupting the stable platform that the system provides to the complementors. Since technology evolved rapidly in the CS world, it would seem that any long-lived system must be designed so that it is not rendered obsolete by new technology.

Again, this theory can be used to explain the longevity of the Internet. The simple, packet-based platform of the Internet can be implemented on top of all sorts of communication technology. The Internet has accommodated circuits that have increased in speed by at least 10^6 during its lifetime. It has accommodated multi-access local area networks, wireless networks, and the like. The applications running on top of the IP interface are largely unaffected by these innovations.

The theory of the hourglass

The combination of the theory of the stable platform and the theory of technology independence lead to a theory (or a picture) that is a hourglass: a picture of a narrow waist representing the common point of agreement (a global agreement?) on the IP layer, with great diversity in technology below and great diversity in application above.

Once the image of the hourglass was identified and associated with a theory of longevity, further study revealed that the Internet had many hourglasses in it: the reliable byte-stream on which email sits (the Internet standards for email work quite well on transport protocols other than TCP), HTTP, and so on. [other useful examples?]

The theory of cross-layer optimization

The theory of cross-layer optimization is a contrarian theory, contrary to the theory of the hourglass. The theory of cross-layer optimization states that over the long run, the evolution of technology will be so substantial that a stable, technology-independent platform will become limiting, and eventually, uncompetitive compared to an approach that allows the application and the technology to adapt to each other. The application designer will have a harder task than with a stable platform, but in exchange for doing the additional design work so that the application can adapt to different technologies, the designer will achieve greatly improved performance and function.

The theory of cross-layer optimization has been put forward in the past in the context of various emerging technologies, perhaps starting with multi-access local area networks. In the past, the theory of the stable platform has dominated. Today, cross-layer optimization is being put forward in the context of some wireless networks, especially wireless designed for very challenging circumstances, such as battlefield networks. It is not clear whether longevity is the primary requirement for such networks.

The theory of downloadable code

The theory of downloadable code is a theory of change, or perhaps of innovation. This theory states that the need for global agreement can be minimized by the approach

of downloading code into the communicating elements, so that the agreement is achieved not by the mandate of standards but by an agreement to run compatible software.

If the code were downloaded into the network elements that forward packets, this would be the same as the theory of active networks. This theory has not achieve much traction in the real world. However, code that is downloaded into the end-node (most commonly at the application layer, or as a supporting service to applications) has been a very powerful tool to support innovation. New formats for audio and images (still, animated and video) are introduced by allowing end-nodes to download new rendering code. Standards such as PDF, Flash, and [??] enter the market by means of free viewer software provided by the creator of the standard. Pragmatically, once a format can be implemented in downloadable software (as opposed to hardware, for example), proliferation of competing standards does not seem to be an impediment to progress and longevity.

The theory of downloadable code is an example of a theory of the stable platform: in this case the platform is a software platform, not a network service platform (such as the IP layer). The browser of today, with its "plug-in" architecture, becomes a stable platform on which innovation (e.g. new downloadable modules) can be built.

This observation begs the question of what parts of the network could be based on downloadable code, rather than on global agreement. Today, for example, transport protocols such as TCP are more or less a global agreement. Alternatives cannot be downloaded, because the code that implements TCP is embedded in the kernel of most operating systems for a number of reasons: performance, dealing with interrupts and timers, multi-threading, efficient demultiplexing and buffer management, and the like. However, is this a fundamental consequence of some aspect of transport protocols, or just a historical accident? It might be possible to design a framework (or platform, to use the earlier word) into which different protocols at this level could be downloaded, just as the web browser provides a framework for downloadable code at a higher level. Were this framework demonstrated, one could argue that the theory of downloadable code would be a better path to longevity that the theory of global agreement, even at the transport layer of the protocol stack.

Change: hard or easy?

More abstractly, the theory of downloadable code challenges us to take a rigorous look at what makes change hard or easy. The need for global agreement seems to make change hard (if everyone had to change at once).

Version numbers are sometimes put forward as a technique to manage change. Version numbers in protocols can allow two incompatible designs to co-exist, either transiently during a time of change, or (more realistically) forever. Version numbers work so long as it is possible to verify that all the components that will be involved in some operation support at least one version in common.

Changes to production code are often viewed as very hard to make, or at least not quick to make. Vendors need to be convinced of the need for change, and then the change must be scheduled into the development cycle. Especially if the change is based on a standard that requires broad agreement, such changes can take year. However, one

should not mistake the time it takes to make a change with fundamental difficulty. What makes a change easy or hard to implement is more its interaction with other parts of the system (which, according to the theory of ossification, will increase over time).

