Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2013

A. Retana
Cisco Systems
R. White
Verisign

March 14, 2013

A Framework for Measuring Network Complexity
draft-retana-network-complexity-framework-00.txt

Abstract

Network architecture revolves around the concept of fitting the design of a network to its purpose; of asking the question, "what network will best fit these needs?" A part of fitting network design to requirements is the problem of complexity, an idea often measured by "seat of pants" methods. When would adding a particular protocol, policy, or configuration be "too complex?" This document suggests a series of continuums along which network complexity might be measured. No suggestions are made in measuring complexity for each of these continuums are provided; this is left for future documents.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 15, 2013.

Copyright Notice

Table of Contents

1. Introduction

Network complexity is a systemic, rather than component level, problem; complexity must be measured in terms of the multiple moving parts of a system, and complexity may be more than the complexity of the individual pieces, examined individually, might suggest. There are two basic ways in which systemic level problems might be addressed: interfaces and continuums. In addressing a systemic problem through interfaces, we seek to treat each piece of the system as a "black box," and develop a complete understanding of the interfaces between these black boxes. In address a systemic problem as a continuum, we seek to understand the impact of a single change or element to the entire system as a set of tradeoffs. While network complexity can profitably be approached from either of these perspectives, the authors of this document have chosen to approach the systemic impacts of network complexity from the perspective of continuums of tradeoffs. In theory, modifying the network to resolve one particular problem (or class of problems) will add complexity which results in the increased liklihood (or appearance) of another class of problems. Discovering these continuums of tradeoffs, and then determining how to measure each one, become the key steps in understanding and measuring systemic complexity in this view.

This document proposes five such continuums; more may be possible. Others may be added into this document in future revisions, or documented in other drafts, as circumstances dictate.

- Control Plane State verses Optimal Forwarding Paths (or it's opposite measure, stretch)
- Configuration State verses Failure Domain Separation
- Policy Centralization verses Optimal Policy Application
- Configuration State verses Per Hop Forwarding Optimization
- Reactivity verses Stability

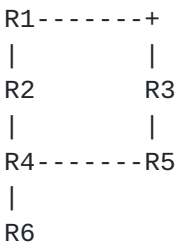
Each of these continuums is described in a separate section of this draft.

2. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", "OPTIONAL" in this document are to be interpreted as in [RFC2119].

3. Control Plane State verses Optimal Forwarding Paths (Stretch)

Control plane state is the aggregate amount of information carried by the control plane through the network in order to produce the forwarding table at each device. Each additional piece of information added to the control plane --such as more specific reachability information, policy information, additional control planes for virtualization and tunneling, or more precise topology information-- adds to the complexity of the control plane. This added complexity, in turn, adds to the burden of monitoring, understanding, troubleshooting, and managing the network. Removing control plane state, however, is not always a net positive gain for the network as a system; removing control plane state almost always results in decreased optimality in the forwarding and handing of packets travelling through the network. This decreased optimality can be termed stretch, which is defined as the difference between the absolute shortest (or best) path traffic could take through the network and the path the traffic actually takes through the network. Stretch is expressed as the difference between the optimal and actual path. The figure below provides and example of this tradeoff.



Assume each link is of equal cost in this figure, and:

- R4 is advertising 192.0.2.1/32 as a reachable destination not shown on the diagram
- R5 is advertising 192.0.2.2/32 as a reachable destination not shown on the diagram
- R6 is advertising 192.0.2.3/32 as a reachable destination not shown on the diagram

For R1, the shortest path to 192.0.2.3/32, advertised by R6, is along the path [R1,R2,R4,R6]. Assume, however, the network administrator decides to aggregate reachability information at R2 and R3, advertising 192.0.2.0/24 towards R1 from both of these points. This reduces the overall complexity of the control plane by reducing the amount of information carried past these two routers (at R1 only in this case). Aggregating reachability information at R2 and R3, however, has the impact of making both routes towards 192.168.2.3/32 appear as equal cost paths to R1; there is no particular reason R1 should choose the shortest path through R2 over the longer path through R3. This, in effect, increases the stretch of the network. The shortest path from R1 to R6 is 3 hops, a path that will always be chosen before aggregation is configured. Assuming half of the traffic will be forwarded along the path through R2 (3 hops), and half through R3 (4 hops), the network is stretched by ((3+4)/2) - 3), or .5, a "half a hop."

