Uptime Considerations for Network Switches

A modest attempt to compare Stackable and Modular Uptime

Contents

Overview

In this document, I attempt to compare the Allen Institute's Ethernet switch failure rate in 2016 against published MTBF numbers for our Ethernet switch platform.

Summary

The published MTBF numbers are not sufficiently detailed to perform an accurate calculation.

For example, the published MTBF number for our IDF Ethernet switch applies only to a stand-alone configuration – it does not include the additional parts needed to 'Stack' them nor the increased failure rate (from complexity) which occurs when one 'Stacks'.

More importantly, the published MTBF numbers only address hardware failure, whereas uptime is driven by software bugs. Most of our Ethernet reliability issues to date have been caused by software bugs … and we are mostly interested in uptime, not hardware failure per se.

All that being said, I can at least estimate a lower bound for hardware reliability, given our installation.

Predicted Ethernet IDF Hardware Failure Rate = At least 1.7 Failures per Year
In other words, if we experienced 1.7 IDF failures per year, we would be doing pretty darn good.

Actual Ethernet IDF Hardware Failure Rate in 2016 = (3)
We suffered (3) partial failures of power supplies in 2016: the PoE function failed for (3) switches, so phones (and their attached computers) went down; other devices were unaffected.

⇨ In the narrow world of hardware failure, our 2016 experience fits comfortably with published MTBF rates, i.e. our experience is *normal*.

Historical Comparison

Recall that historically, the Institute deployed (mostly) a high-end Ethernet platform (Catalyst 4500). Here, I compare expected hardware failure rates between the two environments: Fremont / Wallingford versus Lake Union.

Predicted Hardware MTBF

	Fremont / Wallingford	**Lake Union**
MTBF using Cat4500[1]	1 failure every 20 years	1 failure every 4 years
MBF using Cat2960X[2]	Almost 1 failure every year	At least 1.7 failures every year

This table illustrates the improvement in hardware reliability which the Modular platform (Cat4500) deliver, compared to the Stackable (Cat2960X) platforms. Naturally, there is no free lunch – the Modular platforms cost more.

[1] Assume (4) Cat4500.
[2] Assume (22) Cat2960X

Background

Our Institute historically occupied a handful of rented buildings in the Fremont / Wallingford area; in 2015, we consolidated into a single building (aka 'B43') in the Lake Union area.

B43 uses (6) IDFs to deliver access-layer cabling. Those IDFs are populated with Ethernet switches as follows.

Installation

IDF	Catalyst 2960X Count	Notes
P1N	2	
2S	8	
3N	8	
4S	11	(8) Member Stack + (3) Member Stack
5N	8	
6S	8	

In 2016, we experienced (3) major IDF-related service disruptions.

Service Disruption

Scope	Description	Root Cause
2S-IDF	(3) Members quit delivering PoE, shutting off phones (and their attached PCs)	Hardware failure in the PoE function of
4S-IDF	(9) Members hung intermittently, causing wide-spread workstation isolation	Firmware bug in the PoE function of po
5N-IDF (now 6S-IDF)	Intermittent port disables & Stack Crashes	Unknown[3]

[3] Possibly a software bug, possibly an intermittent hardware failure.

Data & Methods

233,370 hours: MTBF for Catalyst 2960X-48FPD-L
http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-x-series-switches/data_sheet_c78-728232.html

- Assume that this figure covers a stand-alone Cat2960X, for a total port count of (48).

179,714 hours: MTBF for Catalyst WS-C4510R+E:
http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-4500-series-switches/product_data_sheet0900aecd801792b1.html

- Assume that this figure covers a Cat4510R equipped with redundant power supplies and (8) line cards, for a total port count of (384).

I use this posting as a guide.
http://answers.google.com/answers/threadview?id=390140

Calculation

Catalyst 2960X

First Effort

The hard part of this comparison is figuring out what the MTBF is for an (8) Member Cat2960X Stack.

