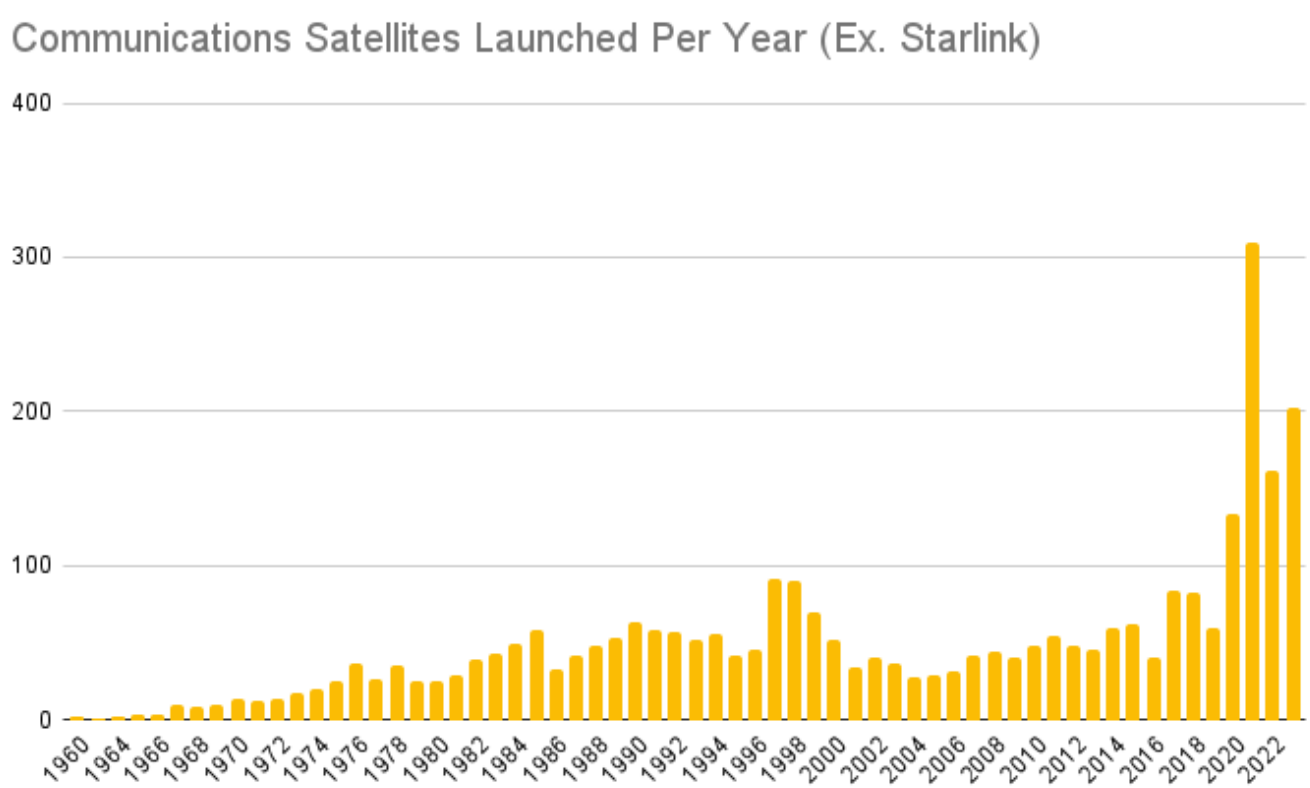
Christopher Kalitin Blog

Fun Charts That Visualize The History of Communicatons Satellites

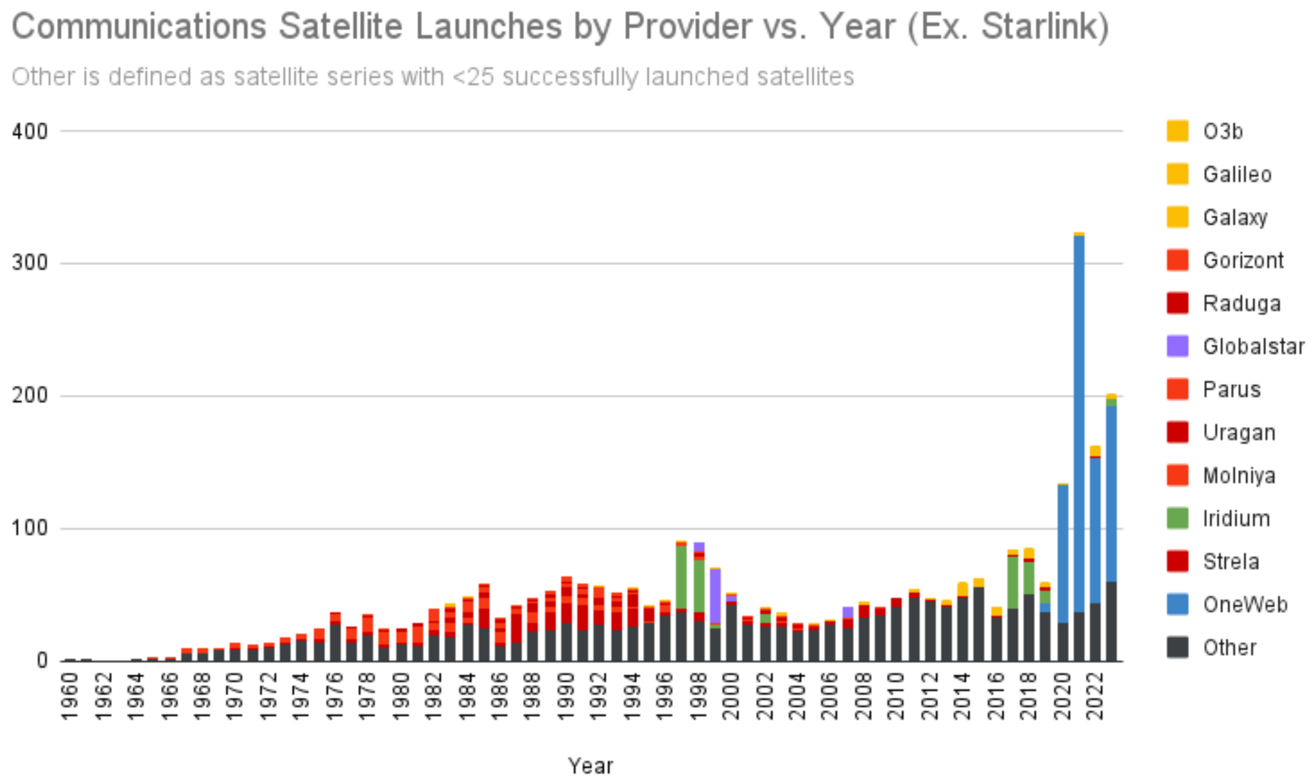
Jan 15, 2025 • Christopher Kalitin



This is the chart from the previous blog post.

