4D Network Architecture: Modularizing Functionality

Ben Lascurain

University of Louisville

4D Networks: Modularizing Network Functionality

Networks require careful configuration, are prone to human error, and have become fragile due to patchy, incremental dependencies. Because it is impossible to know which direction network advancements will lead, it is important to have an architecture that is robust and easily adaptable. The direction that I hope to see networks trending is towards true modularization. Using a 4D network is one way to accomplish this, by drawing clear divisions of duties for the network planes, while wiping away cascading dependencies that inhibit progress and innovation.

Networks have evolved as the need for data and resource sharing has increased. Much of this development was reactive, dealing with issues by creating ad hoc solutions. Early creators and developers of network standards couldn’t have foreseen the extent that networks would be used in day to day functions, and the major flaws in networks demonstrate this lack of planned growth.

Some key difficulties with current networks include configuration fragility, nodes with inconsistent views of the topology, security implemented through static reachability configurations in decision elements (DE), and inefficient transportation of packets due to imbedded path computation logic in protocols. These are just some of the more pervasive issues, but other quarks can quickly become time consuming and frustrating to network administrators because of the decentralized nature of network configurations.

4D network protocol tries to address these issues, and is comprised of four planes**:** the decision, discovery, dissemination, and data plane. However, to understand why 4D networks have this design, an understanding of its three guiding principles is needed. These three guiding principles are: network-level objectives, network-wide views, and direct control.

**Three Guiding Principles of 4D Networks**

4D Networks were designed to better meet network-level objectives, create network-wide views of topology that are consistent and accurate, and separate decision logic from transport protocols allowing more powerful routing algorithms. These three principles should be at the forefront when constructing an efficient and effective network.

**Network-Level Objectives**

Each network has specific objectives concerning things such as providing security without any breaches, or maintaining load balances at no greater than 85% between all servers. Today’s networks rely on proper configuration of each switch and router to reach these network-level objectives. This leaves networks vulnerable to human error, potentially missing configurations required to achieve these objectives. Even worse, this could leave security vulnerabilities undetected until after a breach occurs unless each node is regularly audited.

A 4D network avoids individual component configurations by centralizing the decision logic and network topology state maintenance. By centralizing these functions, 4D networks are able to implement consistent protocols across all autonomous devices, reducing chances of human error. Having the network’s topology mapped in one central location allows for a consistent and accurate state to be passed and updated among all autonomous systems.

**Network-Wide View.** A network-wide view means that throughout the entire network there is one version of topology distributed to all elements. This single version is updated with information collected by each node, but updated and disseminated through one central source. This is an important point because in today’s networks, routing and forwarding tables aren’t always consistent. Forwarding tables are kept and updated by each individual device and don’t proactively share their updated information that they discover with nearby devices. This creates multiple versions of the topology and ultimately leads to slower transport times.

By centralizing the record of the topology state, a consistent view can be used by all autonomous systems. In a 4D network, each switch and router performs discovery functions that actively collect data on the capabilities and status of nearby devices. This ensures routes are available, reducing failed or delayed packet deliveries. The data collected from discovery would not be immediately used by the router/switch to update their own network state, but rather it would be sent to the decision plane, analyzed, and any updates would be passed to all routers and switches. The discovery process will be described in more depth, but for now this should help demonstrate the importance of a network-wide view.

**Direct Control.** In a 4D network, creating and maintaining a network-wide view and meeting network-level objectives are accomplished through implementing direct control over the other planes. Direct control of the traffic is implemented in a 4D network by divorcing the decision logic from routers and switches, and instead handling all decision logic in the DEs. Then the routing commands are then passed from the DEs to the data plane, where it is executed. This allows for a single point of configuration and view of that can be leveraged across all devices.

Through direct control, management of the routing and loads are directed from a centralized source. This gives the network greater stability by removing much of the required router and switch configurations that would otherwise be essential.

