Helping A Future Internet Architecture Mature

Craig Partridge
Raytheon BBN Technologies
craig@bbn.com

This article is an editorial note submitted to *CCR*. It has NOT been peer reviewed. The author takes full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

Some of the challenges of developing and maturing a future internet architecture (FIA) are described. Based on a talk given at the Conference on Future Internet Technologies 2013.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**].

Keywords

Network architecture, future internet architecture.

1. INTRODUCTION

The successful Future Internet Architecture (FIA) research initiatives in Asia, Europe and the United States have led to multiple proposed new Internet architectures. The maturity of these architectures varies widely. Some efforts are still refining their initial visions while others have working prototypes. Collectively, however, these efforts have evolved from initial ideas into working concepts.

At an FIA meeting in the first part of 2013, the topic of how to help an FIA mature came up. A number of people (including the author) started to draw lessons from the 1970s and 1980s.

For those who don't remember, the 1970s and 1980s were an era of many internet architectures, offering competing visions, and often-competing products in the marketplace. Protocol suites such as IBM's Systems Network Architecture (SNA), Novell Netware, XEROX PUP, the Open Systems Interconnection (OSI) standards, DECNET and TCP/IP all vied for research and commercial attention. At the FIA meeting, there was a spirited discussion of some of the lessons the participants had learned from working in a world of competing architectures.

I subsequently imported some of the discussion into a keynote address I gave at the Conference on Future Internet Technologies (CFI 2013) in Beijing.

In the talk, I sought to identify what I thought were the key lessons to be learned from the 70s and 80s (with the warning that my experience was all in the 80s), in the hopes those lessons would be useful to current FIA researchers. Several CFI participants asked me to reduce the talk to an editorial.

2. ADVICE FOR A MATURING ARCHITECTURE

The advice here is for the architect or architects of a new network architecture that has already done some research and development and shown promise. Such architecture has resolved its initial goals and aspirations into an image of the future it wishes to realize. I refer to this resolution as the architectural "vision."

The goal of this advice is to help the architect(s) to follow their vision and mature their architecture to the point that it makes an impact beyond the research community. I'm interested in helping good architecture ideas reach the development, implementation and product communities, as so many of those 1970s and 1980s architectures did.

This editorial is not about winning a protocol war. One of the characteristics of the 1980s was that the networking community understood how to borrow/import/steal good architectural ideas from each other's architectures. That borrowing is important intellectual cross-fertilization, and requires architectures to be mature enough that they are worth borrowing from. Helping today's future Internet architectures reach that level of maturity is the goal of this paper.

3. SIX PRINCIPLES

To help mature a future Internet architecture, I offer six principles.

3.1 Stick to Your Vision

Every internet architecture reflects an insight or vision about how data communications should evolve. Here are a few examples from the past:

- SNA: sought to provide a flexible way to interface terminals and printers to programs over a common wiring plant[4];
- TCP/IP: sought to use packet switching to enable the interconnection of different networks[2];
- XTP (Xpress Transfer Protocol): sought to show that by designing protocols for easy hardware implementation, we could drive down network costs and get improved performance[8].

While these visions sound simple, actually fleshing them out took years. The most notable example is that the TCP/IP effort, which began in 1974, took four years (!) to realize that the Internet Protocol (the IP part of TCP/IP) was required[7]. Commercial acceptance did not come until the late 1980s.

Do not expect that once you have your vision, that progress will come swiftly. It probably won't. You will often feel you are slogging your way forward with great effort. That's normal and a consequence of living on the edge of the future.

3.2 Don't Let Applications Distract You

You probably have a killer demo that illustrates how your network architecture is great for a particular type of application. That's good, as it illustrates how you envision data will flow through networks using your architecture and what makes your architecture appealing.

The problem is that, if your architecture gets much traction, people will begin to show up to ask you how to implement their difficult application in your architecture. My advice is to ignore them, for two reasons.

First, if you focus on supporting specific applications, you tend to compromise your architecture to support those applications. Today's applications are indifferent guides to the future. As an example, consider the US gigabit testbed efforts, which were required to highlight applications. The testbeds had great demos and did wonderful work educating people on the challenges of gigabit networks, but none of them developed a good architecture.

Second, if your network architecture is successful, the applications will likely adapt. Applications are more malleable than people think. For instance, consider the video over the Internet issues of the late 1980s and 1990s. Video codec designers of the time were insistent that any network carrying video could not drop packets. Their argument was that every datum of the video stream had to be delivered, because any data that did not need to be delivered was already compressed out of the video stream. Eventually, however, people developed video codings that were robust to the loss inherent with the Internet and enabled a booming network video industry with companies such as YouTube and Netflix.

As a network architect, you will do far better to think about where you want to position work in the network, and how you want data to flow between entities.

3.3 Accept You Cannot Answer Some Questions

The video codec story from the previous section illustrates an important point. The video experts of the time insisted that the Internet's transmission model was flawed. An obvious answer would have been to change the Internet architecture to support video, based on this expert advice.

Instead the Internet community decided not to change the Internet. The community decided to assume that, eventually, the video problem would be solved.

The Internet community was correct, but it meant for several years that questions about how the Internet would support video were met with variations of "we don't know."