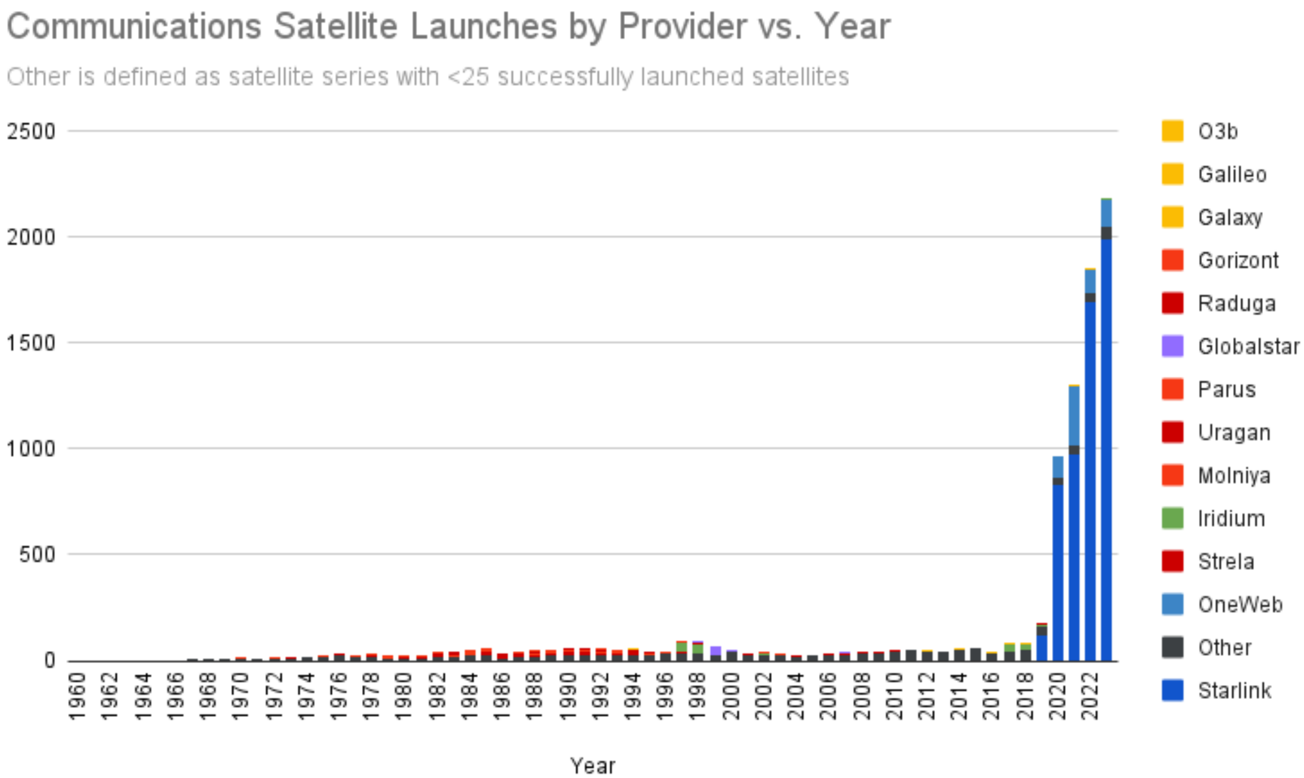
In my blog post from 6 days ago I wrote about the insights I gained along the way to chart out communication satellite bandwidth cost over time. As part of that blog post I created the chart you see above of communication satellites launched over time and described some of the the bumps you see on it. However, without any colour coding or categorization, it's difficult to parse. So, I suffering in Google Sheets for a few hours and categorized the data by Satellite Series so you could better visualize my points.

So, here's 4 fun charts that visualize the history of communications satellites with an explaination under each one of interesting points.

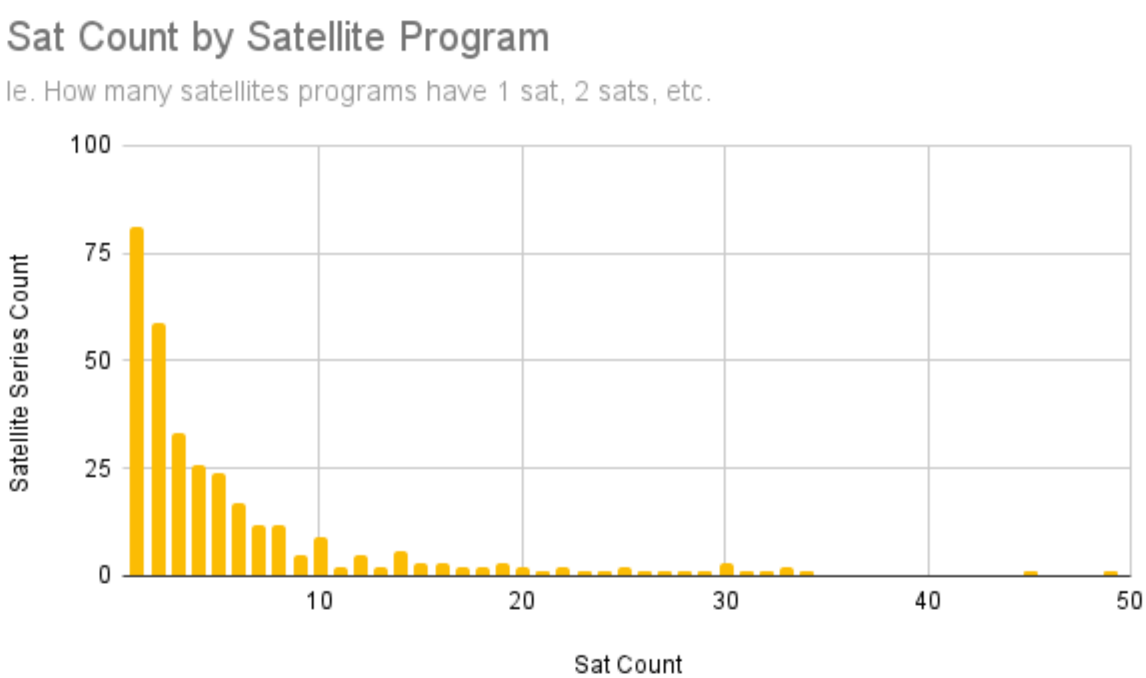


This is the same chart as the one above but with colour coding for each satellite series. Just appreciate for a minute how information dense it is now!

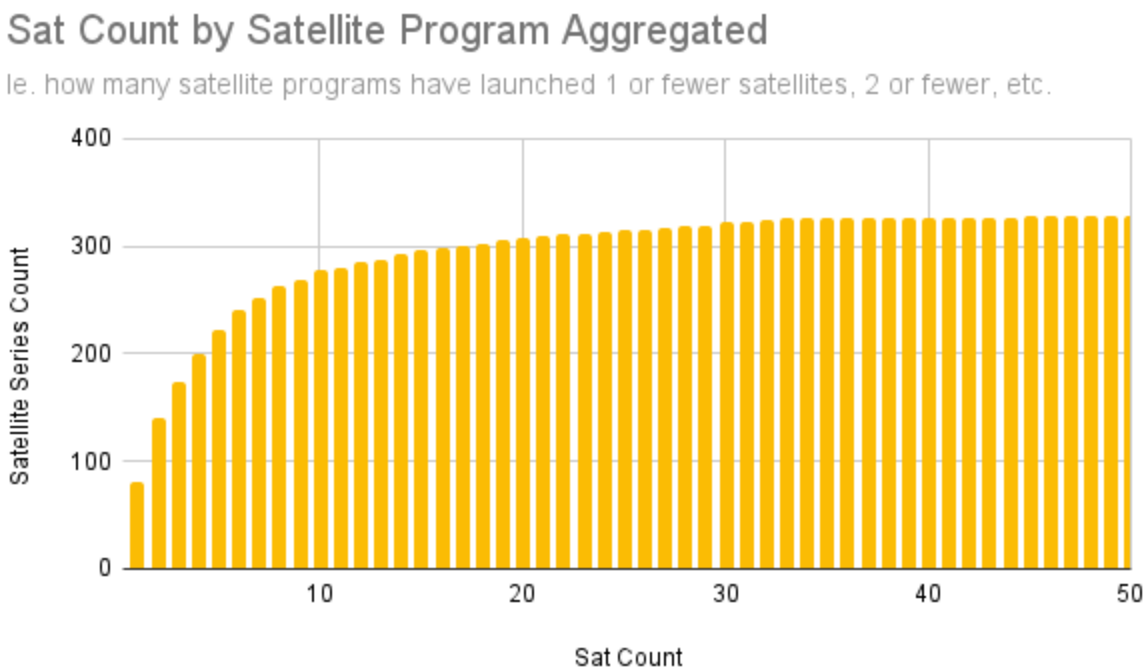
By removing Starlink, we can better see the history of Communications Satellite launches. You can clearly see the Iridium V1 bump from 1997-1998, and directly after it the Globalstar bump, which were Iridium's chief competition back in the day. This also illustrates the massive launch requirements for internet constellations vs. call constellations with the size of the Iridium and Globalstar bumps relative to OneWeb (and of course Starlink, but that's on a whole other level). All the red on the chart is all the other Soviet/Russia satellites and the yellow is western communications satellites and Galileo, which is the European GPS system.



