Future Internet Architecture

and challenges of the current Internet

CHIRAG ARORA

S1456717

Master Thesis

Philosophy of Science, Technology and Society

Faculty of Behavioural, Management, and Social Sciences

University of Twente

Supervisors : Dr. JOHNNY HARTZ SØRAKER &

Dr. MICHAEL NAGENBORG

Acknowledgements

As there are no clear or even obscure boundaries between my life and times in writing this thesis and my life and times in general, the people I am about to thank have had contributed more to life than just helping me in writing this thesis. Of course, this puts me in the precarious position of choosing a few among many. But, perhaps the reader and I can agree that my failure in acknowledging everyone is somewhat inevitable and only trivial, and hopefully I have paid my respect to most, if not all, while sharing our conscious experiences.

With the above caveats in mind, I would first like to thank my supervisors Johnny Soraker and Michael Nagenborg, who have provided me with good advice but more importantly, with a lot of fuel for interesting thoughts both beyond and within the classroom. Along with Prof. Ambuj Sagar, who has been a friend, mentor and inspiration to me for a number of years now, they epitomize what I like best about Academia and why I see a life for myself within it.

Being a new country, far away from where you were born, can make you think and re-think the meaning of the word ‘home’. The very fact that I have not formed any clear opinions or found any real answers to what that word may mean, is in itself a testimony to all the friends I have made in Enschede and Europe in general. Among my ‘international’ friends who really deserve a mention are Jurjen, Thijs, Jannis, Sabrina,Tunmisse and Jerfy who have lived through my incessant story-telling and my love for exploring the ‘other’ side in debates, and who have also played along just as well. A similar role has played by Vinayak and Ateeth, who have also made sure that I am able to get my share of exercise in the week by biking to a place that is away from the grey of the city. The time spent with them and the trees who have witnessed our average poetry recitation skills has been as free from the fetters of the past and future as any time can be.

A thesis about Internet shouldn't go without acknowledging the role of this technology in

letting me feel even among those who are physically far away. Most of them have been long enough in my life to exclude any possibility of an acknowledgment that can be written in a few lines and still reflect everything I have to say. So I am just going to put out some names: Geet, Anuraag, Ritesh, Abhishek, Chetan, Rajat, Ankur, Ashu, Amit, Parv. Finally, I would end this rather vain exercise by thanking my Parents and my Brother, who have been great friends and had an overwhelmingly pleasant role in my Being.

Let me also just add these words by Rilke, which reflect what I may or may not have achieved on this page, or in my thesis, or life in general, but aspire(d) to:

"*...And I want my meaning*
true for you. I want to describe myself
like a painting that I studied
closely for a long, long time,
like a word I finally understood,
like the pitcher of water I use every day ,
like the face of my mother,
like a ship
that carried me
through the deadliest storm of all."

Abstract

In the last three decades, the Internet has seen unprecedented growth, scaling from a few hundred early users to more than 3 billion global users, enabling a transformation of a wide range of economic, social, political and cultural practices. However, along with this growth, Internet has also witnessed a wide range of growing conflicts among its stakeholders. The issues raised through these conflicts, which include concerns over privacy, security, censorship, etc., are at the heart of debates over Internet's design and regulation held between stakeholders such as governments, private enterprises, activists and designers. Given the impact of Internet's technical architecture over human life, a number of scholars within the Internet research community, arguing that the current architecture is too constrained to cater to contemporary concerns, advocate for a clean-slate architecture design for a future network. While a clean-slate architecture provides us with an opportunity to find solutions to our current moral dilemmas, it also leaves us with the responsibility of anticipating moral dilemmas that may arise in future because of such a design. This thesis aims to take up this responsibility and provide a step towards a future Internet. Using the conceptual notion of 'architecture', which allows for a decomposition of a complex system, like the Internet, this thesis argues for four critical meta-requirements for a future Internet architecture. These meta-requirements, which are properties of the system critical to it beyond its basic functionalities, cater to four characteristics of a complex system like Internet which underlie its current as well as potentially future challenges: inherent uncertainty and lack of predictability of future use of a network like Internet, diverging stakeholder interests and views, lack of feedback or data about how the network is being used, and diversity of societies Internet is embedded in. I conclude the thesis by providing design principles, which are philosophical guidelines,for a future Internet, along with evaluation tools that a research team, which I argue should be multi-disciplinary, involved in development of a future Internet can use to evaluate the different specific implementations of these design principles and hence, move towards the development of a future network that can support the socio-economic goals of a diverse world.

Table of Contents

Chapter 1 : Introduction

In the past three decades, Internet has gone from a small community of users with mutual trust to one with a wide variety of stakeholders who do not know each other, and often find themselves having diverging and conflicting interests. For example, many users want privacy in their communication, while governments want to tap conversations for a variety of reasons. Certain users want to exchange media, such as music and movies, with their peers while the media industry, as the legal right holders, wants to stop them. These conflicts, along with other challenges borne by the increased size of the Internet, have seen a number of short-term technical fixes that may be harmful in the long term (Braden et al, 2000). For example, the increased incidences of security attacks, in the form of hacks or computer viruses, has seen deployment of Firewalls on computing devices which block potentially (but not always actually) harmful applications. Similarly, to provide additional services, network providers use technologies such as Deep Packet Inspection which breach the privacy and transparency of Internet communication as it looks through the data flowing across the Internet.

These changes violate the Internet's original design principles and are being widely debated among scholars, policy makers as well as in forums such as Internet Engineering Task Force (IETF) (Braden et al, 2000; Van Schewick, 2012:!5). These debates raise the concerns that such short-term fixes may destroy, or atleast hinder, Internet's capability to facilitate innovation, economic growth, free speech, knowledge exchange, and democratic exchange. Forums such as IETF, and other Standards Developing Organizations (SDO), have also become important sites for raising and addressing questions over the moral as well as legal responsibility of technical designers to protect or enable exercise of 'human rights-by-design' (Cath and Floridi, 2016) and thus, highlight the growing impact of the technical architecture on human life (Brown et al. 2010; Clark et al. 2005; Denardis 2013, 2014; Lessig,1999).

As a response to the various conflicts over Internet's design, usage and regulation, which need to be ethically addressed, a number of scholars within the Internet research community have suggested a clean-slate architecture design, that is, a new network that is unconstrained by the current system. While a clean-slate architecture provides us with an opportunity to find solutions to our current moral dilemmas, it also leaves us with the responsibility of anticipating moral dilemmas that may arise in future because of such a design. This responsibility demands a contribution from ethicists and philosophers, and this thesis is dedicated towards this responsibility.

However, before conceptualizing the design of a clean-slate architecture for a complex network like the Internet, we require a nuanced understanding of a.) what the term architecture means, as well as how they are designed in the context of network systems, b.) what is the architecture of the current Internet, c.) how does this architecture lead to the conflicts of Internet's design and regulation, d.) what can we learn from these conflicts and its relation to architecture in order to design a new architecture. The aim of this thesis is to address all of these issues in corresponding chapter 2-5. In Chapter 6, I will draw on these lessons and discuss how design of a new architecture can proceed. Here, I will also argue that any research aimed at designing a new network architecture to match the scale and utility of Internet, should involve a multidisciplinary team which includes not just technical experts but also non-technical experts such as those with knowledge of economics, social science, philosophy, law and policy making.

The complexity of large systems, such as the Internet, requires the designers to decompose the system into smaller components. The architecture of a system describes how such a decomposition is to take place as well as how different components interact with each other to produce the desired functionality. As Barbara Van Schewick (2012) defines it, architecture is a "high-level system description that specifies the components of the system, the externally visible properties of the components, and the relationships among components". Different architectures can however, map onto the same functionality. Yet, these architectures, with same functionality, might differ in some other system properties such as reliability, modifiability, security, and so on. Some of these

system properties can be extremely critical to the socio-economic impact of the system. For example, for a system present under conditions of extreme uncertainty, it might be advantageous for the system to have a capability to be modified easily.

Designers may often have to make trade-offs between the desired system properties as one system property may negatively affect some other property. For example, security as a property may negatively affect a systems capacity to support new applications. Building security into software systems may require designing specific features to be designed into applications, which may decrease the incentive for application designers. Similarly, addition of security features may also block applications originally compatible with the system. One example is the use of Firewalls on network systems. Firewalls monitor and filter data traffic going in and out of a communication network based on predefined rules. Since designing rules to separate between malicious and non-malicious data can an error-prone exercise, particularly in large networks, firewalls often result in blocking of non-malicious data or functions (Braden et al, 2000). The decision of trading off one system property can therefore be critical in determining the behavior of the system, such as its socio-economic impacts. In the context of this thesis, which aims to provide a step towards a new network architecture, this decision regarding the trade-off between different critical system properties (also referred to as meta-requirements in this thesis), is central to the design of a new architecture.

In order to facilitate the decision making regarding the trade-off between different meta-requirements, I discuss how these meta-requirements were mapped onto the architecture of the current Internet, as well as how that architecture relates to the current conflicts over Internet's design and regulation. In the former discussion, I introduce the concept of 'design principle', which is used to translate the desired meta-requirements onto the system architecture. Design principles describe the known interactions and outcomes in different architectural choices and guide the design of a system for specific meta-requirements. In particular, I discuss three design principles, modularity, layering, and the end-to-end principle and how they are implemented in the Internet architecture. As design principles are conceptual tools, scholars often disagree over their technical

implementation within the Internet architecture. Therefore, in this thesis, I will discuss and clarify these debates.

An understanding of design principles in the current architecture, and the system properties they facilitate, provides a foundation to better understand the debates surrounding the conflicts over Internet's design and regulation, in relation to its architecture. In this thesis, I aim to unpack these relations and show how the current debates such as those over 'Network Neutrality' or the lack of security measures against cyber-attacks through hacking and computer viruses can be conceptualized as debates over the design of Internet architecture. For example, the opponents of network neutrality argue that it restricts the provision of Quality of Service (which can enable services such as faster video streaming for customers willing to pay more) while the proponents argue that going against network neutrality would involve violation of the design principles of the Internet and that the resulting network would not only be architecturally incoherent but also less conducive to future innovation (Schewick, 2012).

However, since the central aim of this thesis is to provide a step towards a future Internet architecture, the answer to the question of how these debates should be settled within the context of current Internet is not particularly important. As mentioned earlier, designing a clean-slate architecture not only provides an opportunity to cater to some of our current problems, but also a responsibility to anticipate or cater to problems that may arise in future. Therefore, in this thesis, I aim to identify some common underlying conditions that not only give rise to the multiple challenges for current Internet (such as debates over network neutrality, cybersecurity, the economic challenges for service providers, conflicts over censorship and privacy on the Internet, and so on), but which may also give rise to future, albeit different, challenges for any network deployed globally on a large scale. In other words, the task in hand is to identify and address the underlying conditions that give rise to conflicts in their general, rather than specific technical forms. I discuss these issues in Chapter 5, which builds upon the discussion on the relation between architecture and challenges of current Internet presented in chapter 2-4. In Chapter 6, I give recommendations for the design of a future network like the Internet, bearing in mind that

it should address the underlying conditions that give rise to conflicts, as mapped out in Chapter 5.

Keeping the considerations presented above in mind, the central question for this thesis can be stated as :

What meta-requirements should guide the development of a future Internet architecture, and how can these meta-requirements be translated into architectural principles such that they lead to a future Internet architecture for an open and diverse world?

This general question can be broken down into five specific research questions which map onto the chapters 2-6 in this thesis:

1. What is a network architecture and what are some basic principles involved in Network architectures?

In this chapter, I discuss the concept of a network architecture as well corresponding concepts of meta-requirements and design principles, which I will argue, provide a more useful way of designing specific architectures aimed towards desirable socio-economic goals. I then discuss some of the basic design principles, in their general form, involved in design of network architectures.

2. What is the architecture of current Internet, what design principles guided this design and how are these design principles implemented in the Internet architecture?

In Chapter 3, I present the architecture of current Internet along with the design philosophy (or the design principles) that guides it design. Here, I also argue that design principles can have specific and critically different technical implementations. This Chapter lays down the groundwork for discussion of the debates and conflicts over Internet's design and regulation and their relation to its architecture.

3. What are the current debates and conflicts over Internet's design and regulation and how are these conflicts related to the architecture of the Internet?

In this Chapter, I map out some of the current debates and conflicts over Internet's design and regulation as well as their relation to the architecture. These include issues such as the economic challenges faced by Internet Service Providers, cybersecurity, Network Neutrality, as well as the debates about the use of use of technical extensions that violate the original design principles of the Internet architecture. An understanding of these issues allows for a reflection on the underlying causes of these problems in their general form. Such a reflection, on the underlying causes of the conflicts over Internet's design and regulation, is particularly important in the context of this thesis, which aims to provide steps towards a future network architecture (rather than provide solutions for these conflicts within the current Internet architecture).

4. Why do conflicts emerge over Internet's design and regulation and what meta-requirements should a future Internet architecture have in order to address the concerns raised by these reasons?

The opportunity to design a future Internet architecture also needs to be complimented with the responsibility of approaching this design through an ethical reflection of potential challenges of the future that any large network with a scale of Internet is likely to face. Therefore, in this chapter, I present some of the underlying reasons for emergence of conflicts over Internet's design and regulation in their general form. The approach in this thesis is not just to provide solutions to, or perspectives on, current debates over Internet's design and regulation but also to conceptualize an architecture that is enabling in the future as well, particularly considering that the challenges of the future cannot be predicted with significant accuracy. To this end, I argue that the conflicts over Interrnet's design and regulation emerge because of four underlying conditions: inherent uncertainty and lack of predictability of future use of a network like Internet, diverging stakeholder interests and views, lack of feedback or data about how the network is being used, and diversity of societies Internet is embedded in.

For example, the unanticipated growth in number of users, emergence of business opportunities, increase in computing powers of computing devices, emergence of new technologies such as wireless phones, have led to unanticipated outcomes for the Internet. The history of the Internet points to the inherent lack of predictability of outcomes for a future network of similar scale and reach. Therefore, I argue that the Internet suffers, in the form of the current conflicts over its design and regulation, not because of an error in prediction on the part of the original designers, but rather from the impossibility (or extreme difficulty, at the very least) of predicting future outcomes for a complex system like the Internet.

Can we however, use this lack of predictability of outcomes and inherent uncertainty as an opportunity rather than treating as a risk that needs to be protected against? In this context, I discuss the concept of resilience, that has emerged out of research on anticipation (O' Malley, 2010). Resilience differs from the traditional approaches to uncertainty such as risk and preparedness. While probabilistic risk is a technique of harm-minimization, preparedness involves creation of mechanisms for coping with imagined harms. Resilience, however, is a more encompassing and systematic approach to anticipation as well as toleration of disturbances that occur out of uncertainties. However, I argue that resilience is essentially a reactive concept, still reliant upon a prediction based deterministic outlook of the world, and is harder to apply to complex systems where prediction is difficult, if not impossible. As a substitute for resilience, I argue that the notion of 'antifragility', a neologism coined by Taleb (2012), referring to the capacity of a system to gain from stress or disorder, is more useful. The Antifragile approach towards development of systems moves away from the Newtonian paradigm of deterministic control (Heylighen et al, 2006; Taleb, 2012), and seeks to thrive on uncertainty rather than protect against it. In the context of a network like Internet, this capacity rests upon the network being evolvable, and conducive to innovation. Here, I also discuss that the benefits of an evolvable and innovation facilitative design can already be witness through the history of the Internet.

I further argue that Internet's complexity and the diversity of the stakeholders involved, make it difficult to reach a point of stability, with conflicts emerging in 'run time'. While the designers can tilt the weights towards one stakeholder over the other others, such a move can lead to various harmful consequences. For a global technology laden with uncertainty and a diverse stakeholder set, designers need to be conscious and cautious about not embedding their own values into the design. However, unlike relatively simpler technologies where designers can, to some extent, assess different stakeholder views as well as consequences of their design, a large scale network like Internet adds multiple challenges to such a task. For a network like Internet, new stakeholders or new stakeholder perspectives keep emerging years after the design, thereby, making the methodological use of well known approaches to design such as Value-Sensitive Design (VSD), which aim for creation of technologies such that they realize desirable human values (Friedman et al, 2013), difficult. It is , however, still imperative that designers deliberate upon their choices and are diligent about views different from their own as they have a responsibility to the 'other', and to protect the multiple ways of knowing and being in a multicultural world (Van der Velden, 2007). Therefore, I argue that a future Internet needs to provide sites for stakeholders to put forward their views as well as points of control through which these views have the potential to be materialized. I back this reasoning by providing the limitations of technological 'lock-in' scenarios (Callon, 1990) which eliminate the potential for an open future, which is one of the primary goals recognized by IETF as well (RFC 3935).