On the other hand, when the change is more a bug-fix and the need is urgent (as with the discovery of a security vulnerability) changes can be made in a matter of days or weeks, and the current trend to automate the downloading of new versions (e.g. of operating system and major software packages such as Office) can allow substantial deployment of updates in days.

Overall, there is a trend in the current Internet (and the systems attached to it, such as operating systems) to make change (updates, patches, releases of new versions) easier to accomplish. This trend begs the question of which changes are actually hard to make, and why. The theory of minimal global agreement would suggest that if the right tools are put in place to allow software to be upgraded, there is little that cannot be changed in principle, and more and more that can be changed in practice. With the trend of moving function from hardware to software (e.g. software-defined radios) functions that had traditionally been viewed as fixed and static have turned out to be very amenable to change, and not fundamental at all.

The theory of hegemony

The theory of hegemony is a theory of stability. It postulates that a system will be long-lived if a single actor is in charge of the system, an actor that can balance change against stability, and balance the needs of the various stake-holders in an orderly way. By taking tussle out of the technical domain and into the planning or administrative (regulatory) context of the controlling actor, the platform becomes more predictable and thus more appealing as a platform. So the theory of hegemony is a theory of innovation based on stability.

The telephone system, for most of its life, was an example of a system managed according to the theory of hegemony, with single providers (often parts of the government) in most regimes, and standards set through a very deliberative body: the ITU (or earlier the CCITT). One interpretation of history is that this approach led to a very stable system that was easy to use, but not to a system that drove innovation. However, the low rate of innovation can be explained by the theory of utility: the "platform" provided by the telephone system, the 3kHz channel, was not very general (in other words, not useful except for the carriage of phone calls), so the failure of innovation is due to the limited utility of the platform, not the presence of the controlling interest. However, following the reasoning one step deeper, one could argue that this outcome is due to the lack of interest in innovation by the controlling interests.

The present Internet

A number of theories have been identified as contributors to the observed longevity of the Internet: the theory of the general network, the theory of the stable platform, the theory of semantics-free service, the theory of technology independence, the resulting theory of the hourglass, perhaps the theory of minimal global agreement, and (to some extent increasing over time) the theory of downloadable code (in the end-nodes).

The Internet seems to reject the theory of hegemony, and the theories of composable elements and downloadable code in the network.

Global agreement

The early designers of the Internet assumed that substantial global agreement would be a necessary step toward an interoperable network. (In those days, downloadable code was not a practical concept.) Over time (in an application of what was called above the "test-of-time" theory, the real degree of global agreement has emerged.

Addressing: The original design assumed a single, global address space in which all end-points (or interfaces, to be precise) were found. This idea has been replaced by a more complex concept, in which there are lots of private address spaces, with translation among some of them using NAT devices, but there is still a single *common addressing region*—a region in the "center" of the network where a pool of addresses are given a consistent common meaning. Services that want to make themselves widely available obtain an address (or an address and a port) in the common addressing region, so that other end-points can find them.

TCP: The original design was careful *not* to make TCP mandatory—this is the role of UDP. As well, the original design was careful to say that alternatives to TCP should be anticipated. The socket interface to TCP is not a part of the Internet standards, Over time, however, in an example of the dynamic form of the theory of the stable platform, enough applications have used TCP that it is mandatory in practice, which means that other applications take on little risk in depending on it, and TCP has emerged as a required point of global agreement.

TCP-friendly congestion control: This idea was no part of the original design—in the beginning the designers did not have a clear idea about dealing with congestion. However, in the 1990s (more or less), as conggestion control based on the slow-start algorithms and its enhancements matured, there was a sense that every application, and every transport protocol, needed to behave in the same general way. So there was a call for a global agreement on the congestion behavior called "TCP-friendly". To a considerable extent, this norm was imposed, but it seems today as if there is a drift away from this approach (based on economic issues and the theory of tussle) to a model where the network takes on a more active role in enforcement.

DNS: The architects of the Internet have always been ambivalent about whether the DNS is a core part of the architecture. It is not strictly necessary: one can use other tools to translate names into addresses (as some applications do), or just type IP addresses where one would normally type a DNS name (e.g. in a URL). However, as a practical matter, the DNS is a necessary component of the Internet for any real use, and the amount of tussle of the DNS (trademark, diverse alphabets, governance, TLDs, etc.) both suggest that it is a point where global agreement is required, and also that it is a prime illustration of tussle. One can look at the simple interface (send a name, get a number) as a stable platform interface under which all sort of change has happened.

The Web: The web standards have emerged as a critical platform for the growth of the Internet. While the web is "just one application among many" it is clearly (as of now) the dominant application, and as such, embodies many attributes that can be

explored using these various theories—tussle, platform, downloadable code and so on. But without global (if rough) agreement on many aspects of the web, the Internet experience would not be what it is today. The specification and deployment of SSL and TLS a good example of the injection into the Internet of a new set of points about which there needs to be widespread (if not quite global) agreement.