This chart is the same as the one above but includes Starlink, notice how you can learn pretty much nothing except that yea Starlink is still winning and Elon is the most important living human.



Just for fun, you can see the power law distribution of satellite programs by the number of satellites launched. As you’d expect, most satellites are one-offs or part of a small series, you get what'd you'd expect after that, an exponential decay.



This chart is another way to illustrate the power law distribution of satellite programs as you can see how many satellite programs have launched 1 or fewer satellites, <=2, <=3, etc. The previous chart is the derivative of this function.

Subscribe

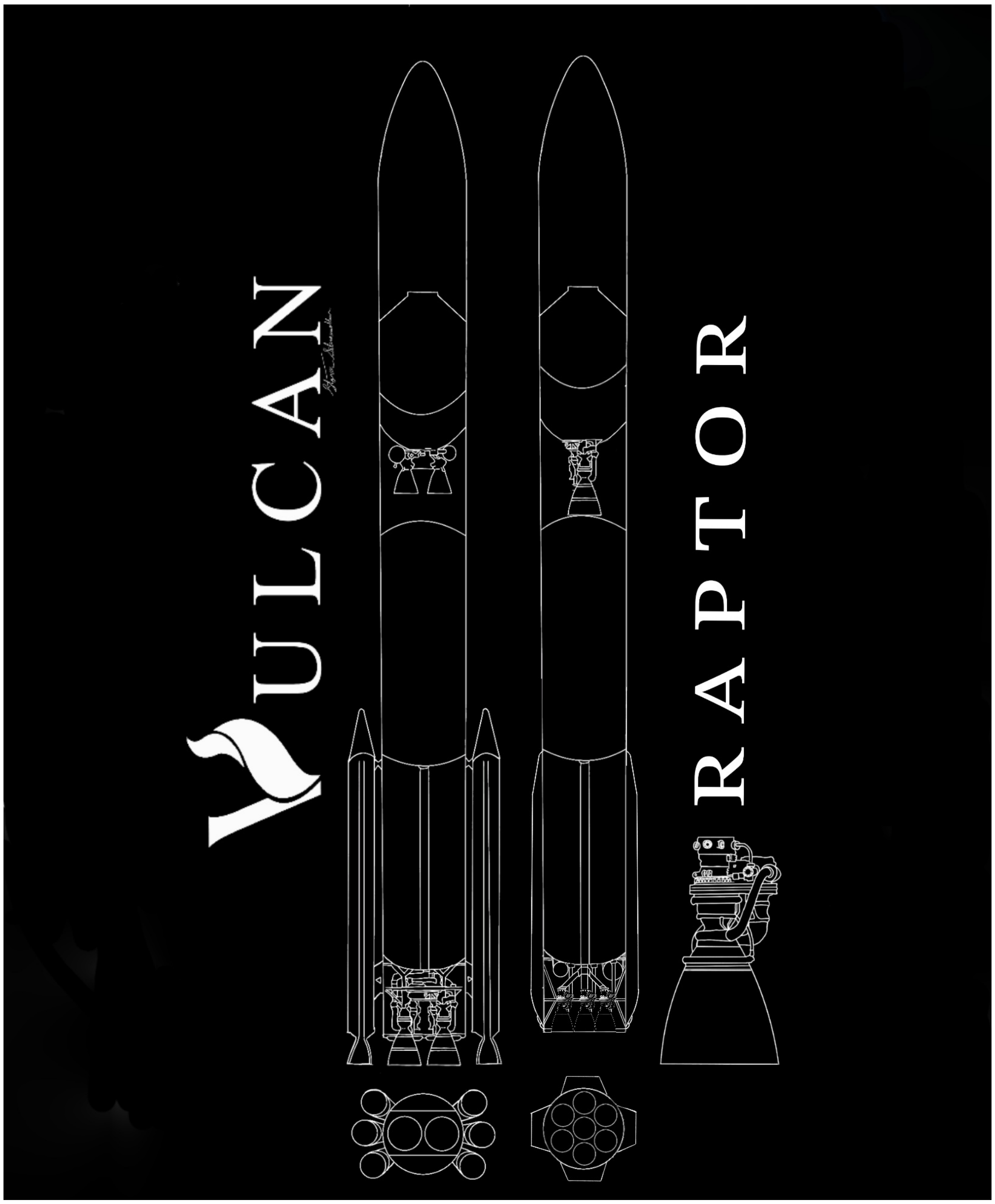
CKalitin
x.com/CKalitin

Geohot made a blog too. You should be working on hardware

Christopher Kalitin Blog

A New Paradigm of Launch Vehicle Architecture Design Tradeoffs

Jan 12, 2025 • Christopher Kalitin



This image I got from @StormSilvawalk1's reply illustrates the two paradigms well. 10 separable vehicle parts (SRBs, stages, fairing etc.) vs. just 2 for Starship & Neutron.

We are currently living through a paradigm shift in launch vehicle architecture design. We are seeing old space companies like ULA or Ariane Space that are used to building highly customizable vehicles try to adapt to a new paradigm of reuse and high flight rates. Every day we continue to see these legacy perspectives bump up against new ideas that are optimized for different requirements. These companies once had extremely successful business models, but now they are struggling to adapt to the new paradigm of launch vehicle design, sometimes resulting in laughable concepts.

We're Seeing a Paradigm Shift From Customizable Vehicles to Vehicles Focused on LEO

The old paradigm of vehicle design was optimized for infrequent launches to a variety of orbits. The Evolved Expendable Launch Vehicle (EELV) program vehicles are great examples of this. The Delta IV and Atlas V were both optimized to have a wide range of configurations to meet the various target orbit and mass specifications that the US government laid out. This led to 11 flown configurations (according to Wikipedia) for the Atlas V and 5 flown configurations for the Delta IV. These configurations varied the number of boosters and size of the fairings and were the proper tradeoff for the requirements that Boeing and Lockheed Martin were given at the time.

The new paradigm ushered in by SpaceX's Falcon 9 is focused on reuse, high flight rates, and standardizing your vehicle for a specific orbit. Falcon 9 was originally optimized only to deliver the Dragon capsule to the ISS which is in Low Earth Orbit (LEO). Now, its primary use is launching the Starlink satellites to LEO. To a first approximation, Falcon 9's architecture is solely focused on getting as much mass to LEO as possible. Starship shows this goal even more clearly. It is a vehicle architecture that is completely focused on getting as much mass to LEO as possible. This means that through reuse and an extremely high flight rate they can achieve low cost, and then leverage refilling missions to get to Mars or other high-energy destinations in the solar system.

The Paradigms Decide Vehicle Architecture Optimizations

Both the old paradigm of vehicle customizability and the new paradigm of achieving low cost through reuse and high cadence are sound given their respective requirements (reaching EELV target orbits vs. getting tons of mass into LEO). However, issues start to arise when the companies that were dominant in the old paradigm try to adapt to the new paradigm.

Legacy space companies like ULA or Ariane Space are used to building customizable vehicles (varying booster counts or fairing sizes) because this was the most economical way to launch all the payloads they were tasked with launching. It is unfortunate to waste a full vehicle rated for 20t when your payload is only 10t, so instead you rate your base vehicle for 10t and add boosters onto it to be able to launch 20t. Atlas V took this to an extreme with some of its launches using only one booster (imagine that offset thrust)!

This approach has no place in the paradigm shift to reuse. Because the Falcon 9 is capable of booster reuse, SpaceX is not particularly impacted by whether their payload is 2t or 20t, they get the booster back either way so they have no reason to spend all the money required to create 10 different variants for different payload classes.