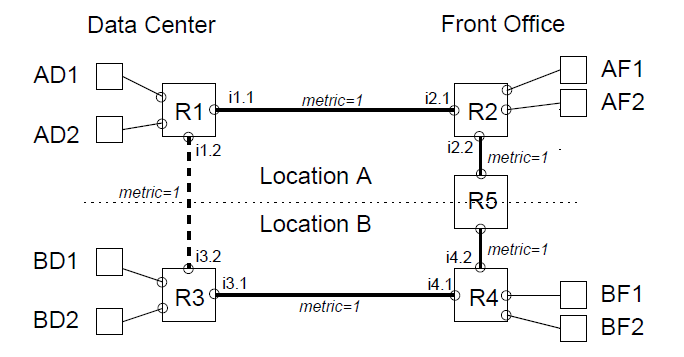
To illustrate the of the issues of relying on routing and filtering protocols to implement security measures and reachability objectives, we’ll examine this simple hypothetical, as laid out in *A Clean Slate 4D Approach to Network Control and Management* (Greenburg et al., 2005). Imagine a company with two sales divisions, division A and division B. Both of these divisions have front offices (AF and BF), and both have their own respective databases (AD and BD). Now these divisions, although part of the same company, do not want their databases accessible to the other division’s front offices. So AF1 shouldn’t be able to access BD1 or BD2, and BF2 shouldn’t be able to access AD1 or AD2, but both front offices should be reachable to the other division’s front office (AF1 should be able to reach BF2). These traffic rules would be implemented by filtering and dropping BF1/BF2 packets that reach Interior Gateway Protocol (IGP) i1.1 and dropping packets from AF1/AF2 that reach IGP i3.1. This will effectively prevent either front office from reaching either opposing offices’ database. But let’s say that the company wants a link between the two division databases, so now BD1/BD2 can reach AD1/AD2. If these new gateways, i3.2 and i1.2, aren’t configured with the proper filters, a backdoor for either front office to access the other’s database is now open. AF1 can go R2 (router two) to R1 to R3. This violates the security measures we have for the entire network, but was overlooked because each router requires proper configuration to restrict reachability. This error may be obvious and not likely to happen in a network so simple, but if we scale this to the size of many of today’s networks and consider all of the overlapping and complex protocols, it’s easy to see where a mistake could be made.

Figure 1. Schematic of a simple network layout that demonstrates fragility of implementing network objectives through protocol configuration. Reprinted from “A Clean Slate 4D Approach to Network Control and Management,” by A.G. Greenberg, 2005, *Computer Communication Review*, 35, p. 41-54

**4D Network’s Four Planes**

To accomplish network-level objectives, maintain a network-wide view, and have direct control over the elements from a central point 4D networks implement four planes. The discovery, dissemination, data, and the decision planes accomplish networking in a modularized way. These planes specialize each layer with no overlap of duties. This makes a 4D network an ideal architecture to encourage improvements one layer at a time, without worrying about cascading dependency issues once the changes are implemented. The next few sections will dig into each layer’s purpose, benefits, and some possible issues with implementing each plane.

**Discovery Plane**

In the discovery plane each switch and router are responsible for proactively gathering information to update the central routing and forwarding table that is maintained in the decision plane. The information network nodes gather should be specific and valuable to achieving network-level objectives. For example, they should collect data about surrounding devices’ capabilities, other neighboring devices that it connects to and their capabilities, and the capacity of the interfaces it owns. More importantly, the discovery plane proactively seeks out and automatically updates the decision plane on any changes. This topological view is forwarded to the decision plane without updating each node’s network states, so they remain consistent. Only after the decision plane evaluates all available data does it update every switch and router’s tables.

**Dissemination Plane**

The dissemination plane is composed of the pathways that will carry decision elements to switches and routers. Without this dissemination plane, the decision plane would have no way to configure the data plane because there are no preconfigured protocols (besides a credential key) on any of the devices in a 4D network. The dissemination plane utilizes the same pathways as the data plane but isn’t restricted by the same protocols as the data plane because it needs to freely transmit decision elements, configurations, and the topology state throughout the network.

Now this poses an obvious threat to planning security for a 4D network. Yan et al. (2007) experimented with 4D Networks by creating the Tesseract project, which addressed this security issue by bootstrapping security before initial deployment through uploading network decision element certificates via USB. After uploading the certificates, the switch constructed a device ID and a private/public key. It then stored its device ID, key pair, and decision element certificates in nonvolatile memory. It then encrypted this information and stored it back onto the USB, which was then uploaded into the decision element. This effectively created a secret channel to each switch and a verifiable way to know which device IDs are valid before deployment. This is just one solution to securing a centralized network with a free-flowing dissemination plane, but as we can see, it is considerably less effort than configuring each network.