The packet header: Participation in the Internet does require agreement on how a packet is formatted, and what (at least some of) the fields mean. The address field may be rewritten as the packet traverses NAT boxes, but there are still some constraints imposed by Internet addressing (e.g. the length, the TCP pseudo-header and the like) to which all players must conform. Despite the push to deploy IPv6, the IP header seems to be a manifestation of the stable platform, rather than something that is shaped by a theory of change.

The future

There are a number of considerations and theories about how to design a future network such that (among other things) it is long-lived.

Security

A first order objective for a future Internet is that it be more secure. Security (attack and defense) is perhaps the extreme example of tussle; it implies ongoing (and unpredictable) change, even if attack and defense stays in some tolerable balance. So the presence of attackers in the system would seem to imply that at least some part of a future Internet must seek longevity using a theory of change, not a theory of stability.

Any component in the network will be a target for attack. So the theories of building blocks and composable elements might seem to lead a a future network with more options for security vulnerabilities. This concern must be addressed by advocates for those theories.

Management

The discussion of the Internet of today focused on the data plane, and considered issues of addressing and naming from the point of view of global agreement and stability. That discussion paid little attention to issues of management, in part because that area is so poorly developed from an architectural perspective. In a future Internet, management must receive more attention, for a number of reasons. This objective will lead to the creation of a new set of interfaces, and will raise a new domain to which these various theories must be applied. Many of the Interfaces will be between peer components (between ASes or ISPs) so they are not *platform* or layering interfaces. It is not clear what theory of longevity should apply to such interfaces.

Virtual networks

The idea of virtual networks is an example of a theory of the stable platform as a basis for change. It is also a new approach to a theory of the general network. The concept of the virtual network arises (abstractly) as follows. A network is build out of communication technology (e.g. fiber links, wireless, and the like), and processing nodes

(which play the role of routers, packets processors and the like). At this low level of abstraction, the network is not ideal (since real impairments arise from the choices made in the deployment of the elements), but it embodies only these intrinsic impairments. If these low-level resources are virtualized in a complete way (so that no aspects of the functionality other than raw performance are lost in the virtualization), then the platform provided is as general as it can be, since the only impairments are the intrinsic ones arising from the choices made about the physical elements.

According to this view, the higher-level networks that are build on top of the virtualized resources can be replaced more easily than the physical elements, and multiple such nets can be run at the same time, which is a tool to deal with change. So the generality of the virtualized elements is the stable platform over which change can be accommodated. To the extent that new technology is introduced into the substrate, the higher-level networks may have to be extended to incorporate it—virtual networks are not technology-independent.

The interfaces that are implied by the virtual network concept are those by which virtual resources are allocated, software is downloaded, and the like. We must assume that (according to the theory of tussle), these interfaces will become points of contention, security attacks and the like.

If tussle (and the attacks and defense of security) arise as issues at the virtualization layer, then change at the substrate layer may erode the stability of the platform, which (according to the theory) will increase the risk of those who build on top of the platform. So one of the challenges to those who advocate this approach is to argue that the resulting platform can be stable enough to succeed in the market-place.

Highly programmable networks

Virtual networks can be seen as an end-point along a spectrum where the ability easily to replace components different applications could exploit two networks designed on completely different principles. The point of global agreement is the set of tools used to download software into the elements, to allocate the virtual resources and the like.

There are other points along this spectrum, and (as noted above) the current Internet may be moving in this direction along the spectrum, (more so at the higher layers), where the ability to download code into browsers and the like is allowing competing standards to co-exist with little cost except for occasional confusion at the user level and the increased richness of security vulnerabilities that may be introduced through these new interfaces.

A more general service platform

Today, most applications get access to the Internet via a “socket” interface that presumes a two-way interactive flow among the end-points, which in practice means TCP. In tomorrow’s network, many nodes may only be connected intermittently, and many applications may be able to tolerate a more “store-and-forward” mode of transport between the end-points. So a more general application API may be an important part of building a more general version of the stable platform.

Citations

Belady, L. A. and M. M. Lehman (1976). "A model of large program development." IBM Systems Journal **15**(3): 225-252.

Clark, D. D., J. Wroclawski, et al. (2005). "Tussle in cyberspace: defining tomorrow's internet." IEEE/ACM Trans. Netw. **13**(3): 462-475.

Saltzer, J. H., D. P. Reed, et al. (1984). "End-to-end arguments in system design." ACM Trans. Comput. Syst. **2**(4): 277-288.