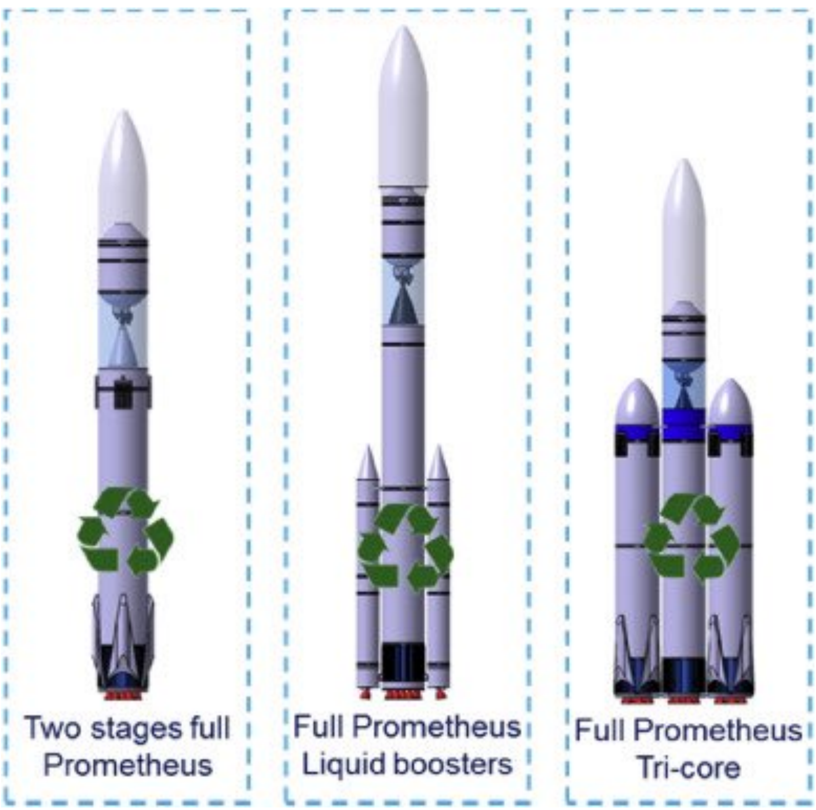
A defining characteristic of this paradigm shift is the approaches to lowering the cost of missions. The legacy model was to use as little hardware as possible for a given mission to save money (hence the 11 Atlas V configurations). The new model is to use a standard vehicle for all your missions to save money. This is because the primary cost savings of reuse come from high cadence (this was part of the reason the Shuttle was never cheap). Flying a standardized vehicle for all your missions has immense value in the cost savings the high cadence provides.

With the partially reusable Falcon 9, there are still some areas where SpaceX doesn't get cost savings from high cadence and they revert to the old model of using as little hardware as possible to decrease cost. For example, on Transporter missions, they use a shorter Merlin engine nozzle extension to save the cost of the Niobium.

Looking into the future, SpaceX is fully bought into reuse and the result is they are developing a single vehicle that they can throw at every problem. With Starship they're again going all in on a single vehicle and throwing it at every problem.

Everyone on X already understands all my points intrinsically through memes. A meme is worth a million words, and it spreads as a mind virus.

Not Having Full Buy-in To The New Paradigm Explains All The Foolishness We're Seeing



The Ariane NEXT concept illustrates why Ariane Space is not adapting to the new paradigm well.

There is this hilarious, infamous clip of an Ariane Space exec saying SpaceX's plans are a dream and if they ever come true that Ariane Space will be quick to follow. This is a phenomenon of believing the current paradigm of launch vehicle design will continue long into the future and a belief that Ariane Space would be fast enough to keep up with any paradigm shifts. Both of these points turned out to be false, hence why this clip is hilarious.

One of Ariane Space's concepts for a reusable vehicle is Ariane NEXT. The end goal of Ariane NEXT is a partially reusable 3-core vehicle, similar to Falcon Heavy. One of the interim steps described in a paper on the concept (Section 4.2) is to have smaller liquid boosters based on the same engine to increase the payload capability of the vehicle. There is also some discussion of a hydrolox upper stage, while the standard vehicle is methalox.

The Ariane NEXT concepts are an example of Ariane Space still thinking in the old paradigm of having a highly customizable vehicle. Instead of completely buying into having just a single vehicle (or at most 2 versions like Falcon 9 and Falcon Heavy), they want to have 3+ versions (single core, small boosters, large booster, plus a potential hydrolox upper stage). This is an approach that will not get them the cost savings of having a single standard vehicle (like the Falcon 9 booster).

The architecture of the Vulcan rocket shows that ULA is another old-space company that isn't adapting to the new paradigm well. Vulcan is a customizable vehicle in that it can have 0, 2, 4, or 6 solid boosters. Almost as an afterthought, they included some reuse in the design with their SAFER concept, that decouples, reenters (with an inflatable heat shield), and parachutes the engines back to Earth. This shows even less buy-in to the new paradigm, as they are seemingly trying to reuse as little of the vehicle as possible while keeping the ability to add solid boosters to increase payload capability.

The new paradigm of reuse incentivizes a single vehicle that can be reused as much as possible for all missions. ULA's Vulcan architecture is not optimized for this. Like I said in the first section, these design decisions make complete sense if you're operating in the old paradigm, but they are not the right tradeoffs for the new paradigm. Vulcan is optimized to launch US Military payloads through the NSSL program, but not to launch as much mass to LEO as possible which is the requirement for upcoming megaconstellations (See the Constellations - The Next Paradigm section). When launch is no longer extremely supply-constrained, it will become clear that Vulcan is not optimized for competing with vehicles optimized for economical reuse.

Conclusion

The fundamental issue with Ariane Space's and ULA's approaches is that the new paradigm of reuse and high flight rates incentivizes integrating as much onto a single reusable booster as possible. Don't have solid boosters, just make your core bigger and get good at reuse. Don't even have separable fairings! Integrate those onto a reusable stage so reusing them is even easier!

ULA and Ariane Space have clearly not committed to this new approach and are hanging on to the old paradigm of having a highly customizable vehicle. This approach may work for them in the short term as they have government support, but in the long term, it will become clearer and clearer that this approach will not be commercially competitive with other vehicles optimized for reuse.

Updates

[Update Jan 12 2025]
Casey Handmer explains the issues with increasing the number of vehicles your architecture uses in this blog post. It's a good read to get more context on this issue and described the issues with increasing the complexity of you architecture well.

Another way to consider the paradigm shift I described is architecture complexity vs. launch cost (ie. do you make more vehicle configurations or just make a single vehicle that is oversized for most tasks that'll be cheaper through reuse).

Subscribe

CKalitin
x.com/CKalitin

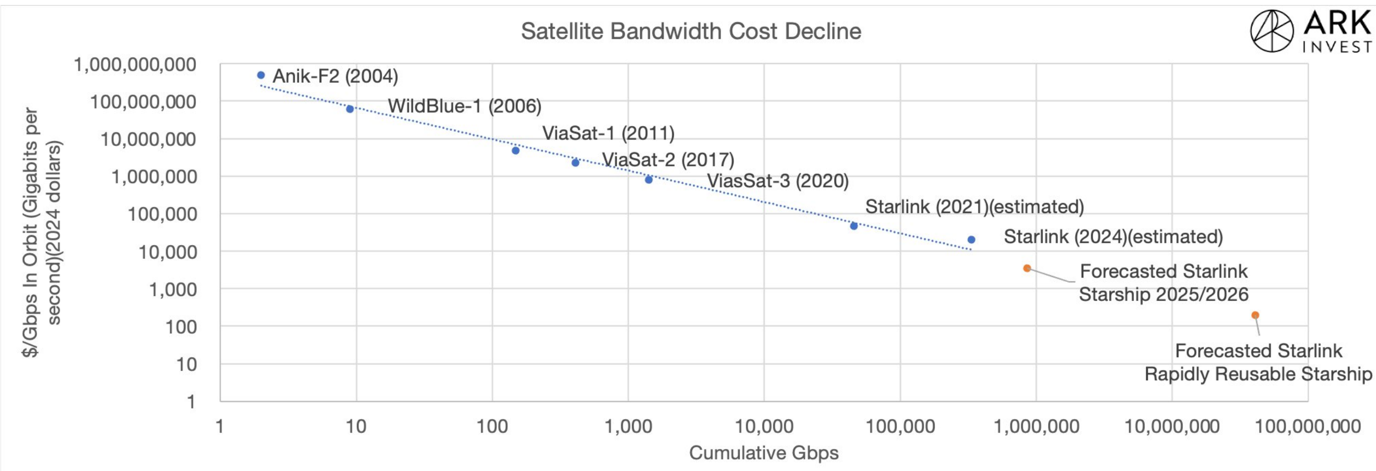
Geohot made a blog too. You should be working on hardware

Christopher Kalitin Blog

Estimating Cost per Mbps of Communication Satellites Over Time

Jan 9, 2025 • Christopher Kalitin

Find my raw data here.