Similarly, drawing from the debates over current Internet's design and regulation, I argue that a future network can significantly benefit from addition of intelligence. Here intelligence refers to the ability of the network to collect as well as process data about its own use characteristics. Such an information loop, or feedback loop, is present in many natural as well as manual complex systems with many advantages (Astrom and Murray, 2010). In case of a future network, such a Feedback can facilitate a diagnosis in cases of failure as well as allow the network to evolve, based on the information collected, while preserving its critical system properties (one of which is evolvability itself). Finally, I argue that the architecture for a future Internet should be minimally defined and made

heterogeneous (flexible over different regions over network), because there will be few requirements that will be truly global.

Drawing on these underlying reasons likely to cause challenges in any future network, I derive meta-requirements for a future network : Evolvability, distributed control to stakeholders, Intelligence within the network, and Heterogeneity.

5. How can these meta-requirements, derived in chapter 5, be translated into architectural principles such that they lead to a future Internet architecture ?

In this Chapter, I discuss how the desired meta-requirements for a future Internet architecture, as derived in Chapter 5, can be translated into design principles. I also discuss how these design principles can guide the design of a future Internet architecture along with some of the challenges that may come up in this task. In order to derive these design principles, I compare and critique design principles for a future Internet suggested by other scholars and address their shortcomings using the arguments developed in chapters 2-5.

However, it should be said that the purpose of this thesis is not to provide specific technical implementations of these design principles, as this is beyond the scope of this thesis as well as expertise of this author. Any project aimed at development of technically implemented future Internet should involve a multi-disciplinary teams of experts, whose expertise ranges from technical fields to non-technical fields such as economics, humanities and law. Such a project is likely to be iterative in its process and would require the efforts of a multi-disciplinary team in not just the design phase but in tests and pilot phases as well. This thesis does however, provide guidelines for such a project to be undertaken and to move towards the goal of a future network than can support the socio-economic goals of a diverse world.

Chapter 2:
Network Architecture and Some Basic Principles

As the central question for this thesis is to propose steps towards the architecture of a Future network, it is essential that the notion of what a network architecture and why it is important as a concept is clarified. In this Chapter, I aim to provide this clarification as well as introduce the corresponding concepts of meta-requirements and design principles which are central to the arguments to be followed in this thesis.

In designing of a complex system, designers typically seek to decompose the system into pieces or components in order to reduce the complexity of the problem of design. An architecture describes the decisions about such decomposition of a system, and represents the necessary information regarding how the different components work together to satisfy the required functionality. As Barbara Van Schewick (2012) defines it, architecture is a “high-level system description that specifies the components of the system, the externally visible properties of the components, and the relationships among components”.

The choice of the requirements then, is critical in development of the architecture and its usefulness. In the context of Internet architecture, David Clark (1988), one of the initial researchers in the DARPA Internet Project, provides a summary of the requirements underlying the Internet architecture. These requirements can be summarized in their order of importance (most important requirements first) as following :

1. Internetworking: existing networks must be interconnected.

2. Robustness: Internet communication must continue despite loss of networks or [routers].

3. Heterogeneity: The Internet architecture must accommodate a variety of networks

4. Distributed management: The Internet architecture must permit distributed management of its resources

5. Cost: The Internet architecture must be cost effective.

6. Ease of Attachment: The Internet architecture must permit host attachment with a low level of effort.

7. Accountability: The resources used in the internet architecture must be accountable.

The order of these requirements is important, shedding light on the context within which they were decided. For example, as the network was being designed to operate in a military context, the goal of accountability or cost effectiveness feature below the requirements of robustness or distributed management.

Yet, these functional requirements do not map onto a specific Internet architecture. Many different architectures can be shaped to provide the same functionality. For example, changes to one component can be offset by changing components interacting with it, such that the final outcome remains the same. This interchangeability can be seen as a powerful feature of softwares as they can be much easier to change as compared to hardware systems which have to contend with natural physical laws (Schewick, 2012 :22). Thus, while functionality does depend on and acts and guide to a correct architecture, it does not map onto a specific architecture.

2.1 System Properties or Meta-Requirements

Specific architectures, however, do differ in terms of a number of system properties. For example, properties such as security within the network, reliability, usability, performance can change with specific operation of the system (Schewick, 2012: 22). Similarly,

properties like modifiability and testability depend on architectural choices that define the development and maintenance of a system. For example, a system is more modifiable if to implement a functional change, the components required to be modified are few or concentrated together. In design of certain systems, besides the functional requirements, one or more system properties maybe critical. For example, a system which has a lot to gain from improvements to its components and where this improvement is very likely as well, modifiability might be a very critical property. For a particular system, some of these properties maybe critical to the overall system while some may not be (however, they may be desired within specific components or parts of the system). In this thesis, properties that are critical to the overall system beyond its basic functional requirements, will be referred to as meta-requirements.

As these properties can be dependent upon specific architectures,it is important that an architecture supports the desired properties. This task is however, made difficult by the fact that some of these properties may have a competing relationship to each other, requiring a trade-off. That is, one property may negatively effect another property (Kazman et al, 1998). For example, security as a property may negatively affect a systems capacity to support new applications. Building security into software systems may require designing specific features to be designed into applications, which may decrease the incentive for application designers. Similarly, addition of security features may also block applications originally compatible with the system. One example is the use of Firewalls on network systems. Firewalls monitor and filter data traffic going in and out of a communication network based on predefined rules (Braden et al, 2000). Since designing rules to separate between malicious and non-malicious data can an error-prone exercise, particularly in large networks, firewalls often result in blocking of non-malicious data or functions.

The trade-off between such properties is not an architectural choice (although it guides architectural choices) and depends on what is desirable for that particular system. The desirability of a property may in turn depend on the socio-economic goals (a combination of social and economic factors) of the system.

2.2 Design Principles

It can be difficult to directly translate the desired properties into an appropriate architecture. For this purpose, design principles can be useful.

Design principles describe known interactions and outcomes in different architectural choices. They may constrain decomposition of a system into components, distribution of functionality among components, or interactions between components. In software terms, design principles are termed as 'architectural styles' (for a detailed description of architectural styles, see Shaw and Garlan, 1996). A design principle guides the design of a system for specific properties (and trade-offs made between them) and resulting architectures differ depending upon which design principle was used to design them (Schewick, 2012 : 23). The choice of a design principle depends on the desired properties of the system.

Design principles can not only play an important role in initial architectural design, but continue to be critical as they guide the design that follows. Adhering to the original design principles can ensure longevity for an architecture as well as coherence between different components. Changes in components that violate design principles can change the balance (or the selected trade-off) between different properties. It should be noted that in this thesis, 'design principles' cater to the overall architecture of the system (or its meta-requirements) and not to the design of individual parts of the system, unless explicitly stated.

2.3 Internet Design Principles

As argued before, design principles shape the architecture and reflect the trade-offs made between different properties of the system. In this section, I will describe three design principles- modularity, layering and end-to-end arguments- that have critically shaped Internet architecture and are the heart of the economic or other conflicts over Internet's design and governance. The following paragraphs would describe the trade-offs

underlying each of these design principles and the constraints that they impose on the Internet architecture.

2.3.1 Modularity

System architectures can differ in the degree of coupling (whether loose or tight) between their components. Here coupling is a measure of the degree of interdependence between components (Schilling 2000 : p312). As a design principle, Modularity refers to a design where components are intentionally made highly independent (or loosely coupled). Components of a modular architecture are referred to as *modules*. In designing a modular architecture, architects decompose the system such that dependencies among modules is minimized.

Modularity's goal is to allow design of components independently such that they can still work together. To achieve this, Modularity employs strict separation of concerns and information hiding (Schewick, 2012 : p38). The following discussion should provide more clarity on these mechanisms.

In a Modular architecture, information can be distinguished into two types - visible information and hidden information. An architecture's visible information is that which is relevant to design of all modules or at least more than one module. This information is specified as part of the design process and is typically not allowed to change later in the detailed design stage (Schewick, 2012: 38). For each module, this visible information defines its inter-dependencies and points of interaction with other modules.

In contrast, hidden information in an architecture only concerns a particular module, and is therefore, hidden from everyone except the designers of that particular module. Since it is independent of the overall architecture, this information need not be specified during the initial design phase, and is free to be changed during detailed designing of the particular module involved. It is also free to evolve within the framework provided by the visible information of the architecture.

In a modular design, each module is like a blackbox, and its data and services are restricted to its own interface. Design of other modules is not allowed to make any assumptions about this interface, restricting the design to only the visible information of the architecture. Further, as the hidden information related to one particular module does not concern other modules, each module can proceed to its detailed designing independently.

As an example, consider the design of a personal computer. The peripheral devices, such as keyboard, mouse, printers, screens, attached to it can be seen as different modules. As part of the design specifications, the interfaces between these peripheral devices are defined by industry specifications. For example, the size of the plugs, or shape of the ports. Thus, a designer of such a peripheral device is only concerned with those specifications and need not worry about if the computer will be able to read data from it or not. All peripheral devices can thus be designed independently.

Another advantage of modular systems is that they can be understood and tested more easily, hence, increasing the correctness of the system. In a highly modular architecture, the connections between components are weak (that is, they are loosely coupled). This means that communication between components is weak and the interfaces between them are simpler. This simpler interface makes testing and identification of errors easier, specially because highly independent components can be tested separately. In contrast, the effect of errors within a highly coupled system is more complex to understand and identify.

Modularity, however, is not without its costs. As already mentioned, the visible information of an architecture is not changed during detailed design and is difficult to change later in the system. The flexibility provided by the hidden information is then in contrast to the inflexibility of the visible information. In order to prevent this from becoming problematic for the system, designers typically put components that are likely to benefit from a later change or variation together or in the same module (Schewick, 2012: 40).

Modularity may also negatively affect the overall performance of a system. The performance of a software system, among other things, depends on the communication between modules (Schewick, 2012: 43). The performance of a system can increase with decrease in inter module communication, as lesser distance can provide faster communication between modules. However, in a modular architecture, components that are likely to change together are put close together while those that are unlikely to change together but may still be cooperating, are separated. This separation between cooperating modules, which need to communicate with each other, can thus, negatively affect performance by slowing down the communication between them.

Overall, Modularity is a preferable design principle for a system which is likely to benefit with a trade-off where goals such as modifiability, less complexity, independent design, easy error identification are preferred over potentially higher performance of the system or constraints by visible information.

2.3.2 Layering

In a modular design, there are no impositions regarding interactions between modules. In this sense, layering is a special kind of modularity, where in addition to having a modular design (loose coupling between components, for example), there are additional restrictions on inter-module interactions. As the name suggests, in a layered architecture, modules are organized in layers. A module assigned to a particular layer can only have dependence on modules on the same layer as itself or a module in a lower layer. However, it is not allowed to have a dependent relation with a module in a higher layer (Schewick, 2012: 46).

Similar to a modular architecture, design of a layered architecture proceeds on the basis of ‘visible’ and ‘hidden’ information. A layer’s interface with other layers is designed based on visible information, while its internal implementation is hidden from the designers of other layers and can proceed independently.

A layered architecture can further be categorized into three categories: Pure, Relaxed and Relaxed with a portability layer (see figure 2.1; Schewick, 2012: 47). In the pure version of layering, each layer can only use the layer immediately below it (That is, it can only access services of the layers below it). In the relaxed version, layers are permitted to use any layer below them. In a relaxed layering with a portability layer, which is also the layering in Internet architecture (see chapter 3), each layer can use the layer immediately below it as well as a portability layer. This portability layer is one of the lower layers providing service to all the layers above it. The choice between the three versions of layering depends on the type of system as each may offer different advantages to different systems (for example, Internet benefits from a relaxed layering with portability layer. See section 3.1 for more details).

As layering is just a special case of modularity, it has some of the same advantages and disadvantages as modularity.

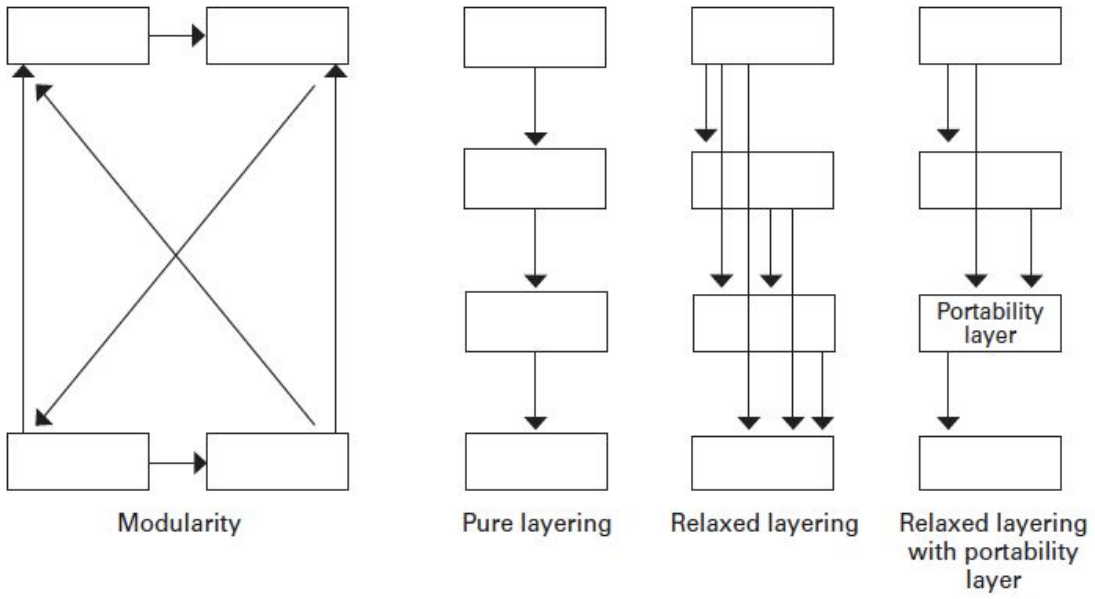

Figure 2.1. Modularity and Layering. Figure from Schewick, 2012 :46

2.3.2.1 Layering in Network Architectures

A communication network's goal is to allow applications to interact through the network. As this task is fairly complex, designers use a desirable version of the layering principle, decomposing the system into various components.

In terms of devices or computers interacting in a network, there are broadly two types of components: those 'in' the network and those 'on' the network. Computers 'on' the network are those that are used by users and support application programs. The computers used at our homes and offices, for example, belong to this category. Because the communication begins or terminates through one of these computers, they are also called 'end' devices.

Computers 'in' the network are those that implement the network.These include modems and routers installed to provide Internet access and send data across from one user to another. As they allow communication to flow through them, they are called as the 'core' of the network. This distinction between 'end' or 'core' devices is not unique to the Internet. For example, a telephone network with switches has telephones as end devices and switches as the core of the network.

Each computer, whether its an end computer or from the core of the network is further divided into layers, arranged vertically (See Figure 2.2). Each layer interacts with another layer through the use of architectural components known as *protocols*. Protocols can be thought of as rules defining how layers interact. Each protocol, belonging to a layer, provides it services to layer above it, and uses the services from the layer below it. Protocols' service works by exchange of messages between the same protocol located on different computers. In this way, layering in network architectures operates horizontally (through exchange of messages between the same protocol belonging to a layer) and vertically (through the services of lower layer protocols).