Traffic engineering through various tunneling mechanisms is, at a broad level, adding control plane state to provide more optimal forwarding (or network utlization). Optimizing network utilization may require detuning stretch (intentionally increasing stretch) to increase overall network utilization and efficiency; this is simply an alternate instance of control plane state (and hence complexity) weighed against optimal forwarding through the network.

4. Configuration State verses Failure Domain Separation

A failure domain, within the context of a network control plane, can be defined as the set of devices impacted by a change in the network topology or configuration. A network with larger failure domains is more prone to cascading failures, so smaller failure domains are normally preferred over larger ones. The primary manes used to limit the size of a failure domain within a network's control plane is information hiding; the two primary types of information hidden in a network control plane are reachability information and topology information. An example of aggregating reachability information is summarizing the routes 192.0.2.1/32, 192.0.2.2/32, and 192.0.2.3/32 into the single route 192.0.2.0/24, along with the aggregation of the metric information associated with each of the component routes. Note that aggregation is a "natural" part of IP networks, starting with the aggregation of individual hosts into a subnet at the network edge. An example of topology aggregation is the summarization of routes at a link state flooding domain boundary, or the complete failure to advertise topology information in a distance-vector protocol.

While limiting the size of failure domains appears to be an absolute good in terms of network complexity, there is a definite tradeoff in configuration complexity. The more failure domain edges created in a network, the more complex configuration will become. This is particularly true is redistribution of routing information between multiple control plane processes is used to create failure domain boundaries; moving between different types of control planes causes a loss of the consistent metrics most control planes rely on to build loop free paths. Redistribution, in particular, opens the door to very destructive positive feedback looks within the control plane. Examples of control plane complexity caused by the creation of failure domain boundaries include route filters, routing aggregation configuration, and metric modifications to engineer traffic across failure domain boundaries.

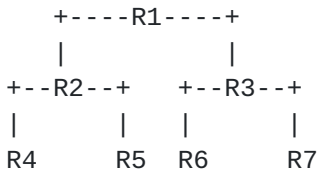
Returning to the network described in the previous section, aggregating routing information at R2 and R3 will divide the network into two failure domains: (R1,R2,R3), and (R2,R3,R4,R5). A failure at R5 should have no impact on the forwarding information at R1. A false failure domain separation occurs, however, when the metric of the aggregate route advertised by R2 and R3 is dependent on one of the routes within the aggregate. For instance, if the metric of the 192.0.2.0/24 aggregate is taken from the metric of the component 192.0.2.1/32, then a failure of this one component will cause changes in the forwarding table at R1 --in this case, the control plane has not truly been separated into two distinct failure domains. The added complexity in the illustration network would be the management of the configuration required to aggregate the contorl plane information, and the management of the metrics to ensure the control plane is truly separated into two distinct failure domains.

Replacing aggregation with redistribution adds the complexity of managing the feedback of routing information redistributed between the failure domains. For instance, if R1, R2, and R3 were configured to run one routing protocol, while R2, R3, R4, R5, and R6 were configured to run another protocol, R2 and R3 could be configured to redistribute reachability information between these two control planes. This can split the control plane into multiple failure domains (depending on how, specifically, redistribution is configured), but at the cost of creating and managing the redistribution configuration. Futher, R3 must be configured to block routing information redistributed at R2 towards R1 from being redistributined (again) towards R4 and R5.

5. Policy Centralization verses Optimal Policy Application

Another broad area where control plane complexity interacts with optimal network utilization is Quality of Service (QoS). Two specific actions are required to optimize the flow of traffic through a network: marking and Per Hop Behaviors (PHBs). Rather than examining each packet at each forwarding device in a network, packets are often marked, or classified, in some way (typically through Type of Service bits) so they can be handled consistently at all forwarding devices. Packet marking policies must be configured on specific forwarding devices throughout the network. Distributing marking closer to the edge of the network necessarily means configuring and managing more devices, but produces optimal forwarding at a larger number of network devices. Moving marking towards the network core means packets are marked for proper handling across a smaller number of devices. In the same way, each device through which a packet passes with the correct PHBs configured represents an increase in the consistency in packet handling through the network as well as an increase in the number of devices which

must be configured and managed for the correct PHBs. The network below is used for an illustration of this concept.



