A New vision for network architecture

David Clark
M.I.T. Laboratory for Computer Science
September, 2002
V3.0

Abstract

This is a proposal for a long-term program in network research, consistent with the objective of “cognitive computing”. It attempts to build on the successes of today without being locked into a strictly incremental path of progress. It recognizes the important work that DARPA has done, such as Active Networks and wireless, and builds on this work. It provides a framework and rationale for a lot of projects now being done, including work on measurement and modeling, new architecture explorations, and security. Many of the specific ideas in this proposal are not new. What is new is the proposed framework, the vision of how these ideas can be drawn together to help conceive a network of the future.

Introduction

In some respects, networking seems “done”. The Internet is a success, every personal computer now comes network-ready, email and the Web are commonplace, and the technology has a feeling of maturity.

But the sense of maturity is false. For all sorts of reasons, it is time now to revisit our fundamental assumptions, to envision what networks must do in ten years, and to propose some radical changes to our current view of networks.

False sense of maturity

We must not be lulled by the sense of the network’s maturity. The apparent maturity of the network derives from the maturity of the personal computer, which grew up at the same time as the Internet. But we are seeing an explosion of networked devices that look nothing like the PC. These include portable devices, networked vehicles, and embedded sensors and controllers. Ten years from now, the most common networked end-node will be an embedded processor, not a human-centered “personal” computer.

An architecture with dead ends

Another reason for a radical rethinking is that the Internet, in rushing to its success, may have gone into some dead ends from which it will be hard to extract the future without new approaches. Once we look with unbiased eye at the Internet, its success is accompanied by many frustrations.

<u>Fault isolation:</u> When the Internet fails, it is almost impossible for the end user to tell what has happened, to figure out who should be notified, or what to do to correct the fault. The distributed operation of the Internet means that the network operators may not be much better off.
<u>Routing:</u> The dynamic routing of the original network did not take into account administrative and policy constraints, so routing today is more and more defined by static

policy tables. These make the network brittle to failure, hard to configure, and even harder to reason about globally.
Configuration: The devices in the Internet today, such as routers, are manually configured and managed. This approach is tedious and error-prone. We lack high-level tools for management and configuration.
Isolation of application from technology: A strength of the Internet is that it can be made to run over essentially any network technology. This is because it makes a very simple use of this technology. The drawback of this approach is that if the underlying technology has special features useful to particular applications, or in special conditions, there is no way to "get at " these features.

These lacks are not an accident—it is not that the Internet designers just didn't get around to these problems yet. These problems arise precisely because of the simplicity and transparency of the core of the network. The core of the network has, by design, no idea what the applications above it are doing, or what the traffic patterns on the net signify. That is why the network can be open to the easy deployment of new applications. But exactly because the core does not "know what is going on", it cannot easily detect when something has gone wrong. It cannot easily develop high-level models of desired behavior without knowing what the desired behavior is. It cannot distinguish a new sort of attack on a host from a new application. So to get to the next level of network functionality and utility, a new design for the network is needed that marries the transparency and utility of the Internet with a new approach to management, control, deployment and configuration.

A very short history of networking

The telephone system is a very successful network. It is marked by a complex and powerful core, very simple edge-devices (the phones themselves) and a clear model, built into the core of the network, of what the network is for. The telephone system is designed knowing that the purpose of the network is to carry telephone calls. Since the network has a model of what its purpose is, it can do diagnosis and fault isolation of the total system.

The recent success story is the Internet. The design of the Internet is almost the exact opposite of the phone system. The core of the Internet is very simple, and has no model of the applications that run over it. The applications are mostly embodied in very complex end-nodes—servers, personal computers and so on. The simplicity of the core allows new applications to be deployed at will, but mean that the core cannot detect when problems arise at the application layer.

As the Internet matures, it is taking on some of the complexity of the telephone system. At the same time, the operators of the telephone system are looking at the Internet as a possible core for a next generation telephone system. So there seems to be a debate that frames the future network as some compromise between the Internet and the telephone system. This debate is flawed. We should envision a future network that is not a simple hybrid of these two networks, but instead one that takes the good ideas from each, but adds radical new ideas to the mix. New ideas are required because a simple hybrid of the telephone system and the Internet cannot resolve the essential incompatibility between the two—the fundamental difference between the simple transparent core of the Internet and the complex but application-specific core of the phone company. These ideas are the opposite of each other, and a new approach, not a simple blending, is required to get the good features of both.