Minimally, I propose the following:
Lamba 1 = 1,000,000 / 233,370 = 4.3
Lamba 2 = 1,000,000 / 233,370 = 4.3
Lamba 3 = 1,000,000 / 233,370 = 4.3
Lamba 4 = 1,000,000 / 233,370 = 4.3
Lamba 5 = 1,000,000 / 233,370 = 4.3
Lamba 6 = 1,000,000 / 233,370 = 4.3
Lamba 7 = 1,000,000 / 233,370 = 4.3
Lamba 8 = 1,000,000 / 233,370 = 4.3

Lambda Stack = 4.3 * 8 = 34.4
MTBF Stack = 1,000,000 / 44,512 = 29,070 hours

Since this calculation ignores Stacking Cables and Stacking Modules, the actual figure would be lower.

More generally, I have (5) eight-Member Stacks[4], so I would, using this calculation, experience a Stack-wide failure as follows:

29,070 hours MTBF / 5 Stacks = 5814 hours

[4] Ignoring for the moment, miscellaneous boxes – a (2) Member Stack in one IDF, and the ~13 stand-alone Catalyst 2960X scattered around the Data Center as mgmt plane switches.

Or roughly 1.5 failures every year.

Now, I've glided over an assumption here – that assumption being that if *any* component fails, then the *entire* Stack goes down. That may be true sometimes: the intermittent Stack Instability I'm experiencing currently (see the TAC SRs referenced in the Appendix) – being a prime example. Or, to take another example, a failing Stacking Module can interfere with traffic for most of the Stack. However, one could argue that in the general case, a single Stack Member fails and does not affect the others.

Second Effort

OK, if we focus on the cases in which a single Member fails without affecting the rest of the Stack, then this becomes simple division.

(45) IDF-based Switches
233,370 hours MTBF / 45 = 5186 hours
Or roughly 1.7 failures every year.

Though again, since this calculation ignores Stacking Cables and Stacking Modules, the actual figure would be lower.

Catalyst 4500

Assuming that the published 179,714 figure covers an entire Cat4500, fully-loaded with line cards, then the arithmetic becomes:

179,174 hours MTBF / 5 IDFs = 35835 hours
Or roughly .26 failures every year. [i.e. one failure per four years]

Sanity Check

Do you buy my math?

And do you buy the assumptions?

- Does the Cat4500 MTBF figure include the presence of dual power supplies? [with the idea that the failure of a single power supply does not count as an event contributing to MTBF]
- Do you know of a way to include the failure rates of Stacking Cables & Modules?

I note that I have also posted this query on a Cisco LinkedIn forum and a Cisco Community Forum. 2017-03-03 –sk

Appendix

Larger View

At the end of the day, I suspect that the dominant factors influencing customers' choice between Stackable & Modular access-layer switches comes down to a mix of acquisition cost and personal preference.

However, as one of the people who has to scramble during a failure, I am interested in *supportability* – how much of my daily (nightly and weekend) time I have to invest into a platform just to keep the lights on. In an effort to inform this discussion with data, I attempted in the pages above an MTBF comparison. Here in the Appendix, I expand on the larger challenges in choosing between Stackable and Modular platforms.

Note that I am narrowly interested in delivering IDF-based switching to non-profit biomedical research institutes, which are typically characterized by dense deployments of wired devices, unpredictable and typically large data flows, wide range of client operating systems. Typical IDFs sport 200 – 400 active Ethernet ports; in our environment, current predictions suggest that these densities will climb toward 500 active ports over the next decade.

Contributors to Service Disruption

Hardware Failure

Hardware failure rates grabs organizational attention, because (a) it is sort of easy to measure, and (b) human cognition grasps it more easily than invisible causes (e.g. software bugs). However, I argue that hardware failure typically plays a role in 10-15% of service disruptions: http://www.skendric.com/problem/incident-analysis/2012-06-30/What-Takes-Us-Down.pdf

So, already my approach here is flawed in that it attempts to tackle such a small aspect of the entire challenge.