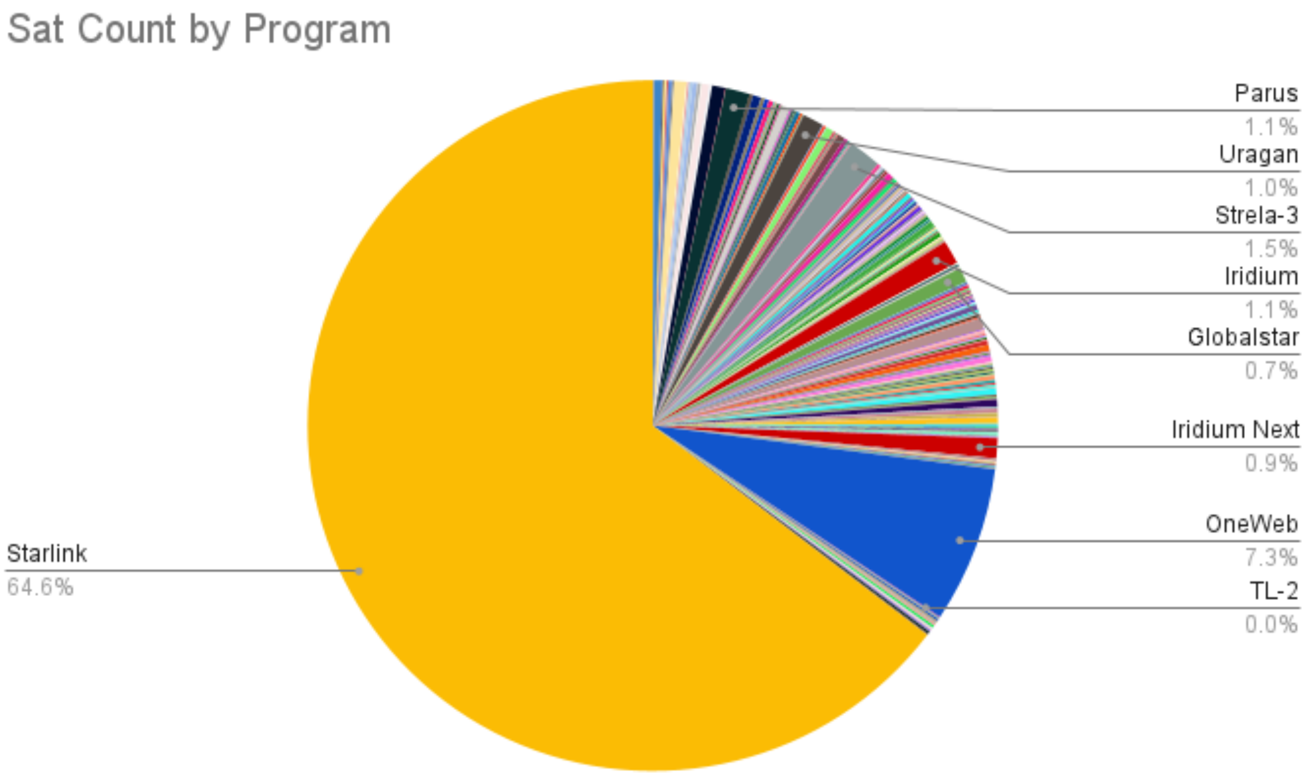


This is the Ark Invest chart that inspired this blog post.

Three hours ago I saw (Space) Case Taylor reply to a post Sam Korus of Ark Invest made in which he showed a chart of satellite bandwidth cost over time. You can see this chart above and it clearly shows the exponential cost decline of satellite bandwidth over time, the beauty of learning rate in production. As an exercise and an opportunity to learn more about the communication satellite industry, I decided to derive a similar chart myself.

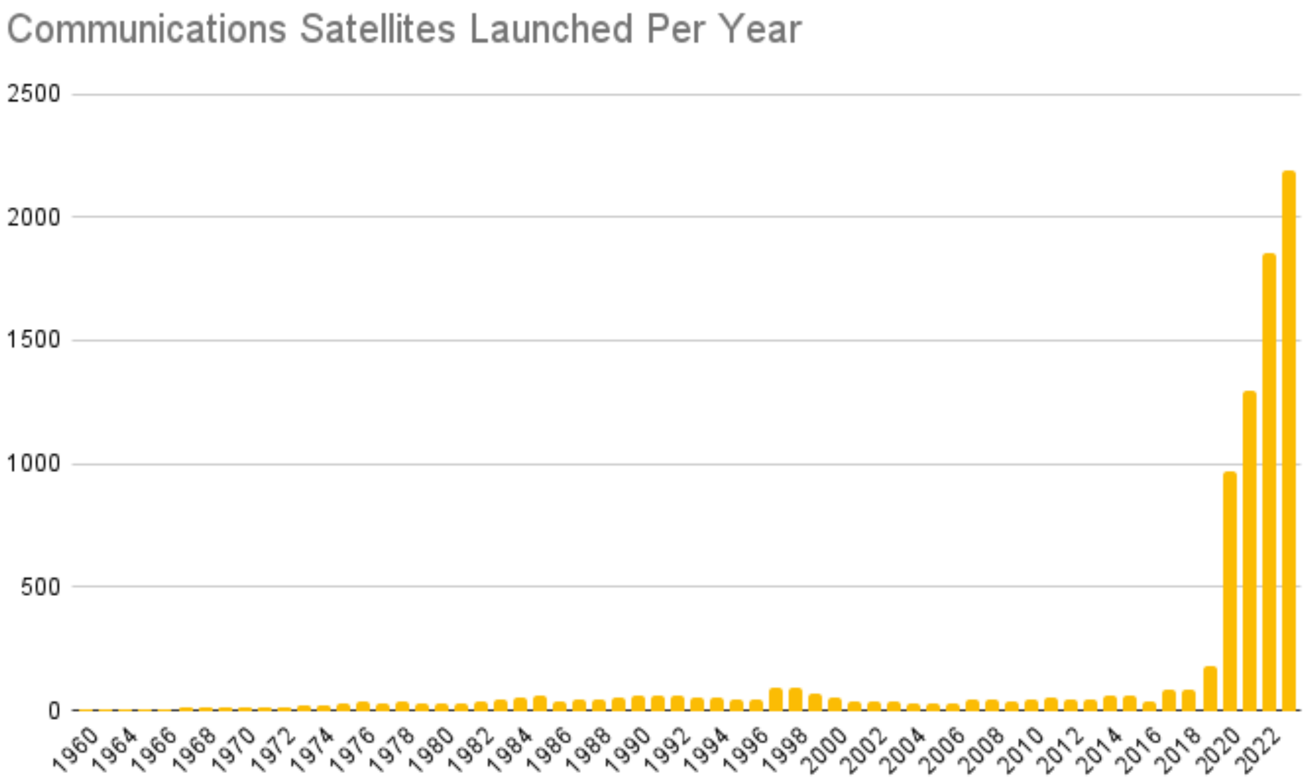
Read the sequel to this blog post to get better charts the describe the history of Communications Satellites.