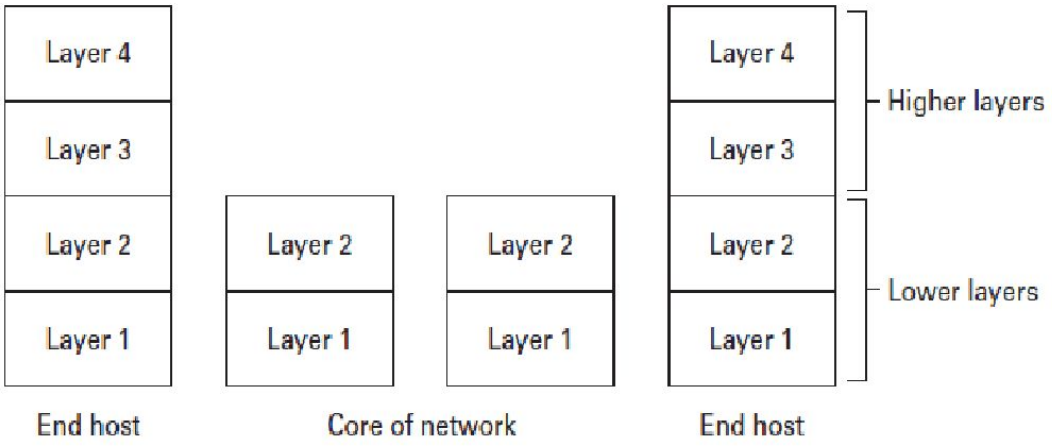

Figure 2.2. Layering in network architecture. (figure from Schewick, 2012:51)

The functionality of end computer and those in the core depend upon how functionality is distributed among the layers of the network. Typically, higher layers are only implemented on the end computers, while lower layers are implemented on all computers. Thus, higher layers usually run application programs that users interact with.

Finally, a layer may operate hop-by-hop or end-to-end. A hop-by-hop operation is one where a protocol operates at each step (consisting of computers) between a destination and the host (where message starts). Whereas an end-to-end operation is one where an operation happens directly between the destination device and the host (Schewick, 2012: 63).

2.3.3 End -to-End Principle

Saltzer et al (1981) first identified, named and described a design principle used implicitly in system design, known as the end-to-end principle. Schewick (2012:58) argues that there are actually two versions of this principle, described by the authors in different papers. The differences between the two principles are useful in understanding the current Internet architecture to follow in the next chapter as well as in highlighting

how minor differences in design principles can lead to significant differences in system properties. Schewick (2012:58) terms this two principles as a ‘narrow’ version and a ‘broad’ version. In the following paragraphs, I will describe both versions and how they support different system properties.

2.3.3.1 Narrow Version

“A function should not be implemented in a lower layer, if it cannot be completely and correctly implemented at that layer. Sometimes an incomplete implementation of the function at the lower layer may be useful as a performance enhancement” (Schewick, 2012: 60).

As mentioned earlier, the higher layers of a network are implemented on the end computers only while the lower layers are in all computers and therefore, they are in the network. The narrow version of the end-to-end principle highlights that there are some functions that can only be implemented end-to-end (that is between original source and destination of data) and therefore should be placed in higher layers. The placement in higher layers is because they cannot be completely implemented at the lower layer.

But what kind of functions cannot be implemented at lower layers? Any function that cannot be completely implemented at the lower layers without having access to the services of a higher layer (or in other terms, without violating the rules defined by layering principle). For example, consider a computer where messages flow across the three layers of network. Now this computer needs a function for error control which checks against corruption of data as it flows across layers. If this function is implemented at a lower layer (say layer 2), it will not be able to work on data that gets corrupted after data leaves layer 2 to layer 3. The function can only be successfully implemented at layer 3. The narrow principle, however, mentions the possibility of incompletely implementing at a lower layer for performance enhancement. In the same example, consider if the function is also implemented at layer 2. In this way, the function implemented at 3 does not need to check all the layer interfaces for error correction, dividing the task with

implementation at layer 2, hence, increasing the performance. The narrow version of the end-to-end argument allows this.

2.3.3.2 Broad Version

“ A function or service should be carried out within a network layer only if it is needed by all clients of that layer, and it can be completely implemented in that layer. ” (Schewick, 2012 : 67).

As the broad version does not give any exemptions, and requires that they be needed by all clients of the layer, it dictates that application specific functions should only be implemented at higher layers of a network and not on lower layers. Lower layers on the other hand, should provide very general services that can be used by all higher layers (Schewick, 2012 : 67).

In the example of the error function given above, the broad version would not have allowed its redundant implementation at a lower layer as well. The broad version, thus, can be said to not value the performance of a system. However, the broad version makes this trade-off for some other system properties such as evolvability, application autonomy and reliability.

Evolvability - As argued above, the broad version of the end-to-end principle dictates that the lower layers only implement general functions to be used by higher layers. Thus, the application specific functions are implemented only at higher layers of the network. However, if, for example, one follows the narrow principle, and a redundant copy of the function is also implemented at a lower layer, this will usually enhance the performance of the system. Yet, this functionality will be detrimental to programs that do not need it, hence, constraining the kind of programs that can be put on the system. In this sense, the system is less evolvable in the narrow version.

Consider the example of the public switched telephone network designed to transmit

human speech. The network transmitted sounds within the frequency range of 300-3,400 Hz (Schewick, 2102: 69). However, the strength of the signal deteriorated significantly when transmitting over large distances through cables. To counter this loss, network providers used devices known as *load coils*, which boost signal strength, on long cables (longer than 5000 meters). One of the side effects of these devices was that they cut off frequencies above 3400 Hz, which was not problematic as telephony did not use frequencies that high. However, Digital Subscriber Line (DSL), a technology introduced later and used to transmit digital data over the telephone network, used frequencies above 25000 Hz. Thus for DSL to be used the load coils had to be cut off. While the installation of load coils had optimized the performance, it had made the layer *in* the network application specific and not usable for DSL. The broad version of the end-to-end principle would not have allowed this, keeping the network open to new technologies more easily.

Application Autonomy - Application autonomy is a concept that refers to the idea than an application or higher layers on which it is implemented should decide the services they need based on their specific needs (Schewick, 2012: 71, Saltzer et al, 1984, 281). The merit in this approach is that the designers for lower layers do not have to guess or make assumptions about what kind of applications would be implemented on the system.

Reliability - Implementation of application specific functions in the network (in the lower layers) can introduce additional points of failure. These additional points will not only be difficult to identify but also out of the control of an application designer. Thus, application specific implementation in higher layers, which the broad version dictates, can increase the reliability of a network.

In this chapter, some basic principles involved in network architectures were introduced. It was argued that there are some meta-requirements or system properties that are critical to a system, beyond its functionality. Further, these system properties can be contradictory to one another, and therefore, require trade-offs to be made. To implement these properties, based on the desired trade-offs, system designers use Design Principles.

Design principles allow designers to effectively translate the required system properties into architectural features. Some of the design principles involved in network architectures, and the system properties they support or constrain, were discussed. An understanding of these design principles is critical in understanding the architecture of the Internet, and consequently the problems with the current Internet. These problems are particularly rooted in the use of the Internet, which is constrained (and defined) by the design principles of the Internet. Th next chapter will focus on locating the discussion of design principles presented here in the context of the current Internet architecture.

Chapter 3:
Original Internet Architecture

In order to understand the problems of the current Internet, particularly the conflicts over its design, use and regulation, it is important to understand the architecture of the Internet. As argued in the previous chapter, a useful tool to understand the architecture and the system properties it leads to, is through design principles. In this chapter, I will discuss the layering and end-to-end design principles within the context of original Internet architecture, and how their implementation promotes certain system properties on the expense of others. This would lay down the foundation for the discussion on the contestations over Internet's design, use and regulation.

3.1 Layering

The Internet' architecture is divided into four layers[1] based on functionality. Each layer has one or more protocols to implement these functions. These layers are: link layer, the Internet Layer, transport layer and the Application layer (Galloway, 2004 : 40-42).

3.1.1 Link Layer

On the Internet, messages are broken down to smaller units called data packets. The link layer is responsible for transporting these data packets to a computer or device attached to the network physically. This layer includes technologies that enable our connection to Internet such as WiFi or ethernet. As these technologies are different, the link layer has more than one protocols.

[1] Some scholars may differ on the number of layers that can be conceptualized for Internet architecture. For example, Solum (2004) defines 6 layers. However, the difference between the number of layers is conceptual and depends on the argumentative context. In Solum (2004), the application layer as defined in this thesis has been divided into another layer, the content layer. However, for the purposes of this thesis, application layer consists of everything user interacts with, including the contents on those applications and the division of application layer into further two layers does not conceptually alter the arguments of this thesis (or even benefit them).

3.1.2 Internet Layer

The Internet layer has a single protocol called the Internet Protocol (IP). The IP is responsible for enabling any pair of end computers to exchange data packets between them. The Internet layer is the highest layer to be implemented in the core of the network. It is implemented on end computers as well as routers that make up the core of the network.

IP's services are often referred to as being 'best effort', in that while it does do its best to get data packets across, it does not guarantee anything against data loss, delays or bandwidth problems (Peterson et al, 2007: 236-238). Thus, IP provides an unreliable service. Further, IP's service is connection-less, in that it does not establish a virtual connection between the host and the destination computers. Each data packet (also known as data gram) is treated independently and moves from one connection to the other until it reaches the destination. Because of this method, it is possible that two data packets belonging to the same message may take different paths. To identify different machines on the network, IP uses an addressing scheme, where a unique identifier - IP address- can be derived for each host and router in the Internet.

3.1.3 Transport Layer

The IP does not differentiate between the applications on the same end host. Transport layer protocols, however, do differentiate between different applications running on the same host and enable exchange of data between a sending application on the source computer to a receiving application on a destination computer (Schewick, 2012: 86). This allows multiple applications on the same end computer to run and communicate through the network simultaneously.

The transport layer consists of two protocols : Transfer Control Protocol(TCP) and the User Datagram Protocol (UDP).

TCP is the popular of the two protocols and provides a connection oriented reliable data exchange service. It lets two application layer processes on different computers establish a connection, exchange data, and then close this connection. TCP ensures that there is no data loss, errors or duplication of this data. The data arrives at the destination as a continuous stream. In contrast, UDP sends data as separate messages, and does not offer guarantees against data loss or duplication. The differences between the two protocols are function based, that is, they are used to support different functions. TCP is used for applications that require reliable data transfer (for example, e-mail) while UDP is used for applications that do not require reliable service, and cannot tolerate time delays (for example, digital telephone services such as Skype) (For more on this, see section 3.2).

3.1.4 Application Layer

As the name suggest, the application layer has all the user applications, such as E-mail,World Wide Web (WWW), Skype, video streaming applications, operating on it. The application layer has a range of different protocols that allow application programs to communicate with one another through the specified conventions of the protocol. For example, Hypertext Transfer Protocol (HTTP) is the protocol for communication between web browser and webs servers. When a user requests a page on a web browser, which could be Internet explorer, Google Chrome or Mozilla Firefox, the browser contacts the web server using HTTP and then receives the page from the server.

Application protocols can be standard or proprietary. For example, the protocol for communication on Skype between different computers is proprietary and can only be used for that particular application (and not by some other telephony application). However, protocols for some other applications have been standardized by The Internet Engineering Task Force (IETF). One good example of this is E-mail, which uses the Simple Mail Transfer Protocol (SMTP). As a consequence of this standardization, one can send and receive e-mails to any e-mail address regardless of which e-mail service they are using (a Gmail program can send emails to a Yahoo mail program, for example).

As described in the previous chapter, layering in network architectures is of three types :pure, relaxed, and relaxed with a portability layer. Internet employs a relaxed with a portability layer type of layering. Here, the portability layer is the Internet Layer. Thus, higher layers, the transport layer and the application layer, can employ the services offered by IP, the protocol of the Internet Layer. As it is with the rules of layering, the application layer can employ the services of transport layer protocols, TCP and UDP. As they belong to lower layers, TCP,UDP and IP remain unaffected by the innovations at the application layer. Since the portability layer is the Internet layer, the Link layer only has to interact with the Internet layer. This allows for easier changes to be made to the link layer or to hardware connected to it, allowing innovations in technologies for connectivity such as WiFi, as designers only have to ensure that it complies with the Internet layer and not all the layers above it.

3.2 The End-to-End principle

As mentioned before, the end-to-end principle can guide functionality of the layers. In the previous chapter, two versions of the the principle were discussed : narrow and the broad principle. The narrow principle says that functions should be designed end-to-end between end hosts (destination and source) as they cannot be fully implemented in the lower layers (implemented only in the devices that are in the core of the network). Yet, the narrow principle does allow for exceptions to this rule on a case by case basis, specially if the violation can improve efficiency of the system. The broad version, however, does not allow a violation to this implementation, and recommends that functionality be assigned to a layer only if it is necessary for all clients of that layer. .

In case of Internet, the broad principle has guided the assigning of functionality between the transport layer and the Internet layer. Most significantly, it led to splitting of TCP and IP into different protocols and them being placed in two different layers (Schewick, 2012: 90-93). As mentioned in chapter 2, the end-to-end principle was identified, named and described after the design of the Internet architecture, but was used implicitly by the designers.

How has the broad version led to the splitting of the TCP and IP into different layers? Initially, the designers proposed a single protocol, Internetwork Transmission Control Protocol (Schewick, 2012:91). This protocol combined the aspects of IP and TCP describe above. In particular, it involved the hop-by-hop data transfer (as done by IP) and reliability and error control (as done through TCP). However, designers then realized that not all applications needed a reliable transfer service. One example would be the video telephone services.

In real time digital speech, reliable service is not a prime requirement. The application layer digitizes the analog speech, breaking them down to data packets, and sends them across the network. These packets must then arrive at regular intervals to be converted back into digital speech. If, however, the packets do not arrive when expected, it is not possible to convert them back into analog signals in real time. A reliable delivery, however, can cause delays in the transport service, to ensure that all the packets arrive without damage or loss. This is because a reliable transport protocol asks for re-transmission of any missing data or broken data packets. This delay can completely disrupt a speech reassembly algorithm. In contrast, if some of the packets are missing or damaged, it is easy to cope with for the algorithm and the two users involved in the exchanged may still be able to understand each other even if parts of the speech are missing (or they may increase the reliability by asking each other to repeat speech). It was thus decided that more than one transport protocol should be there and that applications should not be forced to require the services of a reliable service protocol. Consequently, TCP and IP were split into two different protocols (Clark, 1988).

The decision to split TCP and IP is perfectly in line with the broad version of the end-to-end principle.The broad version of the principle only allows very basic and general services to be implemented in the lower layers. The Internet layer as it was finally designed, is based on that principle. The IP is therefore, only aimed at providing a basic building block which every service can use. At the layer above, the transport layer, two protocols were thus, added. One, TCP, which provided for applications that required a

reliable service, and UDP, which provided a application level interface to services that did not require a reliable delivery.

3.3 End-to-End principle and controversies surrounding Internet Policy Guidelines

The end-to-end principle has led to some controversies among the scholars of the Internet. Schewick (2012: 103-108) particularly cites two set of arguments against the use of end-to-end principle in the design of Internet. The first argument goes that the Internet's architecture was not designed according to end-to-end arguments or that it does not offer anything more than what is already included in the layering principle. The second argument is that end-to-end argument is too restrictive for evolution of Internet and that it prohibits some provisions, such as Quality of Service (explained later), which may be necessary in today's Internet.

As can be seen from the discussion provided in this chapter and the previous one, the argument that end-to-end principle offers nothing more than the layering principle is flawed. The initially proposed single transfer protocol, Internetwork Transmission Control Protocol, for example, did not violate the layering principle (lower layers did not have access to the higher layers) but it did violate the end-to-end principle. Further, the argument that the end-to-end argument did not shape Internet's architecture because it was articulated and defined after the design of Internet architecture is also flawed. This argument ignores the fact that the design rules that later came to be known as the " end-to-end arguments " had been implicitly used for several years before they were formally defined. The 1984 paper by Saltzer, Reed and Clark (which is a revision of their 1981 paper) does not claim to invent the principle, but only formally articulate an already used principle.

The second argument against end-to-end principle is that it does not allow for provision for Quality of Service (it is another debate whether Quality of Service should be provided or not, for more detailed discussion on this topic, see section 4.2). A network providing

‘Quality of Service’ can provide different kind of services to different applications. It can, for example, guarantee a minimum bandwidth or a maximum delay, or it can give priority to some data over the other (hence, allowing some applications to run faster than others). For this, the network needs to specify which data packets will get which service quality, and then provide data packets with that quality. These two tasks are constrained, but not completely, to different degrees by the broad version of the end-to-end principle.