A proposal: the Knowledge Plane

The simple and transparent data transport plane of the Internet has been a success, so this idea should not be discarded. Instead of trying to compromise and add some measure of application knowledge to the data transport plane, a better alternative is the construction of a separate **knowledge plane** in the future network. The knowledge plane is a distributed and decentralized construct within the network that gathers and aggregates information about network operation, and provides an integrated view and a consistent set of control signals.

Edge-device participation: Since applications reside in the edge-devices of the network, these devices are in a good position to detect a broad class of failures, and determine if corrective action is necessary. But today each edge-device is isolated, with no way to determine if other devices are having trouble, if the cause of the trouble has already be identified and flagged for correction, and so on. The knowledge plane is a distributed capability for edge-nodes as well as the network to add information, and query the aggregated information about the state of the network.

High-level assertions about intended operation: The knowledge plane is a framework to log and process high-level assertions about what the net is supposed to be doing. It is a distributed context for assertions about goals, operational constraints and alternative approaches. The data gathered there can be used to validate low level algorithms, detect operational flaws, propose (or implement) low-level reconfiguration, and discover inconsistent high-level assertions.

Aggregation of independent measurement: there are many tools being used today that measure some aspect of the network—latency, loss rate, route instability, forwarding failures and so on, to try to determine if there are problems of various sorts. The knowledge plane will provide a framework to combine these observations into a global view. The knowledge plane will have enough information to detect inconsistent measurements (which might signal a broken or compromised network region). Measurements can be compared to the high-level model of what ought to be observed.

From one perspective, much of the knowledge in the knowledge plane already exists in the network today, in the heads of network operators, who form a global network hooked together by email and phone. From this perspective, the knowledge plane has the goal of automating more of network deployment and operation, and reducing the labor-intensive manual procedures of today. Some knowledge may be online, but is considered proprietary by network operators. The knowledge plane represents a way to develop an overall view about what is actually happening, even if some operators do not want to reveal their internal information.

While the network of the future must continue to have a simple, transparent data plane, it is possible to have application-specific components in the knowledge plane. That is, applications as well as the transport and packet layer can make assertions and ask questions of the knowledge plane. So it is possible to develop an application specific view of “what is going on”, similar to the telephone network with its application-specific core.

There are many problems to be solved to build the knowledge plane, similar to the problems that will have to be solved to build any sort of decentralized cognitive system.

- How can a decentralized mesh of facts integrate these to produce “knowledge”?
- How can inconsistent assertions about facts be reconciled?
- When a single failure or change produces many new assertions, how can this flood be controlled?
- How can the knowledge plane protect itself from malicious assertions?
- What are the abstractions that are supported by the knowledge plane? How specific are they to “networking”?
- How can an operator describe configuration and routing preferences at a high-level?
- Who is allowed access to ask which questions of the knowledge plane?

It should be obvious that the knowledge plane is a building block for a network that is more reliable, more robust, and more secure. Much work will be required to realize this objective, but if the network has a high-level model of what is “supposed to be happening”, if it has captured the idea of goals and intentions, then it should be more feasible to detect when things are going wrong, and have a better chance of preventing and correcting problems.

Specific projects

Here are some projects that illustrate the power of the knowledge plane, that sharpen the vision of what the knowledge plane must do, and that yield practical benefit as well as long-term insight.

Why has it failed?

Build a tool that runs on an end node, and performs diagnosis when there is a failure. The diagnostics can check out functions at all levels, from packet forwarding to application function. Initially, this tool would give the user an explanation of what had gone wrong, in terms that are meaningful to the user. There are several current research projects that this goal could build on.

Once the initial tool is constructed, the next stage is for the tool to add assertions to the shared knowledge plane about what it has discovered, and ask the knowledge plane for relevant information. This allows all the users on the network collectively to build a global view of network and service status. This work can be combined with other measurement efforts now going on across the Internet that attempt to build an overall model of network status. Network operators have the option of adding additional facts to the knowledge plane about known failures; in the ideal, a user who trips over a problem might not just get back diagnostic information, but information from the provider about when the problem will be resolved.