Using Jonathan Mcdowell's Dataset



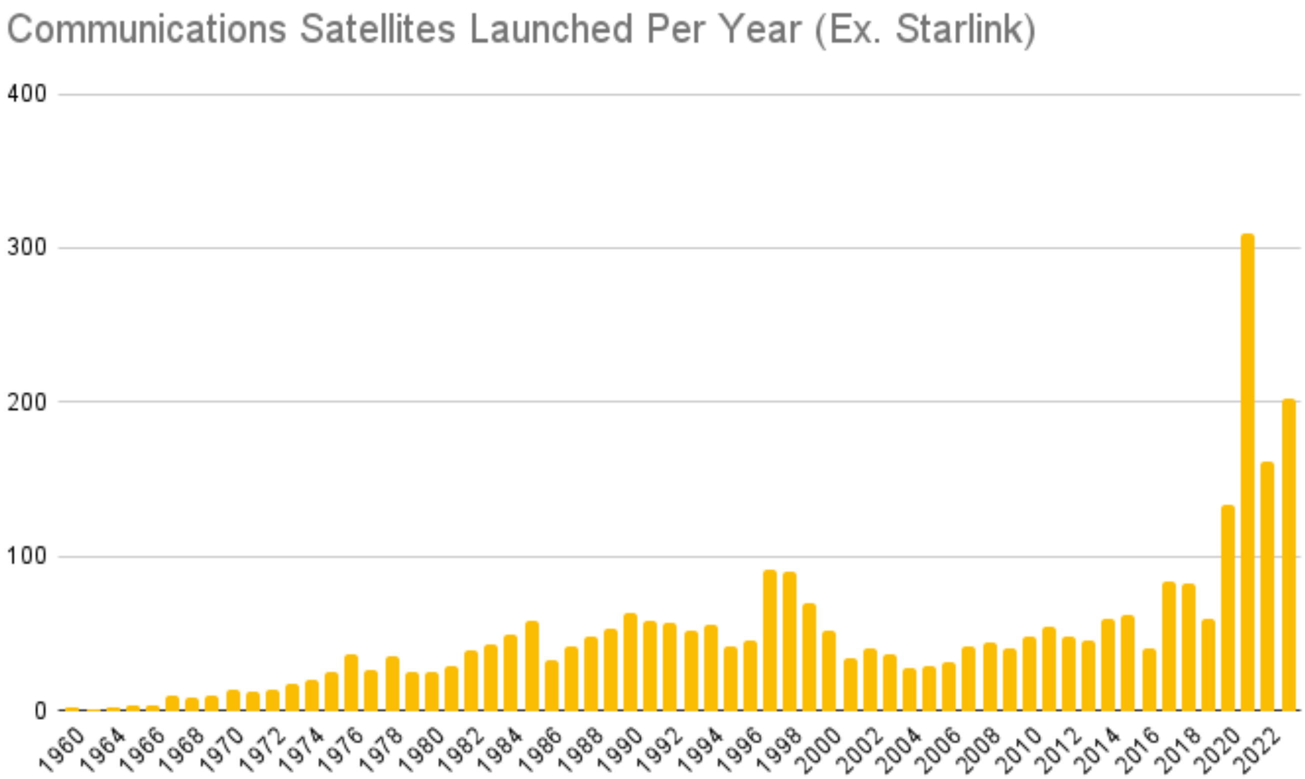
This shows share of total communications satellites launched in all of history by satellite program.

The first step was to filter Jonathan Mcdowell's dataset to include only communication satellites. I've worked with his dataset for over a year and already did this a while ago. This resulted in a list of 8,691 communications satellites successfully launched between 1960 and 2023 (inclusive), the first of these being Transit 1B on April 13, 1960. This data isn't directly pertinent to the result I'd like to get, but the journey is important and there are insights to be gained from this, these insights are the purpose of this section.



That massive wall post 2020 is Starlink.

As is a common theme in trying to analyze the satellite industry, SpaceX's massive market share makes it very difficult to extract any insights from the raw charts, and if you want to learn anything other than "yea SpaceX is still winning" you have to exclude them from the dataset.



Without Starlink, you can get a feel for the history of communications satellites.

Above you can see communications satellites launched per year excluding Starlink and there are a lot of interesting stories behind all the bumps and increases.

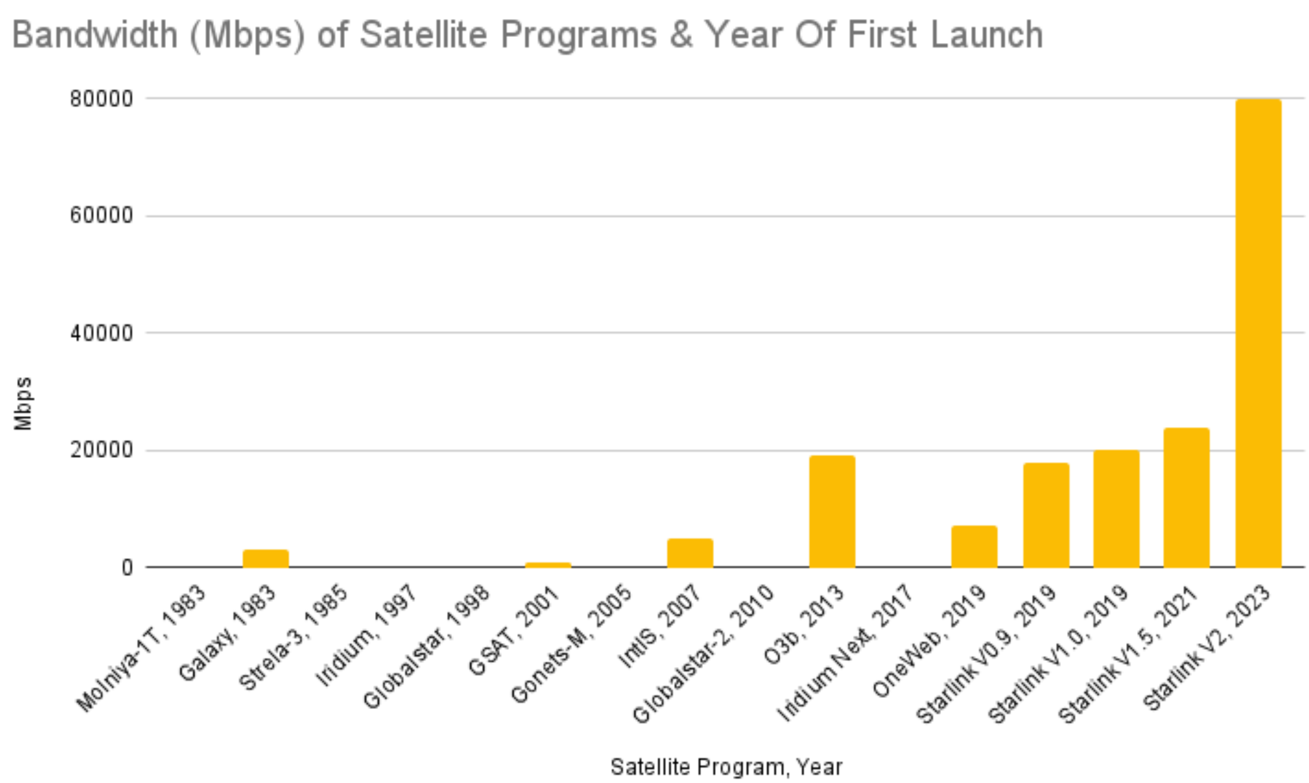
The 1997-1999 bump is mainly due to the initial Iridium satellites. The majority of these satellites were launched by the Delta II out of SLC-2A, Iridium practically owned all launches out of this pad for a couple years. They had 40% of all launch contracts in the world during this time. I've been reading Eccentric Orbits and there are a lot of fun stories about Iridium's early days. Another one is that in the lead up to the initial launches, the GPS IIR-1 launch failed and the Delta II was grounded for a few months, you've probably seen the video of this and Scott Manley has a great video on it, as it's one of the most spectacular launch failures in history. It's also the origin of the "we have had an anomaly" meme circulating on X.