A service that offers guaranteed bandwidth or delay, as required for quality of service, cannot be implemented at a higher layer only and has to be implemented at the Internet layer (as the network will need to specify which packets get which quality of service such that the transport can layer can act on it) (Schewick, 2012: 106). Although the broad version usually implies that application specificity be implemented on higher layers, it does not restrict a service to implemented at the lower layers if it cannot be completely and correctly implemented at the end hosts only (or at higher layers). Therefore, it does not prohibit the provision of services to guarantee bandwidth or delay as the these can still be implemented at lower layers along with the higher layers.

With regards to determining which packet gets what service quality, the broad version may provide some constrains. For example, an architecture where the network (that is the layers below application layer) decides which packets to get which service, would be in violation to the broad version of the end-to-end argument (as the lower layers would have to access information from higher layers to make this decision). However, in an architecture where the lower layers are built with a service which can be used by the application layer to decide by itself the different quality of service, is not in violation with the broad version of the end-to-end argument. In such an architecture, the application layer would choose the desired service and communicate that choice to the lower layers. Thus, the argument that the end-to-end argument constrains provision of Quality of Service is flawed.

In sum, the architecture of the Internet has been influenced by two distinct design principles, layering and end-to-end principle. More specifically, Internet architecture is

based on a relaxed layering with a portability layer. The end-to-end principle has two versions to it, the broad and the narrow, and both have influenced the design to some degree. The broad version, in particular, has influenced the design in terms of splitting up of a single transport protocol into two protocols on different layers. While the end-to-end argument has been argued to be too restrictive for a provision of quality of service, these arguments are flawed.

Similar to the debate over the provision of Quality of Service, other debates over the design and use of Internet are also rooted in its architecture, defined through these design principles. For conceptualizing a future Internet architecture, which is a goal this thesis aims to provide a step towards, it is first important to understand these debates in more depth, particularly their relation to design principles or consequently system properties. The discussion in this chapter has laid down the foundation for a more detailed discussion on the contestations over Internet's design and use.

Chapter 4:
The Internet and its Challenges

The original design of the Internet took place more than three decades ago. Since then, it has grown from facilitating a small community of users to one with over 3 billion users. This growth has further been fueled by technical advances such as increase in computing power, increased affordability of computing devices such as the personal computer as well as Internet's support to economic activities. With this growth, Internet has also seen addition of a wide variety of stakeholders such as lawmakers, those involved in the justice system, Internet service providers, as well as range of users who depend on the Internet for either personal or commercial use. These stakeholders often have conflicting interests, leading to contestations over Internet' design, use and regulation. This chapter will discuss some of these debates, which challenge the long term stability of the current Internet. While this will not be an exhaustive list, it will point to some of the main concerns of various actors involved with the Internet. The discussion here also aims to highlight the relation of these challenges with the architecture of the Internet, particularly the design principles presented in the previous chapter. These challenges make a strong case for the need to think about a future Internet architecture as lay the foundation for conceptualizing a future network architecture (and its meta-requirements and corresponding design principles).

4.1 Economic Challenges for Internet Service Providers (ISPs)

In its early years, the Internet ran through the telephone network with AT&T as the provider. In these years, the use of the Internet was mostly restricted to government research laboratories and Universities in the United States. The Internet service lacked a clear model of a market for data services or for the possibility of an industry catering to

such a demand. However, as the potential commercial response to the Internet began to emerge, a process was set in place for commercial use of the Internet.

The Internet Service Providers first emerged in 1989, becoming popular in the 1990s (Clark et al, 2004). The emergence of the ISP in itself throws light on how the network architecture induces an industry, particularly by the point of open interfaces defined by the architecture. Further, the transport and routing protocols of the Internet were kept open, to create a competitive market. The idea was that this would kept the cost for consumers low. However, the full economic implications of this idea were poorly understood at the time.

As the Internet has climbed a commercial mountain, there are two main factors to consider here. The first is that the highly competitive market makes the service provision economically stressful as ISPs have to keep the prices low (Clark et al, 2004). Second, given the open architecture of the Internet, the ISP can only offer a packet delivery service, while needing to invest a huge cost in infrastructure. The incremental cost of a packet delivery service (after the infrastructure has been set up) is virtually zero. This zero pricing makes it hard for the ISPs to pay for their high initial investments in a competitive market, leading to mergers, bankruptcies or monopolies.

In order to avoid monopolization or bankruptcies, ISPs would want to go beyond being a packet delivery service. Higher-level user-visible service can allow ISPs to differentiate their products from other ISPs and also allow users for more choice. However, to provide such services, ISPs will have to implement functionality *in* the network, that is in the Internet or the transport layer. Such an implementation would be a violation of the end-to-end principle and has thus, been resisted by policy makers and scholars (Clark et al, 2004).

4.2 Quality of Service and Network Neutrality

As described in the previous chapter, the Internet provides a ‘best-effort’ data transport.

That is, it makes no commitments regarding delays, jitters or bandwidths. This paradigm had been successful for the Internet as it had to be built to provide a basic service, which could provide a wide range of services. However, as the Internet has matured, with more users willing to pay more for better services, there is an increasing pressure to provide a more controlled experience specially with in demand services such as video streaming. Therefore, there is a demand for the provision of Quality of Service, where different data packets are treated differently.

The provision of Quality of Service can also be a favorable one for the ISPs, who can move beyond a pure data delivery service. However, such a move has been warned against by many scholars as well as policy makers. Governments all over the globe, including United States, France, Germany, India, United Kingdom, have put up machinery to investigate whether a regulatory action should be taken to limit the ability of network providers in interfering with content and services (Schewick, 2012). The opponents of interference by network providers argue that such a move will break down the ‘network neutrality’. The network neutrality rules,proponents of the rule argue, foster innovation in applications on the Internet, protect the ability and freedom of users to choose how they want to use the network, without interference from ISPs, allow for the Internet’s ability to improve and enhance democratic discourse, facilitate political organization and action, as well as provide a decentralized environment for an open participatory social, cultural and political interaction.

However, due to the increasing commercial pressure, and the economic benefits of Quality of Service, some governments have been and will need to reflect deeper on the issue of network neutrality. Going against network neutrality in itself, however, covers a range of potential moves. For example, the regulators would need to decide if the network providers should have the ability to block applications, services or content (this could also have added benefits of security or the disadvantages of censorship). Another question to consider would be if the network providers are allowed to discriminate among various data packets (and hence, favor some applications or services over others), then what this discrimination should be based on. For example, the network provider could discriminate

between two video streaming sites if it own one of them. Another way would be that the network provider can discriminate between categories of applications, for example, by favoring video streaming data traffic over those from e-mails. Thus, there is a challenge here to think about what the goals of the network should be.

4.3 Cybersecurity

On November 2nd 1988, several computers connected to the Internet (about sixty thousand at the time) considerably slowed down, with a number of rogue computer programs, not initiated by original users, demanding the processing time (Zittrain, 2008: 36). This was because of what is credited as the first Internet virus, the Morris worm. The Morris worm, according to the creator Robert Morris, was not created to cause damage but only to gauge the size of the Internet. This reflected the Internet community back then, which had stakeholders with similar goals and trust (Clark et al, 2005). While the Morris worm did not delete any files from the computers it infected and they were able to recover, it did expose the security flaws in the Internet.
A number of non-malicious viruses followed the Morris worm, although a few did cause more damage than intended (Zittrain, 2008: 44). These viruses were basically seen as bad codes by hackers who intended to follow the ethical hacking rules prevalent within the cyber community. Therefore, while there security flaws in the Internet and a number of viruses were created, there were no major security threats to the computers until the early 2000s (Zittrain,2008:44). A critical factor contributing to the absence of malicious code was the absence for business models or financial incentives to develop them. Malware (such as viruses, worms) were created mostly for curiosity or trick users.

However, the scenario has changed drastically in the last decade and a half. The number of Internet users has crossed 3 billion, and therefore, the Internet is no longer a small community built on trust. The Internet had also moved from Universities or offices to homes, where the investment on device security was significantly less. Most importantly, however, the Internet malware had found several business models for itself. Spam e-mails were being used to get money from users though deceit and fraud. Financial information

such as bank and credit card details were being hacked into. Internet's use as a commercial and a critical communication network meant that attacks on particular set of users were attacks on organizations or even countries. A report in 2014 by Intel security group (previously Mcafee), a security software company, estimated the economic loss to the global economy by cyber crime to be around $ 400 Billion (Mcafee, 2014).

Some scholars, such as Zittrain (2008), have argued that the end-to-end principle restricts efforts to make the network secure. The argument is that security mechanism can only be placed on the end hosts and not in the network. Further, security mechanisms placed on end hosts can make end devices more restrictive, and dampen the innovation which has been possible so far. The Apple iPhone(and the app store) is probably the most famous example of such a device. The iPhone is known to provide a stable experience to its users, with applications that hardly ever crash. One of the reasons for Apple's ability to provide a secure experience is by controlling the content available through its distribution platform, App store. Users and third parties can develop applications for the iPhone, but they go through a review process from Apple, thus allowing the company to maintain quality. While this may seem like a good solution to the security dilemma posed by generative technologies, it may have serious repercussions to the pace as well as magnitude of innovation in the future. While Apple accepts applications from third parties, the decision to accept or reject an application is not transparent and rests solely with the company (Hansell, 2008).

The argument that end-to-end arguments restrict security mechanisms to be put in the network, however, is not entirely accurate. For example, Schewick (2012: 367) has argued that if it is true that certain attacks can only be prevented through implementation in the network, then the end-to-end principle does allow for such an implementation. However, it may be that such implementations, while being architecturally correct, may have negative effects for the evolvability of the network. Fr example, firewalls (which violate the broad principle of the end-to-end principle; see 4.5) has had negative effects on application development. Similarly, Network Address Translators (NAT) make it difficult, or in some cases impossible, to deploy new applications or a new transport layer

protocol (Schewick, 2012: 368). In sum, while it may be possible to add security while following the end-to-end principle, it will be a trade-off with the evolvability of the network, which may have harsh consequences for the future of network and innovation on it.

4.4 Emergence of conflicting view and goals

Unlike the small user community prevalent on the Internet before the 90s, the Internet of today has a wide range of stakeholders with often conflicting views and goals. Users, for example, may want to have private conversations. On the other hand, governments want to intercept these conversations in order to prevent security attacks (or for other goals, such as staying in power). Similarly, many users may want to share music and other media with one another, while the right-holders want to prevent it or identify those sharing for legal actions. Many of these conflicts may be intrinsic to the nature of society (or societies as Internet affects various communities differently). The debates over these conflicts, however, can have a critical effect on Internet's future design and regulation, with a potential to impair Internet's ability to foster innovation or just overall communicative utility.

4.5 Extensions to Internet architecture

The challenges mentioned before have often led to extensions being developed to counter them. Many of these extensions go against the original design principles of the Internet and therefore, are a subject of conflicts themselves. In the following paragraphs, I offer a few a brief description of a few of these extensions.

- NAT devices (Network Address Translators) - They were introduced to deal with the exhaustion of IP addresses (a problem that was not considered in the original design)

and are used to map several IP addresses onto one. This also has an added advantage of security enhancement as computers 'behind' the main computer (to which the other computers have been mapped onto) are protected. However, NAT devices go against the initial principle of universal access (that is any two computers on the network can communicate to one another) and move towards a client-server model (where only some computers are reachable globally in the network). While this change may not seem that important, it can have an impact on deployment of new applications on the network as it moves the Internet fundamentally away from a peer-to-peer model (in which any host is theoretically addressable from any other host) towards a client server model (in which only 'selected' hosts are addressable from the global network) (Braden et al, 2000).

- Firewalls - They were introduced to provide security to certain regions of the network. They violate the original design principles (such as the broad principle of end-to-end argument) which consequently has negative consequences for deployment of new applications.

- IPSEC (IP security) - This allows for encryption of transport-layer (TCP and UDP) headers. While IPSEC is compatible with the original architecture, it has led to contestation from network providers as they can no longer 'snoop' at some information related to the data packets being transferred (Braden et al, 2000). This has led to calls for revisiting the original layering model of the Internet architecture.

4.6 Reflecting on the problems of current Internet for lessons for a future Network

The debates over the design, use and regulation of current Internet do not have easy answers. Clark et al (2004) argue that so far the tendency has been to find a short-term fix to these problems, often by technical extensions to the Internet, such as those mentioned

in the previous section. These short term fixes may be pushing the network towards further inconsistent directions, that may lead to further problems, not yet conceivable. Therefore, the aim of this thesis is to set an overarching vision for a future Internet. The underlying question then is, how would a future Internet be designed, given the problems and how they have come about, of the current Internet? To answer this question, we need a deeper reflection of the relations between the problems discussed in this chapter. Such a reflection also paves the way for conceptualizing the required system properties or overarching meta-requirements of a future Internet. The next chapter will therefore, attempt to tie together the problems of current Internet, extract lessons from it and then reflect over the necessary system properties of a future Internet.

Chapter 5:
Reflections for a future Internet

In the previous chapter, I discussed how the current architecture of the Internet has led to the current debates involving various stakeholders, pushing for a demand for a new architecture or changes within the current architecture. However, since the central aim of this thesis is to provide a step towards a future Internet architecture, the answer to the question of how these debates should be settled within the context of current Internet is not particularly relevant. Rather, in this thesis, and particularly this chapter, I aim to reflect on some common underlying conditions that give rise to the multiple challenges for current Internet (as discussed in the previous chapter) such that they are also applicable for any network deployed globally on a large scale. As a clean-slate architecture is unconstrained by the current Internet architecture, it can be designed to avoid problems identical with the current Internet. However, it also provides an opportunity as well as call for a responsibility to cater to challenges of the future . The task in hand is, therefore, to identify and address the underlying conditions that give rise to conflicts in their general, rather than specific technical forms. In this chapter, I will present this underlying conditions along with deriving normative considerations for an architecture should have in order to deal with this conditions. As mentioned earlier, these considerations will then be used to derive desired meta-requirements for the architecture and would pave the way for discussion in the next chapter on design principles that may guide the eventual development of a future Internet architecture.

5.1 Uncertainty : Challenges and Opportunities

What ties all the challenges mentioned heretofore is that they were unanticipated, either in degree or in whole. From a network connecting a few research institutes and Universities within the United States, Internet is now being used by over 3 billion people

globally. Besides the unanticipated growth in number of users, the nature of use has also been largely unpredictable. From a network made to share a few kilobytes of information, Internet now supports some of the biggest businesses on the planet. Along the way, several other technological changes have also contributed o the unpredictability of Internet use. For example, the computing power has increased significantly in the last few decades, allowing uses such as high-frequency trading, which have a critical impact on today's economy. While such unpredictable uses create challenges for design of Internet only indirectly, some other technologies directly affect Internet's design and raise questions about architecture's longevity. For example, Mobility (because of cell phones and tablets particularly), with the movement of an end point, also raises problems such as those of addressing, routing, security and so on (Clark et al, 2004). In sum, the Internet comes with an addition challenge of unpredictability and any future designs of the Internet would have to take that into consideration.

However, the problem of predictability is not unique to the design of Internet. Makridakis and Taleb (2009) argue that unpredictability is present in many complex systems operating in the real world, such as economy, large businesses, climate and so on. These systems, the writers argue, are different from situations such as a coin toss or celestial bodies, where prediction is easier and justified. In the case of celestial bodies, whose path can be predicted with great accuracy, the predictions are based on identification of patterns which can be expressed in mathematical models, and where the sources of errors are minimal. Similarly, in case of a coin toss, probability mathematics can be used to easily predict that for a large number of instances, the number of heads will be equal to number of tails. It is true that for a small number of instances, for example, say for 10 coin tosses, the outcome is still uncertain (as it can be anywhere between 0 to 10). Yet, the degree of uncertainty is still known in this case.

Complex systems, such as Internet, however, differ from these cases of easy prediction, in at least three respects. First, the errors are not independent of one another, and can multiply to produce large unknown variances. The popularly known, 'Butterfly effect' is an example of such a case, according to which a small change of state in a nonlinear

dynamic system can lead to a large difference in a later state (Lorenz, 2000). Second, complex systems can have highly unlikely events which though have very small probabilities are difficult to predict because these probabilities are unknown (Taleb, 2007). Taleb (2007) terms these events as 'black swan' events. Some examples of such events, given by Taleb, are World War 1, September 11, 2001 attacks, and the rise of internet and particularly Google. Another example which clearly shows the limits of prediction, and occurred after Taleb's book, is that of Fukushima nuclear reactor, which had a catastrophic failure after a Tsunami. The designers of Fukushima had built the reactor to withstand the worst earthquake in history, without imagining worse (Hayashi et al, 2011). While the probability of an earthquake being the worst in recorded in history is very small, it is still unknown. Therefore, and third, complex systems are harder to subject to prediction models because probability of many outcomes is unobservable (or uncalculable) (Makridakis and Taleb, 2009). .