This project will require the creation of a distributed platform on which the knowledge plane is instantiated, the definition of the "knowledge abstractions" that are the language in which failures and status are logged, and the solution of the scaling problems that arise when a failure triggers a multitude of end-node tests and assertions made into the knowledge plane.

Do what I mean!

Build a distributed configuration manager for a region of the Internet that accepts high-level assertions, at the administrative level, about how the components of a network are supposed to arrange themselves, and guides the actual configuration accordingly. The distributed manager should have enough understanding of low level configuration to detect if the network is properly configured according to the high-level constraints, to detect if a better configuration alternative is available, and to detect if the system appears to be corrupted. The system must be able to deal with different assertions made by different parties, and either compose them or detect that they are inconsistent.

Examples include controlling the deployment of a consumer network in the home, an ad hoc network in support of a rapid deployment force, or a network for a small business. A first experiment should focus on a small network, since there will actually be many hard problems to solve, even at that scale, and there will be practical short-term payoff at this scale.

The information in the configuration manager is a part of the knowledge plane, since it provides a high-level model of what is supposed to happen, against which actual observations of what is happening can be checked.

Previous attempts to do "high-level network management" have not been very successful; one possible reason is that previous projects have not been able to find the correct high-level abstractions. One of the central tenets of "cognitive computing" is that there exist suitable ways to abstract detailed behavior, and to talk about goals, plans, constraints and methods at a high level. High-level network management is thus an excellent challenge for cognitive computing, since it will raise many of the difficult issues, and it will have immediate practical benefits as it is successful. Successful accomplishment of this project could lead to substantial reductions in manpower needed to configure and operate networks.

Shadow routing in the knowledge plane

Today, routing computes a "best" route to Internet destinations. But there is no computation of acceptable alternative routes. More interesting, there is no computation of the best route *from* any address. The assumption is that if a packet from a given source arrives over a given interface, then this must be a legitimate path from that source. It is this assumption that allows malformed routing assertions to deflect packets to distant parts of the network, allows DdoS attacks to contain arbitrary source addresses, and so on.

This proposal is to complement the actual routing protocols of the Internet with a "shadow" computation that runs in the knowledge plane. Autonomous Regions (AS's) may make assertions about what connections are acceptable for traffic flowing to and from them. These assertions can be combined with the BGP assertions seen at various places in the network to build up a model of what acceptable routing options are. In this way, the knowledge plane learns what is *supposed* to be happening, and what the range of acceptable routing alternatives is. This can be used to detect potentially invalid BGP routing assertions, false source addresses, operator misconfigurations, and so on.

A Web of Trust and better Spam control

As a first experiment in adding application information to a knowledge plane, design an enhanced email architecture that can control the delivery of unwanted email (spam). The proposed approach is a combination of a number of techniques that are being tried today, together with the addition of a "web of trust" to the knowledge plane. The proposal is as follows. Use some mechanism by which senders can sign mail, so that the "from" field cannot be forged. Have each user build up a private database of users that he is prepared to talk to. If a user receives mail from an unknown sender, build a bottom-up peer-to-peer mesh of parties that agree to trust each other, into which a query can be launched to try to validate the identity (friend or spammer) of the unknown sender. If the sender is not known to any of the receiver's "circle of friends", treat it tentatively as spam. A second layer of shared knowledge can then be used to see if anyone has verified that this message is spam.

This proposal would require the investigation of trust models, and the use of scalable techniques (such as the so-called "small world" models) to search a web of trust. A web of trust, once constructed, could be used to improve the coherence of other applications. For example, if a user is injecting information into the knowledge plane about a possible network fault (the "Why has it failed?" project), other nodes will be more inclined to believe an assertion from a node they trust. If a trusted node is generating spurious information, this may be a signal that it has been corrupted. So trust can be both exploited and validated across applications.

Knowledge-based intrusion detection

There are a number of projects (and a number of products) that perform some sort of analysis to detect network intrusions. In general they look for patterns in data observed somewhere in the network. The current generation of these tools trigger both false positives and false negatives. It has been hypothesized that a next generation of tools for intrusion detection will require that observations from several points in the network will have to be correlated, in order to get a more robust and useful signal. The development of the knowledge plane provides a basis to implement this data gathering and correlation.

This use of the knowledge plane does not require that the plane be global in scope, but just implemented inside a region. So a project on intrusion detection might represent a localized way to experiment with the concept.