In this network, marking and PHB configuration may be configured on any device, R1 through R7. Assume marking is configured at the network edge; in this case, four devices, (R4,R5,R6,R7), must be configured, including ongoing configuration management, to mark packets. Moving packet marking to R2 and R3 will halve the number of devices on which packet marking configuration must be managed, but at the cost of consistent packet handling at the inbound interfaces of R2 and R3 themselves. Thus reducing the number of devices which must have managed configurations for packet marking will reduce optimal packet flow through the network. Assuming packet marking is actually configured along the edge of this network, configuring PHBs on different devices has this same tradeoff of managed configuration verses optimal traffic flow. If the correct PHBs are configured on R1, R2, and R3, then packets passing through the network will be handled correctly at each hop. The cost involved will be the management of PHB configuration on three devices. Configuring a single device for the correct PHBs (R1, for instance), will decrease the amount of configuration management required, at the cost of less than optimal packet handling along the entire path.

6. Configuration State verses Per Hop Forwarding Optimization

The number of PHBs configured along a forwarding path exhibits the same complexity verses optimality tradeoff described in the section above. The more types of service (or queues) traffic is divided into, the more optimally traffic will be managed as it passes through the network. At the same time, each class of service must be managed, both in terms of configuration and in its interaction with other classes of service configured in the network.

7. Reactivity verses Stability

The speed at which the network's control plane can react to a change in configuration or topology is an area of widespread study. Control plane convergence can be broken down into four essential parts:

o Detecting the change

- Propagating information about the change
- Determining the best path(s) through the network after the change
- Changing the forwarding path at each network element along the modified paths

Each of these areas can be addressed in an effort to improve network convergence speeds; some of these improvements come at the cost of increased complexity.

Changes in network topology can be detected much more quickly through faster echo (or hello) mechanisms, lower layer physical detection, and other methods. Each of these mechanisms, however, can only be used at the cost of evaluating and managing false positives and high rates of topology change. If the state of a link change can be detected in 10ms, for instance, the link could theoretically change state 50 times in a second --it would be impossible to tune a network control plane to react to topology changes at this rate. Injecting topology change information into the control plane at this rate can destabalize the control plane, and hence the network itself. To counter this, most fast down detection techniques include some form of dampening mechanism; configuring and managing these dampening mechanisms represents an added complexity that must be configured and managed.

Changes in network topology must also be propagated throughout the network, so each device along the path can compute new forwarding tables. In high speed network environments, propagation of routing information changes can take place in tens of milliseconds, opening the possibility of multiple changes being propagated per second. Injecting information at this rate into the contral plane creates the risk of overloading the processes and devices participating in the control plane, as well as creating destructive positive feedback loops in the network. To avoid these consequences, most control plane protocols regulate the speed at which information about network changes can be transmitted by any individual device. A recent innovation in this area is using exponential backoff techniques to manage the rate at which information is injected into the control plane; the first change is transmitted quickly, while subsequent changes are transmitted more slowly. These techniques all control the destabalilzing effects of rapid information flows through the control plane through the added complexity of configuring and managing the rate at which the control plane can propagate information about network changes.

All control planes require some form of algorithmic calculation to find the best path through the network to any given destination.

These algorithms are often lightweight, but they still require some amount of memory and computational power to execute. Rapid changes in the network can overwhelm the devices on which these algorithms run, particularly if changes are presented more quickly than the algorithm can run. Once the devices running these algorithms become processor or memory bound, it could experience a computational failure altogether, causing a more general network outage. To prevent computational overloading, control plane protocols are designed with timers limiting how often they can compute the best path through a network; often these timers are exponential in nature, allowing the first computation to run quickly, while delaying subsequent computations. Configuring and managing these timers is another source of complexity within the network.

Another option to improve the speed at which the control plane reacts to changes in the network is to precompute alternate paths at each device, and possibly preinstall forwarding information into local forwarding tables. Additional state is often needed to precompute alternate paths, and additional algorithms and techniques are often configured and deployed. This additional state, and these additional algorithms, add some amount of complexity to the configuration and management of the network. In some situations (for some topologies), a tunnel is required to pass traffic around a network failure or topology change. These tunnels, while not manually configured, represent additional complexity at the forwarding and control planes.

8. Conclusion

This document describes various areas of network and design where complexity is traded off against some optimization in the operation of the network. This is (by it's nature) not an exhaustive list, but it can serve to guide the measurement of network complexity and the search for other areas where these tradeoffs exist.

9. Security Considerations

None.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Alvaro Retana
Cisco Systems
2610 Wycliff Road
Raleigh, NC 27607
USA

Email: aretana@cisco.com

Russ White
Verisign
12061 Bluemont Way
Reston, VA 20190
USA

Email: russw@riw.us