The challenge before designers or policy makers then is : How should decisions be made in low or unknown levels of predictability?

A concept that has emerged out of research on anticipation is that of resilience. O'Malley (2010) cites various fields where the concept of resilience has been added to anticipatory approaches to uncertainty and potentially traumatic possibilities. Resilience differs from the traditional approaches to uncertainty such as risk and preparedness. While probabilistic risk is a technique of harm-minimization, preparedness involves creation of mechanisms for coping with imagined harms. Resilience, however, is a more encompassing and systematic approach to anticipation as well as toleration of disturbances that occur out of uncertainties. Lentzos and Rose (2009), for example, write:

..resilience implies a systematic, widespread, organizational, structural and personal strengthening of subjective and material arrangements so as to be better able to anticipate and tolerate disturbances in complex worlds without collapse, to withstand shocks, and to rebuild as necessary . . . a logic of resiliency would aspire to create a subjective and systematic state to enable each and all to

live freely and with confidence in a world of potential risks (Lentzos and Rose,2009:243)

Yet, this account of resilience is essentially reactive (in the sense that it still relies on prediction and anticipation), which may be hard to do because of the problems in prediction present in complex systems as described above. A resilient architecture may not be able to survive, let alone thrive, in conditions that are completely different, and unpredictably so, from those that existed when the architecture was designed. Some discourses such as those of Brooks and Goldstein (2006) have therefore, presented a more proactive approach, where the idea of resilience is more than just withstanding shocks, and aim towards thriving on chaos. Taleb (2012) has argued for a similar approach and uses the neologism “antifragile”, to refer to systems that gain or benefit from stress and randomness. According to Taleb, antifragility, the ability of a system to benefit from stress and chaos, is not only different but better than resilience when it comes to dynamic and complex systems. For such systems, volatility is unavoidable and hard to be indifferent to. Therefore,what one needs is a system that takes into account volatility and tries to thrive through it. What is then the antifragile approach to uncertainty and unpredictability?

The first step in the process, according to Taleb, is to do away with the notion that unpredictability is always risky. Taleb (2012) argues that modern human societies are deeply rooted in a Newtonian view, a paradigm of determinism and control, where anything that is unpredictable, unexpected or outside prescribed norms, is deemed as a risk or a threat to the system. This deterministic view of the world relies on predicting the system, however, is doomed due to the dynamic and complex nature of most systems along with what Taleb calls ‘black Swan events’. On the contrary, unpredictability and uncertainty can lead to advantageous opportunities, precisely due to unlikely events.

To drive his point home, about the opportunistic advantages of uncertainty, Taleb gives several examples. Several of these are found in nature. For example, forest fires, an uncertain event often looked upon as a tragedy, is actually beneficial, when occurring in natural cycles, as it clears the forest off the highly flammable material. Suppressing such

fires can lead to short-term gains, however, in the long run it leads to devastating, even if fewer, fires. Similarly, natural evolution makes use of the randomness in genetic mutation, as well as in environmental conditions, leading to species better adapted to particular environments (and arguably to more complex forms of life in the long term). Taleb (2012) argues that these strategies can be applied in real life by moving to “non-predictive decision making”. As a financial investor, Taleb used this approach by applying a ‘barbell strategy’ (used by other investors too). In Taleb’s barbell strategy (as barbell strategy can be of different types), an investor invests most of his assets (such as 90%) in extremely safe investments which lead to medium to low returns, while the remaining in extremely risky investments which can lead to extremely high returns. In this case, the investor faces low risk but a potential (albeit probabilistically low) for very high returns. In most circumstances, such an investor makes low but guaranteed gains,but in case of uncertain or black swan events, she/he earns a very high amount (because of the extremely risky investment).

Uncertainty also plays, and has played, an important role in Internet history as well. As discussed in the previous chapters, the design principles of Internet architecture, such as the end-to-end principle, allow it to be evolvable and leave room for development of applications not imagined or predicted during the original design. While it is true that other independent events such as the increase in computing powers and affordability of computing technologies (such as the personal computer) have greatly helped in providing the ecosystem for innovative applications, it would not have been possible without the property of evolvability being built into the Internet architecture. Further, this room for evolvability has been critical in the growth of Internet as well as its rise as the most important communication network in modern society. This point can be made more clear by contrasting Internet with other networks prevalent at a time.

Consider CompuServe, one of the first commercial network services that dominated before the Internet took over in the 1990s. CompuServe allowed its users to do a variety of tasks such as send emails to to each other, text chat, read news, play games,etc. However, CompuServe users could not add content to it on their own (Banks et al, 2012).

CompuServe did enter into some agreements with third party developers, however, its functionality remained more or less the same throughout 1980s and 1990s, the period in which it was quite popular (ibid). The main reason of Internet's takeover, in terms of popularity, over CompuServe in the 1990s, was the large amount of content added by its users, attracting other users who either wanted to enjoy this content or add their own (Zittrain, 2008).

The capacity of Internet to facilitate innovation, because of the evolvability built into its design, is a critical aspect of the modern economic growth. It is for this reason that scholars have argued to preserve the 'network neutrality' or against interference from ISPs in a way that goes against the end-to-end principle (which plays an important role in providing evolvability as a system property). Thus, any future design of the Internet must consider two important aspects: a.) predicting future outcomes and use characteristics of a network like Internet is very difficult, if not impossible, and b.) this unpredictability can be used as an opportunity as the network can lead to further innovations.

5.2 Stakeholder Conflicts, Values and Value Sensitive Design

The challenges discussed in chapter 4 exist as conflicts between various stakeholders involved in design, use and regulation of the Internet. The ISPs, for example, would want to be able to provide more services than just data carriage, while governments maybe against this as this may lead to an Internet restrictive of innovations. Users may want to keep their conversations private while governments may want to intercept these in order to protect against security attacks. Corporations would want more tools such as Firewalls, which increase the security of the network, while those advocating for more innovation and uniform accessibility would be opposed to it. These stakeholder conflicts present themselves as a difficult challenge for the engineers designing the Internet, which is exacerbated by the presence of uncertainty and unpredictability of outcomes as described in the previous section. This challenge forces the designers to think about the Internet's design differently. In this section I suggest that a new design for the Internet should cater

explicitly to the conflicts between the various stakeholders, regarding it as one of the central tenets on the design process.

A designer's responsibility to balance conflicting considerations, however, is not unique to the Internet. In many cases, design is often more than just meeting the prescribed functionality and specifications. Vinck (2003) presents several case studies that highlight the importance of balancing conflicts during the design engineering process. For example, building a roadway system may require the designer to collect and analyze data about the various demographic characteristics, such as the kind of vehicles owned by the population, the frequency of travel, degree of importance of various nodes (cities or towns) along the way, the potential placement of roadside facilities, and so on. The choices made by the designer are likely to have a huge impact on the community and its future, forcing the community to adjust its lifestyle according to the artifact once built.

In this sense, designers should be aware that the artifacts they design are not morally neutral. This claim is in line with the concept of embedded values in computer ethics (Brey, 2010). This concept holds that computer systems and softwares are not morally neutral but rather demote or promote particular norms and moral values. Computer systems can, for example, go against realization of values of privacy, property rights, freedom of information, or they can be supportive of these values (Brey, 2010). The task in hand for Engineers and designers is to design technologies in a way that they promote values that are desirable. Value- Sensitive Design (VSD), a well known approach, aims to provide a way to realize this goal of designing technologies such that they account for desirable human and social values (Friedman et al, 2013).

VSD's central aim is to incorporate ethics into the design process through its tripartite methodology (Friedman et al, 2013:71-74). This tripartite methodology consists of a conceptual investigation, an empirical investigation and a technological investigation.The conceptual investigation involves identification of stakeholders as well as the relevant values through a philosophical reflection. A conceptual investigation, for example, would involve deciding between two or more competing values that maybe potentially

implicated. However, it may not always be possible to take such a decision conceptually, and may require an informed empirical analysis of the artifact within its use context. This is taken up under empirical investigations. Empirical investigations can also be used to evaluate the success of a particular design as well as response of stakeholders with that design. Technical investigations focus on the relationship between usability and values (there may be a potential trade-off , for example). Like empirical investigations, technical investigations are also done through testing in use contexts. However, the focus of technical investigations is on the technology itself rather than the individuals or groups using it. While the three parts of the methodology are listed separately, they often overlap in practice.

However, a number of factors hinder the use of VSD, as described, to be used for design of Internet or its architecture as a whole. For example, what separates Internet from artifacts considered in case studies by Vinck (2003) or examples used by Friedman et al (2013) (such as a roadway system or use of HDTV display technology in office environment) is that, in latter cases, the conflict is more likely to be concretized (to potential satisfaction of some and dissatisfaction of others) before or during the design process, while for the Internet, conflicts are more likely emerge and reinvent themselves during ‘run time’. Similarly, in case of Internet, due to the rapid of change and innovation, new stakeholders may emerge while the old ones may re-evaluate their positions (explained later in this section). Further, as Internet is embedded in a range of societies, with differing moral norms, it makes it very difficult, if not impossible, for a value set to be chosen that is satisfactory across all cultures. Yet, this does not mean that the insights provided by VSD are not valuable. As will be discussed in following paragraphs as well as section 6.2 of this thesis, designers do need to deliberate upon the ethical implications of their designs, particularly how they affect various stakeholders, as well as be diligent in identification of present and potential stakeholders as far as possible. The following paragraphs will, however, discuss in further detail how Internet makes for a unique case that needs to go beyond VSD and where designers need to leave room for unpredictable outcomes as well as for a plural set of values to be embodied within Internet in the future as well as in different ‘regions’ of the network (see section 5.4). I will also reiterate the

implications for design, given the inherent uncertainty and complexity of a technology like the Internet, and the insufficiency of VSD in providing normative considerations for designers to cater to ethical challenges raised by the technology in section 5.5.

In the essay “Technology is society made durable”, Latour (1990) argues that technology stabilizes the society in a particular form. Rather than an external framework in which technologies, and its users, are embedded, society is the result of how the actors within it are aligned. In the actor network theory, defended by Callon (1987) and Latour, these actors include humans but also non-humans in the form of technology. Callon (1990) further argues that these actor networks, by virtue of providing stability, restrict the society to certain directions, or creating ‘lock-ins’. Therefore, paradoxically, while technology has a revolutionary potential, it can also limit our ability to change things or break down power structures.

This tendency of technology to lock-in can be detrimental for a variety of reasons. For example, in *The Innovator’s Dilemma,* Christensen (1997) describes incumbent businesses get locked into a rigid system, making change harder to come by. Such businesses then miss out on the radical disruption, which is carried out by other businesses, either new or small enough to not be restricted in a way the big businesses are. One of the reasons given by Christensen for the rigidity of big businesses is that they rely too heavily on predicting customer needs based on past data, and miss out on disruptive innovation that can cater to or even create future needs. In this way, the downfall of big companies occurs because of a lack of openness to change (particularly of a radical kind). In case of Internet, Lessig (1999) describes a similar process, albeit from a different perspective. For Lessig, the Internet supports a lack of centralized control and an openness to change. However, law makers, according to Lessig, have been pushing the Internet to a new set of values, such as centralized control, regulation and loss of freedom. As happens to rigid businesses, Lessig (2001) argues that this lack of openness to change, or the new rigidity being introduced to the Internet through centralized control and regulations, will destroy the innovative character of the Internet as well as the progression of new ideas in contemporary society.

Like Lessig, Clark(2005) has argued that the Internet demonstrates a tendency for openness to change (or evolvability, as it has been called in this thesis), and that this tendency needs to be preserved. This openness to change, which is also the cause for emergence of 'run-time' conflicts, according to Clark, manifests itself through a variety of different forms. First, new applications are constantly developed, changing usage behavior with them. This can further cause a need for design and re-design of standards and application protocols. Movements such as the open source movement, has enabled users to configure their own technologies, further adding to changes in use of the Internet. Although it is not always possible, users can sometimes have the choice to choose among service providers. Further, these changes open up the gates for entry of new actors, much more frequently than it is possible in other technologies. An example is the case of creation of Voice over IP (VoIP), which brings the telephone network (and the actors within it) into the conflict over Internet (Clark, 2005). This feature of letting new actors in the network, makes the management of conflicts a heterogeneous process. Unlike some technologies, where the conflicts are managed largely by the designer, Internet's conflicts are not restricted to its engineering team (the Internet Engineering Task Force or IETF). Instead, it would require a whole set of stakeholders such as lawyers, lobbyists, legislators, law enforcers, hackers, and so on. Further, as a global network, Internet is present in many different societies, with contrasting beliefs, leaving room for a further set of conflicts to emerge and re-emerge. As mentioned earlier, these factors, particularly emergence of new stakeholders or new stakeholder perspectives during 'run time' as well as the heterogeneity of a large scale network, requires a design approach that goes beyond traditional Value-Sensitive design methodology.

The argument here is not that the Internet cannot be made into a stable technology, but that such stability is made extremely difficult because of Internet's complexity and the diversity of the stakeholders involved. Moreover, such stability is more likely to be harmful, as has been argued by Lessig, than useful by virtue of restricting its ability to evolve as well as by concentrating power in a few hands. While a case can be made for a scenario, where designers can tilt the balance towards one stakeholder, and even reap

some short term benefits, the inherent uncertainty and low (and of unknown degree) predictability is likely to cause harmful effects in the long run (through ‘black swan events, for example). Thus, given Internet’s characteristics to resist stability, it seems wise to approach its design by explicitly catering to such capabilities. Yet, while Internet has been accommodating to new actors, this need not always be the case, and that the theory postulating long-term harmful effects, as has been seen in other complex systems such as economy, may not hold true for the case of the Internet.

However, there is a strong case to be made for approaching the design of a future Internet in a way that it caters to an open and diverse world. As already argued, Internet is a global technology shaping many different societies. Further, as an information network, its users are significantly dependent upon it to not only gain information but also to voice their own opinions or even preserve their cultural information. On a similar note, Van der Velden (2007) argues that open and flexible design in Information and communication technologies is particularly important because of their relation to the ways of knowing and being in the world. The designers of a future Internet therefore, have the ethical agency, or the capability to act responsibly towards the ‘other’, which would include all those stakeholders, who may not be aligned with the preferences of the designers themselves. A design that allows such conflicts to be represented fairly, in the ‘run time’ rather than solving them technical means even before a struggle between stakeholders ensues, is ethically desirable, particularly in a world that strives to be more democratic and inclusive.

The flexibility to allow for a variable set of social values becomes even more crucial for matters involving law. Technical designs can be done in a way, such that they blunt the tools of enforcement, and therefore embed a bias against law (or state) in the design. Yet, this is not a desirable strategy. A state (or a regulator) left with a blunt tool against a network that goes against the interest of that state, may opt for a total disconnection from the network. An example of this is visible in the case of conflict between Google and China. As Google refused to comply with the Chinese law, the Chinese government responded by a threat to revoke Google’s license to operate in China altogether (Brown et

al, 2012). This has left the Chinese users with Baidu, which is highly regulated. King et al. (2013) argue that the greatest harm in Chinese Internet policy is not so much from censorship of information against state actions, but from the restrictions towards formation of a collective, that can perform actions. It can be thus argued, that the Chinese people may have been better served with a regulated Google than a non-operational one. Similarly, several other countries, such as Iran, Turkey, North Korea, Pakistan have opted to temporarily or permanently ban YouTube and Facebook as it does not comply with their own laws (Bender, 2015). While the morality of government censorship is debatable, what these examples show is that technology is not the only tool in stakeholder conflicts. Treating technology as such can lead to more harm than benefits.