My view is you should accept that, in some cases, you have to say, "we don't know." You cannot be a domain expert in every application that will transmit information through your network architecture. So, sometimes, you need the confidence to say that you're sure your architecture will be able to do something, but you don't yet know how.

3.4 Don't Hoard Your Problems

There's an instinct to solve all the challenges yourself or in your research group. Reasons vary but include a desire to keep all the fun papers for your graduate students or a sense that if the problem is really important, you as network architect, need to be the one solving it.

Unfortunately, if the network architecture proves even moderately successful, there will be far too many challenges for one person or even one group to solve. Rather your focus should be on articulating your vision and encouraging others who share your vision to help out. If you can, also find ways to discourage bad ideas.

Two examples that illustrate this approach come from the operating system community and the Internet:

- Linus Torvalds, the leader of the Linux effort, encourages wide participation in the development and maintenance of Linux, but reviews the proposed changes. In many ways this is like the Berkeley Unix community, where Berkeley encouraged contributions of code but also evaluated what they received before adding it to the distribution.[1]
- Similarly, the Internet Engineering Task Force (IETF) encourages wide participation in the Internet standards process, but also reviews proposals to ensure they are consistent with the Internet vision.

3.5 Don't Worry About Backward Compatibility and Transition

One of the first signs that an internet architecture is beginning to succeed is that the architect starts getting bombarded with questions about how to integrate the new network architecture with today's network(s). In the 1970s the question was how to integrate the Internet with the phone system. Today you'll get asked how to transition your architecture to the Internet.

These are absolutely the wrong questions. Modifying your architecture to improve compatibility or transition usually harms the architecture!

Furthermore, if your architecture truly offers new capabilities, one of two things will likely happen:

- The existing 'Net will evolve to incorporate your architecture. This is usually achieved by layering your architecture over or under the existing 'Net. For example, both the Internet and SNA found ways to layer themselves over X.25 and the Internet found a way to layer itself over ATM[3][9].
- Alternatively, the rest of the world will find a way to transition to your architecture. For instance, in the late 1980s, BITNET, CSNET, USENET and the Netware networks all found ways to transition to the Internet[6].

So transition is ***not*** your problem. Indeed, you will do better focusing on making your architecture better, as that will give additional motivation to others to evolve towards your vision.

3.6 Don't Get Caught Up in Performance Issues

There is often an impulse to worry about the performance of implementations of your architecture. In 1975, Bob Kahn fretted that poor TCP performance would sink TCP/IP's chances of adoption[2]. Besides, working on performance is often fun.

My advice is not to let performance issues distract you. You are developing a *Future* Internet Architecture, and your architecture will run on *future* hardware, using *future* compilers, libraries and maybe languages. Worrying about performance on today's systems is misguided. Your goal on today's systems is simply to show your architecture can be implemented.

As examples in support of this notion, consider the following:

- Cisco's first router was built using off-the-shelf desktop computing hardware (a Motorola 68000-series processor in a Multibus chassis). The features we now

[1] I remember a talk by Kirk McKusick many years ago where he observed that ignoring complaints about BSD often paid off, because eventually one of the complainers would get annoyed enough that they'd "start sending *code*."

[2] Bob asked Bill Plummer to rewrite the TENEX TCP in assembly language to make it faster.

consider critical to router technology such as ternary CAMs, network processors, and switched backplanes were developed years later.

- TCP performance remained dreadful until 1989, when Van Jacobson and Dave Borman showed how to optimize implementations[1][5].
- The 1980s network architecture that worried the most about performance, the eXpress Transfer Protocol (XTP), was the least successful!

4. Conclusion

Ivy Lee is said to have advised, "Do the most important thing first." My hope is that this advice will help the aspiring network architect focus on what is most important to maturing his network architecture, and help him avoid the distractions that are less important.

REFERENCES

[1] Borman, D. 1989. Implementing TCP/IP on a Cray Computer. *ACM Computer Communication Review*, 19, 2 (Apr 1989), 11-15.

[2] Cerf, V.G. and Kahn, R.E. 1974. A protocol for packet interconnection. *IEEE Trans. Communications* 22, 5 (May 1974), 637-648.

[3] Heinanen, J. 1993. Multiprotocol encapsulation over ATM Adaptation Layer 5. *Internet Requests for Comments*, RFC-1483, (Jul 1993).

[4] IBM 1975. *Systems Network Architecture; General Information*, 1st edition, (Jan 1975).

[5] Jacobson, V. 1990. 4BSD Header Prediction. *ACM Computer Communication Review*, 20, 1, (Apr 1990), 13-15.

[6] Partridge, C. 2008. The technical development of Internet e-mail. *IEEE Annals of the History of Computing*, 30, 2 (Apr-Jun 2008), 3-29.

[7] Postel, J. 1978, Draft Internetwork Protocol Specification. *Internet Experiment Note (IEN)*, 28, (Feb 1988).

[8] Strayer, W.T., Weaver, A.C., Dempsey, B.J. 1992. *XTP: The Xpress Transfer Protocol*.

[9] Turner, J.S. 1986. New directions in communications (or which way to the communications age?). *IEEE Communications Magazine*, 24, 10 (Oct 1986).