Another issue with the dissemination plane could be the overhead requirements to all of the commands, protocols, and updated tables and views being distributed. What sort of strain would this impose on the physical capabilities of networks, and how much of the network’s capacity to transport data will be lost to supporting decision plane messages? Requiring a network to support additional traffic to make decisions seems to be a step in the wrong direction, but with bandwidth rapidly increasing rapidly, while maintaining affordability, this overhead may be less important than what the network can gain.

**Data Plane**

The data plane is responsible for handling packets based on the network state that is passed from the decision plane. The network state contains multiple pieces of information that are used by the data plane such as the forwarding table, packet filters, link-scheduling weights, and queue-management parameters, as was laid out in *A clean slate 4D approach* (Greenberg et al., 2005). In order to fit into the centralized architecture that 4D utilizes, the data plane cannot act autonomously from the decision plane. This tethering of decision logic and network objectives ensures consistency across devices. A picture of the importance of separating, or modularizing, each function should be clear. By having the data plane removed from decision logic, we can focus our software development in one centralized area instead of customizing the configuration and patches on each switch and router.

To think that separating out the decision logic from the data plane is without room for concern would be naive. By centralizing all decision logic in the decision plane, we put a much higher value on a single penetration point for attackers. Now, instead of having to crack each switch or router to gain full network access, an attacker would only have to penetrate the decision layer and alter its commands. This puts a larger target on the back of the decision plane, but also allows networks to limit their points of vulnerabilities to a centralized point.

**Decision Plane**

The true benefit of the decision plane isn’t fully understood until it is compared with current packet forwarding configuration procedures. In current network architecture, packet filters are configured on each interface, can only be manually updated, and do not share the configuration with surrounding devices (Xie et al., 2004). This makes reconfiguring routers very time consuming and costly to organizations. Furthermore, a task as simple as redirecting traffic flow for a scheduled maintenance would require manual configuration of a temporary path. All this is avoided in a 4D network because of the centralized nature of the configurations being in the decision element. The decision plane has a clear view of the entire network’s state and knows what sort of filtering and forwarding protocols need to be implemented in order to achieve network-level objectives. This couldn’t be achieved so fluidly in a standard network because each network element acts autonomously with its own perception of the current state.

As you could imagine, having a centralized decision plane would mean that there would need to be redundancies in order to ensure continuous service during a failure in a decision element. As evaluated in *On the Design of Clean-Slate Network Control and Management Plane* (pp. 3), a likely solution would be to have several central DEs, operating as components of the same decision plane. Each of these DEs would have specific nodes under their direct control with redundant control permissions striped and mirrored across other central DEs. This method looks similar to how RAID is implemented. This would not only increase fault tolerance (like RAID), but also could improve performance by distributing the load and allowing parallel access.

**Conclusion**

4D network design addresses the common issues of today’s networks but will likely lead to unforeseen problems. The real question is will these unforeseen problems be better addressed in a 4D environment that has modularized its planes to be more robust and adaptable. 4D networks may not be the ultimate solution for the issues of our current dependency entangled networks, but they are a step in the right direction by prioritizing network-level objectives, network-wide views, and direct control. At the very least, 4D networks will provoke further thought and analysis that challenges the status quo and welcomes innovation.

References

Greenberg, A.G., Maltz, D.A., Hjálmtýsson, G., Myers, A., Rexford, J., Xie, G.G., Yan, H., Zhan, J., & Zhang, H. (2005). A Clean slate 4D approach to network control and management. Computer Communication Review, 35, 41-54.

Greenberg, A.J., Hjalmtysson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G.G., Yan, H., Zhan, J., & Zhang, H. (2005). Refactoring network control and management: a case for the 4D architecture.

Iqbal, H., & Znati, T. (2009). On the design of clean-slate network control and management plane.

Xie, G.G., Zhan, J., Maltz, D.A., Zhang, H., Greenberg, A.G., & Hjálmtýsson, G. (2004). Routing design in operational networks: a look from the inside. SIGCOMM.

Yan, H., Maltz, D.A., Ng, T.S., Gogineni, H., Zhang, H., & Cai, Z. (2007). Tesseract: a 4D network control plane. NSDI.