5.3 Network can benefit from added 'Intelligence'

As argued in the previous chapters, Internet is oblivious to the data being sent through it. This 'transparency' (what goes in, goes out), which is a function of keeping the core of the network 'dumb' (as functions are not added to the core, in line with the end-to-end principle), is a critical strength of the network (Braden et al, 2000 :15). Yet, the lack of intelligence in the network also prevents addition of features that could be helpful for Internet users. For example, in case of a failure, the user is left frustrated without much idea of why the failure occurred. Similarly, the network operators have very little interaction with the core of the network, and its overall functioning. Information about network's characteristics and functioning, for example, about data flow along certain routes, congestion stats, data use by application, data use by region, could be helpful in making the network better and also help in deciding future policies regarding network regulation.

The idea here is to suggest that the network should have an information loop, where it is collecting information about itself, which can be used further, either manually or automatically (or both), to improve its functioning. This information loop can also be

helpful in catering to uncertainty as well as managing stakeholder conflicts. The importance of such an information loops can be seen through abundant examples present in other technologies as well as in nature, emphasized within the discipline of *systems thinking*. Systems thinking advocates for a perspective where certain entities are seen as belonging to a complex system, made up of interdependent parts that give rise to collective behaviors (which are often emergent in nature as the constituent parts do not exhibit these properties themselves). The size and the boundaries of this system may depend on the purpose or method of analysis for conceptualizing a system. For example, it is possible to conceptualize the whole earth as a system with elements such as air flows, water bodies, biodiversity, and so on. On the other hand, a smaller ecosystem confined within a small area may also been seen as a system. Similarly, an organization can be seen as a system consisting of people, structures and processes.Within systems thinking, such an information loop is known as feedback (Astrom and Murray, 2010).

Feedback is a way for a system to respond to external stimuli. This external stimulus could also be another system, in which case feedback makes the two systems coupled. A simple example of feedback is a thermostat, used to control the temperature of the room. The thermostat receives information about the temperature of the room as feedback and then operates it heating element to get the temperature to a desired value. Similarly, biological systems make use of feedback in a number of ways as well,with a diverse range that stretches from molecules to cells to organisms to ecosystems. An example is the regulation of glucose in the bloodstream through the production of glucagon and insulin (Astrom and Murray, 2010: 2). The body attempts to maintain the concentration of glucose, which is used by the cells of the body to produce energy. When glucose levels rise (such as after a meal), the hormone insulin is released, causing the body to store excess glucose in the liver. When glucose levels are low, the pancreas releases the hormone glucagon, which has the opposite effect.

Feedback can add many interesting properties to a system, including the capacity to be resilient to external influences. For example, the glucagon-insulin system is able to maintain glucose even after a heavy meal (Astrom and Murray, 2010:3). Yet, resilience

does not always mean that the system should remain in the same state. Rather, resilience here implies that the system is able to preserve those properties that are extremely important to its functioning. Therefore, it is possible that a system uses its feedback to change and evolve, while still maintaining its most important features.

In case of Internet, feedback about network's functionality can play a similar role. It can allow the network to evolve, based on the information collected, while still keeping intact, the critical system properties. One of those system properties, as has been argued before, can be the evolvability itself. The challenge before the designers would therefore be to keep the features which have made Internet a success and still add intelligence to the network, in the form of feedback information loops, which can help making better decisions for the network, manually or automatically. Adding intelligence can, thus, help towards making the network antifragile, with the capacity to thrive in chaos or unpredictability. An intelligent network, with the data about its current and past environment fit, that is, its use characteristics, would be better equipped to make decisions about losing or substitution of one or more of its features. This is particularly important under conditions where the network faces new conditions (for example, new applications or new hardware support). While a non-intelligent network may be able to resist such new conditions (in case of a resilient network, for example), an intelligent network may be able to adapt to these conditions with the possibility of performing better than before, without losing its critical features.

The difficulty in adding such intelligence can be understood in the light of discussion on end-to-end principle presented in chapter 3. Adding features to the core of network can make the network less conducive to a wide variety of applications (as it will not be general enough to support them). Such a problem can however, be addressed, by keeping the intelligence at the edges of the network, while making an information loop where the core receives information from the edges. Such a solution does not violate the end-to-end principle or the layering principle. In this case, the core of the network only responds to the information given to it by the application (on the edge) and would only respond to the needs of that particular application. A possible design strategy to implement such a

solution will be discussed in the next chapter.

5.4 Architecture should be minimally defined

A final crucial point that comes through the discussion presented in the previous three sections of this chapter as well as from the challenges that current Internet faces, is that there will be very few requirements that will be truly global, that is, they are unlikely to be applicable with the same importance everywhere within the network. A new network architecture therefore, faces a difficulty in developing a single order list of requirements as was done for the current Internet. This makes the necessity of having meta-requirements, or system properties, that are extremely critical to the network even more important. These meta-requirements can then be translated into over-arching design principles, defining a minimal set of globally agreed mechanisms, while allowing the network to have different set of features based on where it is used, and how it is being used.

The challenge for a future Internet is thus, to be able to acknowledge the variability of requirements in different ‘regions’ of the network, while still preserving critical characteristics such as ease of adding new applications to the network. In the current Internet, this variability has been achieved through a number of devices and technologies that perform functions other than those available in the network. The examples of such devices include Firewalls (for added security) and NAT devices (to enable multiple end hosts to share the same address) as discussed in the previous chapter.However, these devices operate in lower network layers and hence, reduce the generality of the network for new applications, making it less conducive to innovation, which has been one of the main drivers for Internet’s growth.

5.5 Moving Beyond Value Sensitive Design

In section 5.2 I discussed Value Sensitive Design, one of the most reviewed approaches,

pertaining to values, for technological design. The motivation for this approach comes from the recognition that technological innovation should focus more than just on its functionality, and that technologies are not morally neutral but rather have political and moral impacts on their users as well as the environment they are used in. I also discussed the methodology of VSD, which is tripartite in its setting, with conceptual, empirical and technical investigations to be carried out. While VSD gives important insights and establishes the need for a proactive approach to ethical design of technologies, I argue that it is not a suitable approach for design of a complex network like the Internet. The reasons for its unsuitability have much to do with the discussions provided in sections 5.1-5.4 of this chapter. I will discuss them in more detail in this section as well as set out the implications for design given this unsuitability of VSD.

The first challenge in using VSD for a network like Internet comes in the identification of stakeholders and their views on the technology, which is an important part of the conceptual investigations within VSD methodology (Friedman et al, 2013). As argued in section 5.2, stakeholders can emerge years after the design of a network like Internet, particularly as innovations on the network lead to new uses. One example of such a case, already discussed above is the creation of Voice over IP (VOIP) technology, which brought in the telephone operators as a stakeholder set to lose out from this technology. Similarly, new uses may also lead stakeholders to change their views of the technology. For example, the early users of the technology felt no particular need for security on Internet, while new uses such as banking as well as the proliferation of computer worms, have led to many users to be concerned about security of their end devices (Zittrain, 2008). These reasons not only suggest that it is difficult to identify stakeholders during the design process but also point to the threat in defining stakeholders during the design process, as the design that follows after such an identification may not allow new stakeholders to join in, and be partial to initially identified stakeholders who may also create hindrances in innovation on the network.

The second major challenge is using VSD for design of a complex network like the

Internet comes through the very concept of ‘values’ as defined within VSD, which is central to it. Friedman et al (2013) state that VSD prescribes an interactional view of values, that is, values are neither seen as inscribed within the technology nor simply transmitted by social forces. Rather, they state, values, within VSD, are seen to come about in the use of technology. This interactional position is supported by the argument that technologies can often have multiple uses as well as the fact they can change considerably over time. For example, a screwdriver is not just used for turning screws but also as a cutting device, a tool for extracting weeds or as a poker (Friedman et al, 2013: 86). In this sense, the values to be conceptualized for a technology should consider the use contexts. However, this is particularly hard to do for a network like Internet. As we have seen from Internet’s history, it has gone from a technology meant for communication within the US defense department to one being used globally for a very diverse set of reasons ranging from shopping to banking to sharing one’s voice globally. In such a scenario, it becomes very difficult for a designer to identify which values are necessary for Internet’s overall design. Indeed, even values like privacy, which are much debated upon within Internet scholarship, are not universally important over the Internet.

Manders-Huits (2011) has given a similar critique of VSD and the concept of values within it, particularly stating the definitions of values remain abstract and therefore, do not offer a practical guidance for their implementation, particularly in cases where different stakeholders may interpret these abstract definitions differently. This is particularly relevant for the case of a global network like the Internet, which is embedded into a wide range of societies with many cultural and social differences. Manders-Huits (2011) also gives examples of how values and their definitions are relative and may even mean opposite, if not different things. For example, while the concept of human dignity may be found in many societies, its definition often changes with regards to other concepts such as the relative positions of different genders and races. As we have seen through human history, not all members within the society may be granted equal rights to their dignity. This also leads back to the discussion presented in section 5.4 and the need for a global and complex like the Internet to allow for heterogeneity in its design.

The central problems in applying VSD methodology to design of a future Internet then come in identifying universal values as well as stakeholders, which as the above discussion points out, is extremely difficult, if not impossible. However, the absence of universal values and a stable set of stakeholders can in itself be seen as an overarching value, of designing for an open and diverse world, to be incorporated in a complex and global like the Internet. The importance of this value is represented by the arguments given in sections 5.1-5.4. For a future Internet, therefore, it is critically important it not only functions as an information network but also that it caters to the uncertainty and diversity of the world, in an 'intelligent' way (see section 5.3). This leads us to four different meta-requirements for the Future Internet architecture, presented in the next section.

5.6 Meta-requirements for a future Internet

In this chapter I presented four major issues that underlie the challenges of the Internet (sections 5.1 to 5.4). The idea was to suggest that these underlying issues can give us normative considerations for a future Internet. These normative considerations can then be used to derive meta-requirements and consequently design principles for a future Internet. Based on the discussion presented in this chapter, I present four meta-requirements for a future Internet:

1. Evolvability - As discussed in section 5.1, the inherent uncertainty of the internet requires an antifragile approach, which is built upon an openness for change. Evolvability, thus, is a critical meta-requirement for a system like the Internet which is subjected to inherent unpredictability and which can benefit from this uncertainty through facilitation of innovation.

2. Distributed control to relevant stakeholders - Internet is not only embedded into a diverse world, it is critically in maintaining this diversity as well. Further, bias of designers, who are also limited in their knowledge of future events, can only be countered by providing distributed control to all relevant stakeholders.

3. Intelligence within the network - An intelligent network, which can collect and integrate information about its own use can be extremely useful in diagnosis of faults, providing directions for future evolution as well as aid in decision making by the stakeholders involved.

4. Heterogeneity - A network built for heterogeneity underlies the understanding that there are few requirements that can be truly global. While evolvability provides the network flexibility over time, heterogeneity provides the network flexibility over space. It also ensures that different societies have a chance to express their own values through the network without being dominated by another through technology.

These meta-requirements provide an overarching vision for a future Internet architecture. Yet, meta-requirements cannot be translated directly into architecture. While design principles can be a helpful tool in achieving such a translation, a detailed account of design principles for a Future Internet architecture is beyond the scope of this author and this thesis. As Design principles can facilitate (or constrain) socio-economic goals, such a task should be down through a multi-disciplinary effort with a team of technical as well as non-technical experts. I will however, attempt to present design principles in their general form (that is not as specific technical implementations) that can be translated from these meta-requirements. These design principles, as presented in the next chapter, can serve as a foundation for future research towards a more concrete future Internet architecture.

Chapter 6:
Recommendations for Future Internet

The last chapter highlighted some of the underlying issues that can be derived out of the challenges faced by the current Internet. From these issues, I laid out some normative considerations in the form of meta-requirements for a future Internet. In this chapter, I aim to present some design principles that can fulfill the meta-requirements derived previously. Since multiple design principles can cater to same meta-requirements, the architecture that results from these design principles will depend on the specific technical implementation of these design principles. However, the purpose of this thesis is not to provide specific technical implementations of these design principles, as this is beyond the scope of this thesis as well as expertise of this author. Any project aimed at development of technically implemented future Internet should involve a multi-disciplinary teams of experts, whose expertise ranges from technical fields to non-technical fields such as economics and law. Such a project is likely to be iterative in its process and would require the efforts of a multi-disciplinary team in not just the design phase but in tests and pilot phases as well. In this Chapter, however, I will provide guidelines for such a project to be undertaken by suggesting design principles on a general level as well as steps that can be taken to evaluate the specific implementation of these design principles. This will help in moving towards the goal of a future network than can support the socio-economic goals of a diverse world.

6.1 Modularity

As explained in chapter 2, modularity refers to a design in which components of the systems are made highly independent (or loosely coupled). This independence of modules from one another gives the freedom to the designers to not only design each of these components separately but also allows easier changes to the system. In the previous

chapter I argued that the ease of change is critical to a complex system like the Internet, because of the unpredictability involved in the outcomes of use as well as the nature of use of the network. Ease of change provides a complex system to take advantage of the opportunity provided by uncertainty, by adapting itself such that it thrives rather than collapses under incumbent conditions. Further, ease of change within a system also allows a system to provide room for various stakeholders to engage with the design, rather than a condition where design is stable and excludes some stakeholders over others concretizing the power relations (more on this in the next section). Most importantly, however, ease of change is critical to retain the innovative character of a network like the Internet, which is particularly important in the economic landscape we live in (and are likely to stay in for atleast a few more decades). This can be better understood with a comparative analysis of a modular architecture with a non-modular, or an integrated, architecture.

An important factor involved in innovation, particularly in the current economic landscape, is the cost of innovation. The lower the costs, the lower the threshold for innovation to take place, and hence, more the possibility for innovation. Costs of innovation can itself be further divided into three parts: cost of realizing an innovation cost of production and distribution, and cost for deployment of innovation (Van Schewick, 2010:118). The first part, the cost of realizing an innovation can be further divided into two categories: cost of developing as well as testing the innovation itself, that is the component on which the innovation(s) takes place and the cost of developing and testing the changes, that are necessary, to other components of the system which enable the innovation. For example, if Apple releases a new version of its mobile operating system iOS, by changing a few parts of the older version, it might also require changing some of the applications so that they are able to perform in the new version of iOS.

For an integrated architecture the possibility of making changes to one component such that other components are not affected is less likely. It is very likely that changing one design condition may lead to a cascade of changes to be made to other components.

Further, integrated architecture also makes it hard to know in advance which components might be affected by such a change, or even which components need to be changed in case the new changes result into a failure after testing. Innovations in integrated architectures is therefore, difficult, costly as well as time consuming.

Comparatively, modularity reduces the interdependencies between components, enabling a significant amount of changes that can be done without requiring a change in other components of the system. For example, for personal computers, the screen technology has changed considerably over the years, moving from the early Cathode ray Tubes to LCD and then to LED, without requiring changes in other components (although they have also changed independently). This reduces the cost of, what can be called as system adaptation, lowering the threshold for innovation within large or complex systems.

The value of innovation, and hence of modular architectures, can be further understood by using the 'options theory' (Van Schewick, 2010:122). The options theory provides an analytical framework for assessing alternative courses of action under conditions of uncertainty. The origin of the framework lies in the subject of finance. For example, a financial option may give someone a right to buy a particular stock at a particular price within a specific period of time (that is, it comes with an expiration date). If at the date of expiration, or before it, the price of the value of the stock is higher than the value at which the person has the option to buy it, she/he may buy the stock and therefore, get a payoff equal to the difference between the two values. In case the value of the stock is lower than the value in the option, the owner can choose to do nothing, as she/he does not have an obligation to buy it (but only a right). In this way, options can be a way to deal with uncertainty. In fact, the value of an option is more under greater uncertainty. Under high uncertainty, the likelihood that the value of the stock will go very high or very low is greater than in relatively stable conditions. Through options, the owner introduces an asymmetry between her/his potential for profits and losses (the potential for profit being high while the potential for loss does not increase).

Similar to financial options, 'real options' provide a way to deal with uncertainty. Unlike

financial options, real options are not bought but come along with investment decisions that open alternative courses of action where gains and losses are asymmetric (with potential for gains dominating the potential for losses). Innovation on either architecture, whether modular or integrated, comes with uncertainty. Particularly in large or complex systems, it is difficult to predict in advance whether the innovation will be valuable or not (or atleast how valuable it will be). Further, innovation on architecture can be itself seen as an option, as it is not obligatory to introduce this innovation in the market. The innovator can choose to introduce it to the market if the result is better than the old system, or not if it turns to be less useful.