Software Defects

Per the link above, I argue that Software defects drive system reliability, so the interesting question is how to compare critical software defects between platforms. This is hard, as I suspect that quality assurance engineers at Cisco know better than I do. For example:

- If one platform records ten times more software defects than another … but sells ten times more units, are the defect rates similar?
- If one platform is favored by companies with deep trouble-shooting resources (who presumably open TAC cases and log bug reports), does this skew Field defect rates?
- Likely other influences

In any case, I don't see anyone publishing "Mean Time To Critical Software Defect" figures.

Centralized vs Distributed Systems

Typically, distributed systems scale better but are more complex and suffer from more failure modes. Using this insight, I would predict that the modular platforms would top out at a lower port count but deliver more reliable service (fewer critical bugs), whereas I would predict that the stackable platforms can deliver more ports per installation but suffer from more software defects, given the need for multiple independent systems to play together.

However, I predict all this from a distance, with limited experience in the product space – perhaps other factors overwhelm this one, in terms of system reliability.

Lots of Parts

I would predict that systems with lots of field-installed parts – Stacking Cables & Modules – suffer a higher failure rate than do factory-built systems (modular). Minimally, the fact that the Stackable approach requires so many more power supplies & fans[5] would shrink MTBF for the Stackable approach.

I would predict that systems which are hand-wired together – Stackables, with their chain of Stacking Cables & Stacking Modules – would suffer from a higher service disruption rate due to mechanical issues: screws not plugged in, cables getting caught on technician's tool belts as they walk by.

Centralized versus Distributed Failure

I propose that Modular platforms (equipped with a single Sup card) tend to fail in an all or nothing way – if the Sup card fries or suffers oddities, then all 384 ports share the same fate (likely down).

Whereas Stackable platforms tend to fail in more granular ways – a single Member fries, knocking out (48) ports, but the remaining ports function fine.

My attempted MTBF calculation above does not address this discrepancy.

Troubleshooting

I propose that troubleshooting Stackable platforms requires substantially more time, and expertise, than Modular platforms.

- Our tools for narrowing the fault domain in a Stack are limited – one ends up doing binary search, which is particular disruptive to end-users
- Distributed systems are complex, and complex systems fail in complex ways.
- By comparison, trouble-shooting Modular systems typically consists of replacing the Sup card: a single step.

For a vivid example of this challenge, see the following TAC cases (all related to the same Catalyst 2960X stack, spanning an 8-month period, on-going as of this writing).

SR681075119
SR681437427
SR681437512
SR681694383

[5] I propose that the most common failure component on network gear are power supplies & fans, with individual port failures coming second, and the "brains" in third place.

Different Brains Experience the World Differently

I personally find the chassis-approach cognitively easy to grasp.

- Centralized forwarding – typically, all frames head to the Supervisor card, then back out again. Myself, I find it easy to visualize where congestion occurs. The Shortest Path First approach used in Stackbles requires more cognition on my part – it varies depending on which port talking with port. And understanding how & where congestion occurs within a Stack is not obvious to me (recall that I live in an environment which specializes in large data flows) -- I have to know which big flows are happening right now, and I don’t have tools for doing this.
- Replacing parts – I personally find replacing a Stackable annoying – all those cables to futz with and drop on the floor, power cords to stuff all the way into receptacles, plus remembering to set the priority of the incoming Member low (bad things happen if the incoming Member becomes Master). I much prefer to replace a Sup card – slide it in, boot, copy the config via USB or TFTP, upgrade the IOS to fit our standard, walk away. However, plenty of folks prefer the Stackable approach – they find cable management easier than I do, and particularly appreciate the automated config & OS loading (I see the benefits of that too!)

I raise these points because they have tangential impact on uptime.

Hardware Installation

The Modular approach requires planning – chassis’ are big and typically require non-trivial support from your shipping & receiving department. They also require muscle on installation – the average network team wants two or even three people to contribute to install a big chassis. By comparison, the Stackable system can be installed more easily by a single technician.