The smaller 2017-2018 bump is the Iridium NEXT launches on Falcon 9. Post 2020, the large increase is due to all the OneWeb launches on Soyuz, Falcon, and ISRO's LVM3.

Getting Price & Bandwidth

With the data on every communications satellite ever successfully launched, I narrowed the dataset down to only programs with more than 20 satellites launched of a single series. This narrowed it down to 33 series of satellites. Some of these could be combined (eg. Jonathan Mcdowell lists many subcategories of Starlink satellites when I only needed V0.9, V1, V1.5, and V2 mini) or ignored (it's difficult to get pricing data on Soviet Molniya satellites launched in the 70s). I was left with a set of 16 satellite programs/series to analyze. With the help of Grok, ChatGPT, and the internet I got price and bandwidth estimates for each of the satellites including manufacturing and launch. Many of the early satellites didn't have good data and the LLMs hallucinated often, but the prices generally seemed to be in the right order of magnitude, which is sufficient for this analysis. Later satellites like Iridium, OneWeb, and Starlink had good data available, so the important data points would be reasonably accurate.

Here I'll note that this is not the approach Ark Invest took, they used a set of GEO & Starlink satellites with good data on them and fit a trendline to the price per Mbps. For my next attempt this is probably what I'll do, start with satellites that have good price data instead of satellites with many launches.



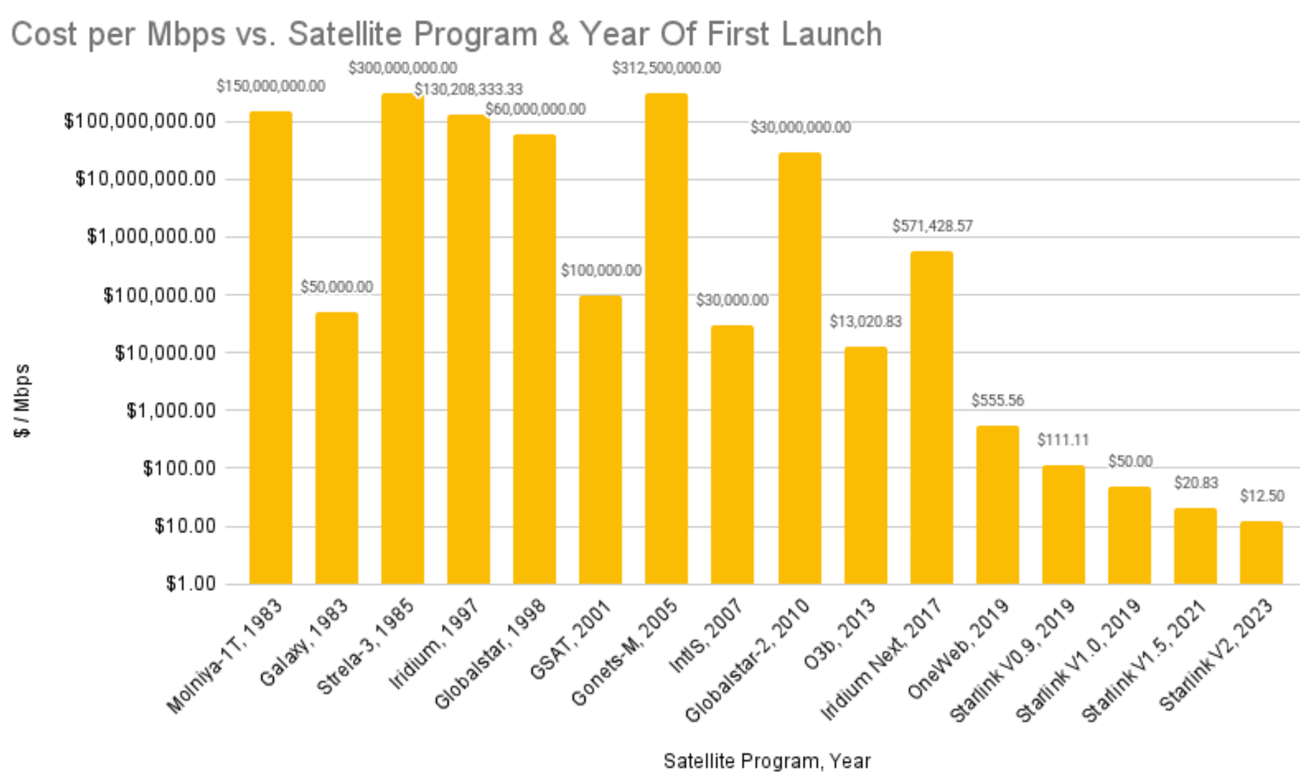
You can see that bandwidth per satellite has been increasing exponentially over time.

Above you can see the bandwidth of single satellites from various programs over time. There are four primary categories here, early communications satellites (Molniya), early LEO satellite phone constellations (Iridium, Globalstar), GEO satellites (O3b, Intelsat (IntlS on the chart), SES Galaxy), and LEO internet constellations (OneWeb, Starlink).

The early satellites like Molniya and Iridium supported such low data rates that they're not even visible with Starlink on the chart. Each Iridium call was on the order of a few kilobits per second for audio transmission, and the Molniya satellites were even worse. In contrast, internet requires megabits per second, a minimum of 6 orders of magnitude more data.

The GEO satellites also had high total throughput compared to the early LEO constellations. These satellites provided TV, data, and phone service to entire continents, while the early LEO satellites served a few thousand customers at a time. This meant total data throughput to customers could be much higher for single satellites, especially given that many people could tune into a single TV channel at once, meaning a single signal from a satellite could be received by far more people than would be involved in a phone call. This is part of the reason I didn't include any GPS/GNSS satellites in this analysis, it's not clear what bandwidth means in the context of a GPS satellite where you're just broadcasting a constant signal to be received by a huge number of receivers.

Results



We see an exponential decline in cost per Mbps over time on this logarithmic chart.

In the previous section, I described the 4 major categories of satellites that I included in the dataset. Because these satellites have different purposes, we see ~3-4 orders of magnitude difference in the 1983-2017 set of satellites. This is because the GEO satellites can deliver far more bandwidth to customers than the LEO satellites like Iridium, so the cost per Mbps differs a lot. This is similar to the issues you encounter when using $/kg as a metric in launch vehicles, Electron is several times higher than F9 but still has customers because its customers have different needs.

Again, this is a reason to do the analysis in the way Ark Invest did instead of the way I did. Maybe Sam Korus of Ark also ran into similar issues as I did in doing his research and he course corrected before publishing a graph. I'm not profesional enough to not show my work, so this blog post is full of interesting intermediary results.

Post Iridium Next / OneWeb in 2017 and 2019, we see an exponential decline in the cost of bandwidth with LEO broadband satellites. Remember that these numbers are mainly illustrative of the shape of the graph since SpaceX doesn't give us precise internal launch cost and satellite manufacturing numbers, but the trend is clear. The cost of bandwidth from space has been exponentially declining and is likely to continue as satellite technology continues to improve, the scale of the satellites increases with Starlin V3, and competition enters the arena.