For large or complex systems then, a more suitable architecture is one for which the cost of acquiring an option (that is, creating alternative courses of action in a way that the potential gains dominate potential losses) is lower. For if the value of acquiring an option is more than the value of innovation, innovators will lose the incentive to innovate on the system. For a modular architecture, as it is easier to produce changes as well as test them, the cost of acquiring the option is generally lower than an integrated architecture. Further, while a modular architecture provides a range of options, integrated architectures provide an option in the form of replacing the system with another. This is because for an integrated architecture, changes to individual components are not feasible, and the innovation needs to be done at a system level. In contrast, modular architectures can be changed at the level of modules, each of which (and their combinations) can produce a range of options. The asymmetry in gains and losses (where gains are more likely than a loss) is thus, much more in a modular architecture than in an integrated architecture because of two main reasons: a.) It is easier to acquire an option in modular architectures (and hence easier to create such asymmetry) and b.) modular architectures can produce a number of such asymmetries (as a range of options can be acquired on a modular architecture).

Yet, the task for the designers for a future network will not be as simple as choosing modular over an integrated architecture. The ease of acquiring an option, and hence of creating asymmetries for innovation, can be different for different modular architectures.

This can also be seen in the current Internet architecture. Within the current Internet architecture, which is modular, violation of end-to-end principle makes it difficult to acquire options as compared to following the principle. For example, if the IP layer of the Internet is modified so as to support reliable data transfer to support applications that require it (and increase their performance), it will be a violation of the end-to-end principle. Further, this violation decreases the range of options available for innovators. For example, an architecture with reliable data transfer built into it, will lead to some delays in data transfer (to ensure reliability), and hence, will be unusable for Internet telephony.

Similarly, if the current Internet architecture was not built with the IP layer as the portability layer, it may have provided less options for the innovators to acquire. As argued in chapter 3, the portability layer of the Internet, that is the IP layer, is used by all the layers above it, while the link layer (which includes hardware such as routers) lies below the portability layer. In the current architecture, changes in the link layer can be done such that the IP layer does not need changing, and hence, the above layers as well can remain unchanged. However, if the IP layer was not the portability layer, innovations in link layer would have required a change in application and transport layer as well, thus, reducing the ease of acquiring, or innovation.

Hence, the task for designers of the future Internet architecture would be to conceptualize multiple modular architectures and then analyze them using the options theory to evaluate which of those designs best increases the ease of innovation.

6.2 Design for 'Tussle'

In the last chapter I argued that Internet opens up conflicts between different stakeholders during 'run time' regarding Internet's design, usage or regulation. Internet exhibits a tendency to be unstable, and further, that this tendency has its merits. Particularly, stability can cause concretization of power structures, and exclude some stakeholders

over the others. Such an exclusion can be harmful for innovative capacity of the complex systems such as the Internet, as well as restrict the different ways of knowing and being in the world. A network architecture that tries to incorporate, rather than shut down, socioeconomic stresses as a result of conflicts between stakeholders, is better suited to socioeconomic goals such as cognitive justice or support for multiple business models over one which the designer thought to be the most attractive at the time of design (which may be ill-suited over long term due to inherent uncertainty in complex systems such as Internet). Such a design approach is able to reduce, if not remove, designer bias. Further, designing while catering to conflicting stakeholders can allow the network to function more freely, without requiring intervention through regulation (which can often be blunt in its operation).

Clark et al (2005) have given a design principle to cater to such stakeholder conflicts: designing for ‘tussle’ (where tussle refers to the interaction between stakeholders with conflicting interests). In this section I will present the foundations laid down by Clark et al (2005) for an architecture that is designed for ‘tussle’ as well as supplement these foundations with examples that can indicate how they might be applied to architectural designs. I will also present some of the difficulties involved in incorporating this design principle.

Clark et al (2005) split the principle of designing for tussle into two further sub-principles: modularize along tussle boundaries and designing for choice.

6.2.1 Modularize along Tussle Boundaries

As already argued, modularity can help in managing complexity and allowing independent design of components in a system. Within the design principle of using modularity along tussle boundaries, the aim is to isolate tussles. That is, tussles are provided separate spaces so that they can occur independently of each other. In this way, functions enabling specific tussles to be carried out should have a minimal or no impact on other tussles. Similarly, the stakeholders not involved in that specific tussle should not

be impacted by that tussle. Stakeholder separation and functional separation can also be though of as two distinct categories. For example, stakeholders might be separated within a functional tussle space, such that their internal choices do not affect other stakeholders.

An example of separation of functions and stakeholders is represented by the separation between inter-domain and intra-domain routing protocols. The inter-domain routing protocol is responsible for sending messages between two different networks whereas intra-domain routing protocols send messages within the same network. There are multiple protocols available for intra-domain routing whereas BGP remains the main protocol for inter-domain routing. Because of the choices involved, intra-domain routing can be done using protocol best suited within that domain. Further, these intra-domain routing protocols can be evolved or even replaced without having an effect on inter-domain routing protocol. To make this more clear, an ISP works within one network (these networks are referred to as Autonomous systems). Within this domain (or network) an ISP can change its interior routing protocol (intra-domain routing protocol) depending on the needs of its users and its own capacities. This change is made easier as it does not affect other ISPs and therefore the stakeholders are separated.

However, separation of tussles is not an easy task and can cause spillover effects. One example which represents the difficulty in tussle isolation or stakeholder separation is that of Network Address Translators (NATs), described previously in this thesis as well. NATs allow small networks (such as those within a company) to manage its internal addressing independently of the ISP (kalogiros et al, 2009). For example, the network can add more internal hosts without seeking permission from the ISP. Here the addressing is decoupled between external and internal networks, as a form of stakeholder separation. The tussle over the address space, however, has other consequences. Particularly, it has consequences for other stakeholders, bringing them into a tussle where they didn’t belong. For example, NATs universal access to end-hosts, limiting the node behind to use only a set of protocols, and also restricting innovation in this sense. Applications such as Skype, which require direct end-host reachability are also unable to function behind NATs. This example shows that modularization along tussle boundaries is not an easy task and may

involve iterative designing from the designers.

6.2.2 Design for Choice (to exert control over tussle outcomes)

The principle of designing of choice extends the separation of stakeholders to giving them control to influence the outcome of a tussle. This can be done by allowing the stakeholders to choose a preferred configuration of a protocol at ‘run time’, that is, the choice is built it within the protocol after its deployment. Stakeholders might use their control themselves, not use it at all, or even delegate to a third party. For example, users may not directly choose a protocol configuration, but may choose ISPs that offer them services with a specific configuration. Delegation of choice can be particularly important for users as not all users may have enough technical knowledge to exercise their control on the network, even if they have the choice to do so. For many users, choice may be a burden rather than a facility.

Tussles may sometimes also lead to stable conditions, at least temporarily. This could be because stakeholders may not find each other’s interests adverse, or may find more value in aligning interests together for some value. This value may be monetary in most cases, but may also be non-monetary in some. One example of a non-monetary value is peer-to-peer (P2P2) file sharing where users share a symbiotic relationship with other users (although there may be direct or indirect monetary benefits to some users)..

For a designer, designing for choice would mean identifying all relevant stakeholders as well as their interests. These interests would determine what kind of stakeholder actions must be supported by the design to allow them to have an influence over the outcome. The sites of these actions are called ‘interfaces’. An example of a protocol designed for choice is SMTP (Simple Mail transfer Protocol) (Clark et al, 2005). The SMTP allows the user to configure the protocol such that the user is able to pick a server (which will send the mail) of her/his choice. This choice maybe made depending on the user’s preferences. For example, a user may choose a server because of its reliability. Or, a user may choose a server because it provides spam filters. Yet, SMTP also serves as an example of a

protocol where the playing field for different stakeholders is not level. In particular, the ISPs can take away the choice from the users to choose their preferred servers. Through the Deep Packet Inspection technology, ISPs can identify which mail server has been chosen and then block the sending of the mail, forcing the user to choose its own server (Kalogiros et al, 2009). Therefore, designers need to be cautious about the way in which the control is distributed over tussle outcomes.

In the end designing for tussle is a beneficial choice but one with difficulties. These difficulties are part technical but they also require diligent evaluation of design choices from non-technical experts such as economists.

6.3 Designing for Network Intelligence

In the last chapter I argued for the benefits of building 'intelligence' in the network. An intelligent network collects data about itself, which can be used for better evolution of the network. In this way, intelligence in the network is an information feedback, akin to the one found in biological as well as man-made complex systems. Intelligent network can provide other benefits such as addition of extra services based on the data collected. This can also be critical in solving the economic tussle over the role of ISPs in the current Internet. In particular, by providing additional services, by using the data collected by the network, ISPs can create sustainable business models, without the risk of the market moving to a monopoly (as unstable business models go bankrupt).

Some examples of additional services that can be made available with the use of information about network usage includes diagnosis of faults, efficient resource allocation (avoiding data congestion, for example), providing quality of service, and setting up of transparent and fair pricing mechanisms (Ford et al, 2009; Clark et al, 2003).

One approach to designing for Intelligence would be to design mechanisms to collect information about network's use and applications running over it in the core of the network. Yet, such an approach would go against a fundamental property of a network like Internet - it's ability to host a wide range of applications. How can the network

collect information about it without losing its ability to be general enough to host a range of applications? Here, I present two approaches towards this end, by Ford et al (2009) and Clark et al (2003) respectively. These two approaches represent a move towards a more specific implementation of the general design principle of designing for intelligence through a feedback loop by which the data about the network's use characteristics is collected. In the next two sub-sections I will present these two approaches as well provide a comparative analysis as a guideline for designers to choose aspects from these approaches for a more specific implementation of this principle. I also offer a third approach as a combination of these two approaches.

6.3.1 Fuzzy Ends Principle

Ford et al (2009) propose a design principle, Fuzzy Ends principle, for a future Internet to provide additional services in the network:

Allow the endpoint to explicitly delegate some functions into the network, so the end is effectively a distributed system." (Ford et al, 2009).

This principle basically states that instead of the network's core being built to take decisions on its own about what tasks to perform, this decision should be left to the end points, which can then delegate this task to the core. The advantage of such a strategy is that it does not require the core to have specific functions which might decrease the range of applications that can be implemented onto the network (or atleast make it difficult to design a wide range of applications) making it non-conducive for innovation, Some of the examples of the services that may be added by use of this principle include content caching and protecting. Regarding the former, a network can be delegated to cache content from a server (an end point) for providing services such as a speedy delivery or content optimization for a particular user. As for protection, a network may be delegated to block content from specific sources such as adult content websites, chat rooms, gambling websites,etc.

While Ford et al (2009) do not explicitly stress upon data collection about the network by the network, the examples I mention above need some kind of data about network's use (for example, which website is being visited). Therefore, I argue that this principle can be a helpful strategy in adding intelligence to the network without losing its ability to facilitate innovation. By stressing upon 'explicit delegation' of tasks from the ends to the network, the principle ensures that the network would not impose itself onto the applications (or require them to have specific features).

6.3.2 Knowledge Plane

Clark et al (2003) give a more abstract proposal to provide additional services on the network using data collected by the network. They argue that this data should be collected within a 'knowledge plane', which is a decentralized and distributed construct within the overall network architecture that gathers, aggregates and manages data about network behavior and operation. It is then able to provide this data in an integrated form to all stakeholders involved (users, providers, and the network itself). One of the important factors considered in their proposal is that construction of such a knowledge plane should not violate the transparency of the network, such that it is still able to host a wide range of applications. While they do not give explicit details about how such a knowledge plane should be constructed, they do provide some guidelines for future research aimed at constructing such a plane to consider. For example, they argue that the Knowledge plane can be designed through a bottom-up approach (Rather than a top-down approach engineered globally) where simple entities can coagulate to form more complex entities depending upon the need. For example, a web server is one simple entity but combined with many other web servers, it can serve as a powerful system capable of fulfilling functions required for the knowledge plane (such as storing and integrating data into meaningful form). This approach is akin to one followed in citizen science projects, where a number of users share parts of the computing capacities of their computing devices, aggregating into a powerful and inexpensive supercomputer.

While the Fuzzy ends and Knowledge principle represent similar aims, to add extra

services to the network using data about its behavior without violating the the transparency of the network, they differ in the scale or level of information collection and processing. While the fuzzy ends principle aims at collection of data at a limited scale to carry out tasks delegated through the ends, knowledge plane aims at collection and integration at a global level. The benefit of a global level integration of data is that, the knowledge plane would be able to give information about failures in a specific region of the network (whereas if it was restricted to implementation at that region separately from other regions, the knowledge plane would be rendered useless along with the failure in that region). Yet, implementation at a global scale makes construction of a knowledge plane would add to the complexity of the system, which may further have repercussions on the performance of the system as well as alter economic incentives(and constraints) significantly.

The two principles can however, be combined into a third approach. As argued before, the Fuzzy ends principle offers a clear approach to retaining the innovative capacity of the network by explicitly stating that the core of the network should not have specific functions. The Knowledge Plane approach however, leaves this question unclear and only states that any functionality built in any region of the network (Whether the core or the ends) should be transparent. However, transparency may not always result in easy facilitation of innovation. If the core of the network was built with functionalities, even if these are made transparent to those willing to innovate on the network, it might make the task of innovating and testing the innovations more difficult and thereby, taking away the economic (or atleast reducing the incentives to innovate). Therefore, the fuzzy ends approach should be followed in its suggestion to keep the decision for functionality at the ends. Yet, the Knowledge plane approach offers an advantageous proposition by suggesting a more global integration of data. A complete global integration of data would, however, require this integration to take place in the network, which would, again, reduce the generality of the core making it more difficult to innovate on the network. An incomplete integration of data, however, might be possible through interaction of different end points, such as the integration of different web servers suggested above. In the end, a specific implementation of designing for intelligence would have to make a

compromise between building functionality within the network or settling for an incomplete integration of data (that is not at a global scale). As argued for other design principles mentioned in this chapter, modularity and designing for tussle respectively, implementation of an intelligent network would require a multi-disciplinary effort requiring a host of technical and non-technical experts than can work together to evaluate the different specific technical implementations of designing for intelligence.

6.4 Designing an evolvable, heterogeneous, intelligent Internet for tussle : Recommendations for Future work

The design principles presented in this chapter contribute together to an Internet built on the meta-requirements argued for in the last chapter. For example, an evolvable Internet requires modularity in design, but it can also be aided with Intelligence built into the network, so as to give a preferable direction to the evolution. The principles also directly compliment each other. For example, stakeholders can use the Intelligence of the network to decide upon their actions supported through tussle aware design.

However, as mentioned earlier, the aim of this thesis, was not to provide specific technical implementations of these design principles, as this is beyond the scope of this thesis as well as expertise of this author. Any project aimed at development of technically implemented future Internet should, and must, involve a multi-disciplinary teams of experts, whose expertise ranges from technical fields to non-technical fields such as philosophers, economics and law. For example, the evaluation of different possibilities within a modular design can assessed through options theory and the economic incentives it provides for innovation. Similarly, whether a transparent functionality can be built into the core of the network to collect data bout is own usage, will also have to be evaluated through an examination of its possible economic repercussions. Law makers will also have to pitch in to make sure that technical implementations do not support one stakeholder unfairly over the other, concentrating power in just a few hands.

Similarly, philosophers, and particularly ethicists, will also have an important role to play

in the eventual design and engineering of the future Internet. This is of course, not unique to the case of Internet. As Carl Mitcham (1998) has argued, philosophy is important to engineering on at least three accounts. First, it would allow engineers to respond to philosophical critics and justify the design choices. Second, philosophy can help engineering with ethics and recognition of social contexts. Finally, engineering provides a new philosophical model or way of life. This thesis has mostly paid attention to the latter two relations between engineering or design and philosophy. There is, however, much more to be done during the design in order to achieve the vision for the future Internet outlined in this thesis.

For example, while I argued that as a meta-requirement, the future Internet should be heterogenous, catering to a diverse set of societies future Internet would be embedded in, this task would require a conceptual analysis and reflection of the potential designs. Such an analysis is necessary to make sure that implicit ideologies of the designers and engineers are not represented in the design, making it antagonistic to users from different cultural backgrounds, even if it was not intended by the designers. Similarly, designing for uncertainty, which goes against the traditional design methodology which often relies on predictive mechanisms, would also require a close examination. To come back to Mitcham's (1998) arguments, and to reiterate what I already stated, this importance of philosophy for engineering is not unique to the case of future Internet. However, this thesis should be seen as another example which makes the importance of relation between philosophy and design explicit, and therefore, compels us to re-think both engineering and philosophy as academic fields, such that the link between the two is fairly represented in their pedagogy as well as practice.

Finally, to conclude this chapter, it should be stated that a project aimed at development of a future Internet is likely to be iterative in its process and would require the efforts of a multi-disciplinary team in not just the design phase but in tests and pilot phases as well. This thesis does however, provide guidelines for such a project to be undertaken and to move towards the goal of a future network than can support the socio-economic goals of a diverse world.

Chapter 7:
Concluding summary

Since its inception as DARPA project around 5 decades ago, Internet has come a long way. It has seen unprecedented growth, both in number of users as well as functionalities it provides to these users. This growth was largely unanticipated at the time of the design, and as a consequence, a lot of conflicts between various stakeholders involved in Internet's use and regulation, that were not addressed by design, have emerged. Some of this conflicts have led to short term technical fixes, such as addition of Firewalls for added security or Network Address Translators for addressing the problem of limited addresses. These short term fixes, however, violate the principles of original design and lead to incoherent architecture, that is likely to produce new problems in the future. As an alternate approach, a number of Internet researchers have advocated for a clean-slate approach to architecture, that is a new future Network like the Internet, that is unconstrained by the current Internet. This thesis is a step towards a development of such an architecture.

However, in order to design a clean-slate architecture for a complex network like the Internet, we require: a.) a clear understanding of what the term architecture means, as well as how they are designed in the context of network systems, b.) what is the architecture of the current Internet, c.) how does this architecture lead to the conflicts of Internet's design and regulation, d.) what can we learn from these conflicts and its relation to architecture in order to design a new architecture. I addressed these questions in Chapter 2-5 of this thesis.

Specifically, in chapter 2, I discussed the concept of a network architecture and argued that corresponding concepts of meta-requirements and design principles can provide a more useful way of designing specific architectures that are aimed towards desirable socio-economic goals. Meta-requirements refer to those properties of the system that are critical beyond the functionality (as many architectures may fulfill the same functionality

but may differ in properties such as security and modifiability). I then discuss some of the basic design principles, which are required to translate meta-requirements into specific architectures, in their general form, involved in design of network architectures.

In chapter 3, I presented the architecture of current Internet along with the design principles that guides it design. Here, I also argued that design principles can have specific and critically different technical implementations. This chapter laid down the groundwork for discussion of the debates and conflicts over Internet's design and regulation and their relation to its architecture. I discussed these debates in Chapter 4. These include issues such as the economic challenges faced by Internet Service Providers, cybersecurity, Network Neutrality, as well as the debates about the use of use of technical extensions that violate the original design principles of the Internet architecture. An understanding of these issues allows for a reflection on the underlying causes of these problems in their general form.

Since the central aim of this thesis was to provide a step towards a future Internet architecture, the answer to the question of how these debates should be settled within the context of current Internet was not relevant to this thesis. Rather, in this thesis, I reflected on some common underlying conditions that give rise to the multiple challenges for current Internet (such as debates over network neutrality, cybersecurity, the economic challenges for service providers, conflicts over censorship and privacy on the Internet, and so on) such that they are also applicable for any network deployed globally on a large scale. As a clean-slate architecture is unconstrained by the current Internet architecture, it can be designed to avoid problems identical with the current Internet. This brings us to the central question of this thesis: *What meta-requirements should guide the development of a future Internet architecture, and how can these meta-requirements be translated into architectural principles such that they lead to a future Internet architecture for an open and diverse world?*

Therefore, in Chapter 5, I argued that the conflicts over Interrnet's design and regulation emerge because of four underlying conditions: inherent uncertainty and lack of

predictability of future use of a network like Internet, diverging stakeholder interests and views, lack of feedback or data about how the network is being used, and diversity of societies Internet is embedded in. For example, the unanticipated growth in number of users, increase in functionalities available to these users, emergence of business opportunities, increase in computing powers of computing devices, emergence of new technologies such as wireless phones, have led to unanticipated outcomes for the Internet. The history of the Internet points to the inherent lack of predictability of outcomes for a future network of similar scale and reach. Therefore, I argued that the Internet comes with an additional challenge of unpredictability and any future designs of the Internet would have to take that into consideration.

Can we however, use this lack of predictability of outcomes and inherent uncertainty as an opportunity rather than treating as a risk that needs to be protected against?. In this context, I discussed the concept of resilience, that has emerged out of research on anticipation (O' Malley, 2010). Resilience differs from the traditional approaches to uncertainty such as risk and preparedness. While probabilistic risk is a technique of harm-minimization, preparedness involves creation of mechanisms for coping with imagined harms. Resilience, however, is a more encompassing and systematic approach to anticipation as well as toleration of disturbances that occur out of uncertainties. However, I argued that resilience is essentially a reactive concept and is harder to apply to complex systems where prediction is difficult, if not impossible. As a substitute for resilience, I argued that the notion of 'antifragility', a neologism coined by Taleb (2012), referring to the capacity of a system to gain from stress or disorder, is more useful. Antifragile systems thrive on uncertainty rather than protect against it. In the context of a network like Internet, this capacity rests upon the network being evolvable, and conducive to innovation. I supported the argument for the need of an evolvable and innovation supportive design by arguing that the benefits of an evolvable and innovation facilitative design can already be witnessed through the history of the Internet. Specifically, I offered a comparative analysis of Internet with an earlier available network, CompuServe. Here, I argued that CompuServe lost out to the Internet (as users left CompuServe) because of lack of antifragility, or more specifically, its inability to evolve

with user demands and support innovation.

In the same Chapter (5), I further claimed that Internet's complexity and the diversity of the stakeholders involved, make it difficult to reach a point of stability, with conflicts emerging in 'run time'. While the designers can tilt the weights towards one stakeholder over the other others, such a move can lead to various harmful consequences. For a global technology laden with uncertainty and a diverse stakeholder set, designers need to be conscious and cautious about not embedding their own values into the design. Similarly, drawing from the debates over current Internet's design and regulation, I claimed that a future network can significantly benefit from addition of intelligence, or feedback, about its own use characteristics. Finally, I contended that the architecture for a future Internet should be minimally defined and made heterogeneous (flexible over different regions over network), because there will be few requirements that will be truly global.

These underlying conditions described above also point to the fact that approaches such as Values Sensitive Design (VSD), which is one of the most reviewed approaches pertaining to values in ethical design of technology, are problematic to be used for design of a complex network like the Internet. This, I argued, is because of two major reasons. First, the VSD methodology requires identification of stakeholders as well as their views on the technology as an important part of its conceptual investigations. This, however, is difficult, if not impossible for a complex network like Internet, as new stakeholders keep emerging while old ones may change their positions on design and regulation of Internet. Secondly, VSD methodology conceives 'values' as being interactional, that is, they only come in to play when the technology is in use. Further, values also remain abstract in their definition and are likely to mean different things for different societies or even stakeholders within the same society. As a result, VSD does not offer itself as a practical guide to design of a global network like the Internet which is embedded into a wide range of societies, with great many cultural and moral differences. The inherent uncertainty as well as the diversity of uses of a network like Internet, would also make it impossible for designers to pick out any particular values to be necessarily catered to. Instead, I argue, that designers should focus on an overarching value of designing for an open and diverse

world which is laden with uncertainty. For such a design, I suggested four meta-requirements: Evolvability, distributed control to stakeholders, Intelligence within the network, and Heterogeneity.

Subsequently, in Chapter 6, I concluded the thesis by translating these meta-requirements into design principles that can guide the development of a future Internet architecture. These design principles were: modularity, design for 'tussle', and designing for Intelligence. While modularity aims at making the architecture evolvable as well as conducive to innovation, designing for tussle aims at providing control to different stakeholders to assert their own preferences onto the network. The principle to design for intelligence requires collection of data by the network about its own usage, and can be helpful in providing future directions for the development of network, giving information to stakeholders about how they can best use their control, and providing diagnostic information in case of network failures. However, the purpose of this thesis was not to provide specific technical implementations of these design principles, since this would also be beyond the scope of this thesis as well as expertise of this author. Rather, in this thesis, I provided design principles for a future Internet, on a general level, along with evaluation tools that a research team involved in development of a future Internet can use to evaluate the different specific implementations of these design principles. For example, the evaluation of different possibilities within a modular design can assessed through 'options theory', presented in chapter 6, and the economic incentives it provides for innovation. Similarly, whether a transparent functionality can be built into the core of the network to collect data bout is own usage, will also have to be evaluated through an examination of its possible economic repercussions. Law makers will also have to pitch in to make sure that technical implementations do not support one stakeholder unfairly over the other, concentrating power in just a few hands. Any project aimed at development of technically implemented future Internet should involve a multi-disciplinary teams of experts, whose expertise ranges from technical fields to non-technical fields such as economics, social sciences, philosophy and law. Such a project is likely to be iterative in its process and would require the efforts of a multi-disciplinary team in not just the design phase but in tests and pilot phases as well.

This thesis does however, provide guidelines for such a project to be undertaken and to move towards the goal of a future network than can support the socio-economic goals of a diverse world.

References

Aström, K. J., & Murray, R. M. (2010). Feedback systems: an introduction for scientists and engineers. Princeton university press.

Banks, Michael A, and John R Vacca. (2012) On The Way To The Web. Apress, New York Print.

Bender, J. (2015). 6 Countries that block social media. Business Insider, April 6, 2015. Accessed on 29TH July, 2016 from http://uk.businessinsider.com/the-six-countries-that-block-social-media-2015-4

Braden, R., Clark, D., Shenker, S., & Wroclawski, J. (2000). Developing a next-generation Internet architecture. White paper, DARPA.

Brey, P. (2010). Values in technology and disclosive computer ethics. The Cambridge handbook of information and computer ethics, 41-58.

Brown, I., Clark, D. D., & Trossen, D. (2010, November). Should specific values be embedded in the Internet architecture?. In Proceedings of the Re-Architecting the Internet workshop (p. 10). ACM.

Callon, M. (1987). Society in the making: the study of technology as a tool for sociological analysis. The social construction of technological systems: New directions in the sociology and history of technology, 83-103.

Callon, M. (1990). Techno-economic networks and irreversibility. The Sociological Review, 38(S1), 132-161.

Cath, C., & Floridi, L. (2016). The Design of the Internet's Architecture by the Internet Engineering Task Force (IETF) and Human Rights. Science and engineering ethics, 1-20.

Christensen, C. M. (1997). The Innovator\'s Dilemma: The Revolutionary Book that Will Change the Way You Do Business.Collins Business Essentials.

Clark, D. D., Partridge, C., Ramming, J. C., & Wroclawski, J. T. (2003, August). A knowledge plane for the internet. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (pp. 3-10). ACM.

Clark, D., Braden, R., Sollins, K., Wroclawski, J., & Katabi, D. (2004). New arch: Future generation internet architecture. Massachusetts Institute of Technology, Cambridge lab of Computer Science.

Clark, D. (1988), "The Design Philosophy of the DARPA Internet Protocols". Proc SIGCOMM 1988

Clark, D. D., Wroclawski, J., Sollins, K. R., & Braden, R. (2005). Tussle in cyberspace: defining tomorrow's internet. IEEE/ACM Transactions on Networking (ToN), 13(3), 462-475.

Denardis, L. (2013). Protocol politics: The globalization of Internet governance. Boston: MIT Press.

Denardis, L. (2014). The global war for Internet governance. New Haven: Yale University Press.

Ford, A., Eardley, P., & Van Schewick, B. (2009, June). New design principles for the internet. In Communications Workshops, 2009. ICC Workshops 2009. IEEE International

Conference on (pp. 1-5). IEEE.

Friedman, B., Kahn Jr, P. H., Borning, A., & Huldtgren, A. (2013). Value sensitive design and information systems. In Early engagement and new technologies: Opening up the laboratory (pp. 55-95). Springer Netherlands.

Galloway, A. R. (2004). Protocol: How control exists after decentralization. MIT press.

Hansell, Saul.2008. 'Apple's Capricious Rules For Iphone Apps'. The New York Times. Print

Hayashi, Y. & Iwata M. (13 March,2011). Japan Struggles to Control Nuclear Reactors, Wall Street Journal.

Heylighen, F., Cilliers, P. and Gershenson, C. (2006) - Complexity and Philosophy - Complexity, Science and Society. Radcliffe Publishing Society, Oxford.

Internet Engineering Task Force. (2004). RFC 3935 a mission statement for the IETF. Retrieved May 1, 2016 from https://www.ietf.org/rfc/rfc3935.txt.

Kalogiros, C., Kostopoulos, A., & Ford, A. (2009). On designing for tussle: Future internet in retrospect (pp. 98-107). Springer Berlin Heidelberg.

Kazman , R. , Klein , M. , Barbacci , M. , Longstaff , T. , Lipson , H. , and Carriere , J. (1998) .The Architecture Tradeoff Analysis Method. Fourth International Conference on Engineering Complex Computer Systems .

King, G., Pan, J., & Roberts, M. E. (2013). How censorship in China allows government criticism but silences collective expression. American Political Science Review, 107(02), 326-343.

Latour, B. (1990). Technology is society made durable. The Sociological Review, 38(S1), 103-131.

Lentzos, F., & Rose, N. (2009). Governing insecurity: contingency planning, protection, resilience. Economy and Society, 38(2), 230-254.

Lessig, L. (1999). Code and other laws of cyberspace (Vol. 3). New York: Basic books.

Lessig, L. (2002). The future of ideas: The fate of the commons in a connected world. Vintage Books.

LORENZ, E.N. (2000). The butterfly effect. In R. Abraham & Y. Ueda (Eds.), The chaos avant-garde: Memories of the early days of chaos theory (pp. 91–94). River Edge, NJ: World Scientific.

Makridakis, S., & Taleb, N. (2009). Decision making and planning under low levels of predictability. International Journal of Forecasting, 25(4), 716-733.

Manders-Huits, N. (2011). What values in design? The challenge of incorporating moral values into design. Science and engineering ethics, 17(2), 271-287

Mitcham, C. (1998). The importance of philosophy to engineering. Teorema: Revista Internacional de Filosofía, 27-47.

McAfee (2014). Estimating the Global Cost of Cybercrime. Report by McAfee, Centre for Strategic & International Studies.

O'malley, P. (2010). Resilient subjects: Uncertainty, warfare and liberalism.Economy and Society, 39(4), 488-509.

Peterson , L. L. , and Davie , B. S. 2007 . Computer Networks: A Systems Approach ,

fourth edition. Elsevier.

Saltzer , J. H. , Reed , D. P. , and Clark , D. D. 1981 . End-to-End Arguments in System Design. 2nd International Conference on Distributed Computing Systems .

Saltzer , J. H. , Reed , D. P. , and Clark , D. D. 1984 . End-to-End Arguments in System Design. ACM Transactions on Computer Systems 2 (4): 277 – 288 .

Schilling , M. A. 2000 . Toward a General Modular Systems Theory and its Application to Interfi rm Product Modularity. Academy of Management Review 25 (2): 312 – 334

Shaw , M. , and Garlan , D. 1996 . Software Architecture: Perspectives on an Emerging Discipline . Prentice-Hall .

Taleb, N. N. (2007). The black swan: The impact of the highly improbable. Random House Incorporated.

Taleb, N. N. (2007). Black swans and the domains of statistics. The American Statistician, 61(3), 198-200.

Taleb, N. N. (2012). Antifragile: Things that gain from disorder (Vol. 3). Random House Incorporated.

van der Velden, M. (2007). Invisibility and the ethics of digitalization: Designing so as not to hurt others. In S. Hongladarom & C. Ess (Eds.), Information technology ethics: Cultural perspectives (pp. 81–93). London: Idea Group Reference.

Van Schewick, B. (2012). Internet architecture and innovation. MIT Press.

Van Schewick, B. (2012). Network neutrality and quality of service: What a non-discrimination rule should look like. Stanford Law Review, Volume 67.

Vinck, D. (2003). Everyday engineering: An ethnography of design and innovation. MIT Press.

Zittrain, Jonathan. (2008) The Future Of The Internet And How To Stop It. New Haven [Conn.]: Yale University Press.