Christopher Kalitin Blog

Using SCPI and PyVISA to Automatically Collect Data for ADC Characterization

Dec 29, 2024 • Christopher Kalitin

This blog post is a copy pasted Monday Update I wrote for UBC Solar. So here are some defitions/explanations of terms & names:

People:

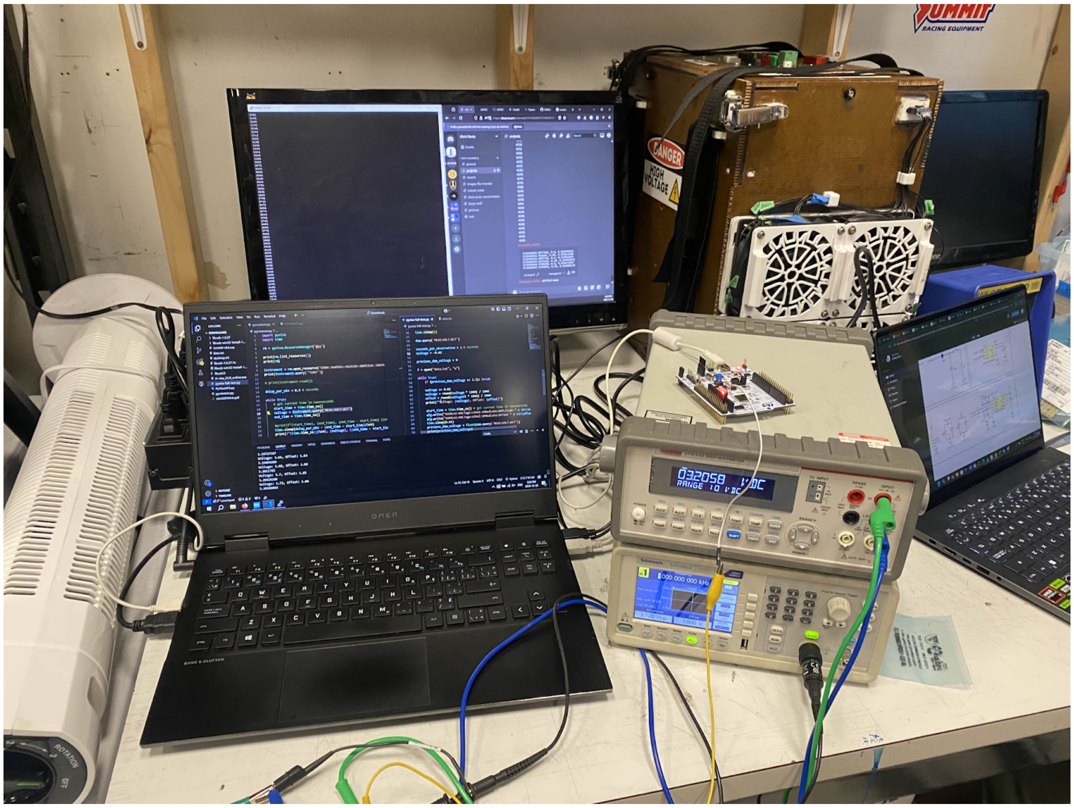
1. Saman - UBC Solar Electrical Lead
2. Mischa - Previous UBC Solar Electrical Lead and Previous BMS Lead
3. Krish - Current UBC Solar BMS Lead

Terms:

1. Monday - Our project management software
2. SCPI - Standard Commands for Programmable Instruments
3. AFG - Arbitrary Function Generator
4. DMM - Digital Multimeter
5. Nucleo - STM32 Nucleo Micro Controller
6. HPPC - Hybrid Pulse Power Characterization (Battery test)

Context: This was part of a UBC Solar project to characterize the current sensor in our battery pack.

Github Project Link containing SCPI Python scripts & STM32 Project: here
Raw Data and Chart: here



My final testing setup

This morning Saman suggested I find a way to use our Digital Multimeter (DMM) and other tools to automatically collect data for ADC characterization. In my previous Monday update I described how we can feed the output of the DAC into the ADC pin to automatically characterize an ADC over a range of voltage values. This plan falls through because I found that the DAC can't be trusted to output a specific voltage value and needs to be characterized itself. The solution is to use external trustworthy equipment to complete the automatic characterization.

I started by figuring out how to program the DMM to automatically record and print values. SCPI (Standard Commands for Programmable Instruments) is the standard communication protocol to do this and there is a python library called PyVisa that allows for simple scripting. This took hours to set up and was most of my day, but it's simple if you know what you're doing, which now I do. If we plan to do similar tests in the future it would be useful for me to write a wiki page on how to set it up (automatic data collection like this was suggested for HPPC testing).

My goal was to automatically generate a voltage and read it with both our DMM and a Nucleo. This way we would assume the DMM is trustworthy and could characterize the error in the Nucleo's ADC.

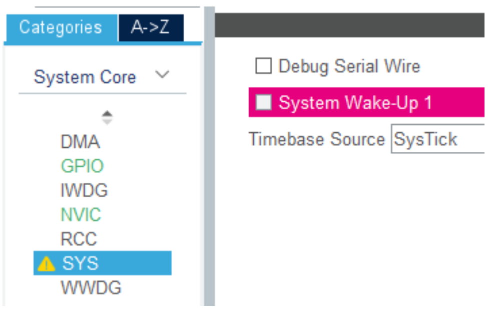
Our Power Supply doesn't support SCPI and can't be programmed, but our Arbitrary Function Generator (AFG) does support SCPI.

I wrote a script that made the AFG output a constant voltage that was read by the DMM. I ran this in a loop from 0 to 3.3V with a 0.02V step size. This 0.02V step size was use for the final of 5 tests

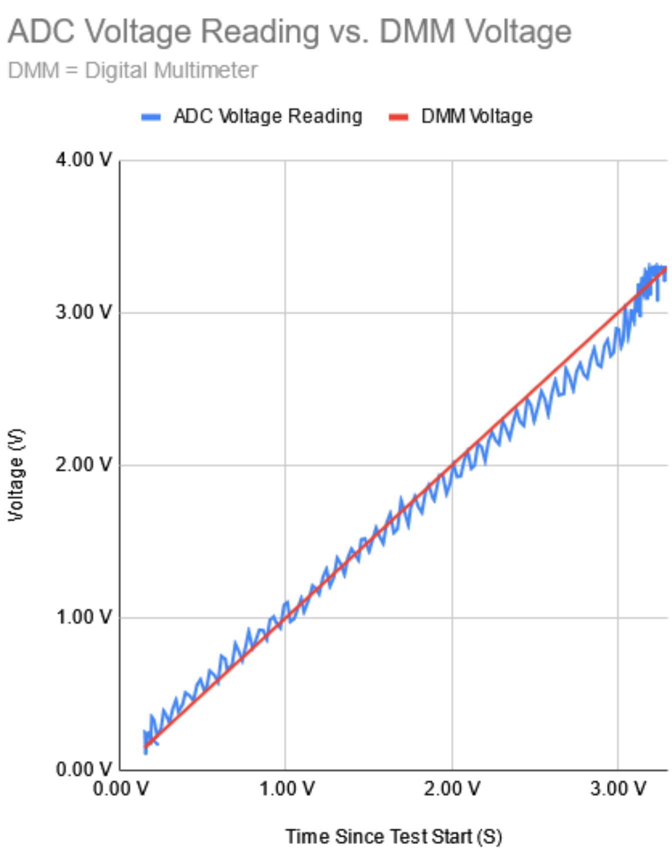
I did with this script and is what generated all the charts in this update.

Note: the AFG doesn't have a settings for a constant voltage, so I made it a pulse with a 99.9% duty cycle which appears to be close enough to a constant voltage for our purposes.

A step is done once per second, so a full test took ~4-5 minutes. This long step length is used so that the Nucleo, AFG, and DMM will be roughly synchronized without any communication between them. To illustrate, It's easier to click start on both in a 1 second window than a 0.01 second window.



Side note: When Krish and I were testing the ECU a few weeks ago we lost the ability to upload code to the STM32 chip, we struggled for 1.5 weeks and Mischa solved our problem in 5 minutes by showing us the Debug Serial Wire toggle in STM32 Cube IDE. This came in handy when I was testing with the Nucleo today, without it my test would've been stopped in its tracks. Thx Mischa.



This is the graph that can be used to characterize the Nucleo's ADC