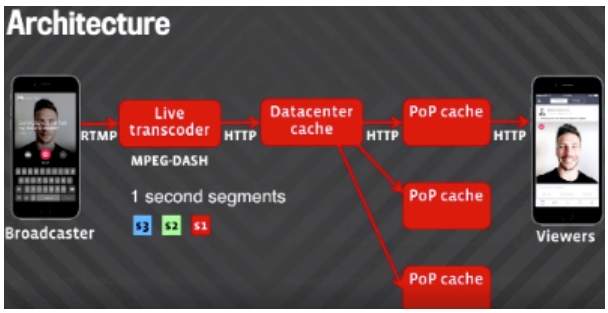
EXAMPLE

How Facebook Live Streams to 800,000 Simultaneous Viewers

High Scalability
Jun 27, 2016 — 7 min read



Fewer companies know how to build world spanning distributed services than there are countries with nuclear weapons. Facebook is one of those companies and Facebook Live, Facebook's new live video streaming product, is one one of those services.

Facebook CEO Mark Zuckerberg:

> The big decision we made was to shift a lot of our video efforts to focus on Live, because it is this emerging new format; not the kind of videos that have been online for the past five or ten years...We're entering this new golden age of video. I wouldn't be

> surprised if you fast-forward five years and most of the content that people see on Facebook and are sharing on a day-to-day basis is video.

If you are in the advertising business what could better than a supply of advertising ready content that is never ending, always expanding, and freely generated? It's the same economics Google exploited when it started slapping ads on an exponentially growing web.

An example of Facebook's streaming prowess is a 45 minute video of two people exploding a watermelon with rubber bands. It reached a peak of over 800,000 simultaneous viewers who also racked up over 300,000 comments. That's the kind of viral scale you can generate with a social network of 1.5 billion users.

As a comparison The 2015 Super Bowl was watched by 114 million viewers with an average 2.36 million on the live stream. On Twitch there was a peak of 840,000 viewers at E3 2015. The September 16th Republican debate peaked at 921,000 simultaneous live streams.

So Facebook is right up there with the state of the art. Keep in mind Facebook would have a large number of other streams going on at the same time as well.

A Wired article quotes Chris Cox, Facebook's chief product officer, who said Facebook:

Has more than a **hundred people** working on Live. (it started with ~12 and now there are more than 150 engineers on the project)

Needs to be able to serve up **millions of simultaneous streams** without crashing.

Need to be able to support **millions of simultaneous viewers on a stream**, as well as seamless streams across different devices and service providers around the world.

Cox said that "It turns out it's a really hard infrastructure problem."

Wouldn't it be interesting if we had some details about how that infrastructure problem was solved? Woe is we. But wait, we do!

Federico Larumbe from Facebook's Traffic Team, which works on the caching software powering Facebook's CDN and the Global Load Balancing system, gave an excellent talk: Scaling Facebook Live, where he shares some details about how Live works.

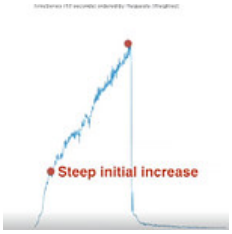
Here's my gloss on the talk. It's impressive.

Origin Story

- Facebook is a new feature that allows people to share video in real-time. (Note how this for Facebook is just another feature).
- Launched in April 2015 Live could only be used by celebrities through the Mentions app as a medium for interacting with fans.
- This began a year of product improvement and protocol iteration.
  - They started with HLS, HTTP Live Streaming. It's supported by the iPhone and allowed them to use their existing CDN architecture.
  - Simultaneously began investigating RTMP (Real-Time Messaging Protocol), a TCP based protocol. There's a stream of video and a stream of audio that is sent from the phone to the Live Stream servers.
    - Advantage: RTMP has lower end-end latency between the broadcaster and viewers. This really makes a difference an interactive broadcast where people are interacting with each other. Then lowering latency and having a few seconds less delay makes all the difference in the experience.
    - Disadvantage: requires a whole now architecture because it's not HTTP based. A new RTMP proxy need to be developed to make it scale.
  - Also investigating MPEG-DASH (Dynamic Adaptive Streaming over HTTP).
    - Advantage: compared to HLS it is 15% more space efficient.
    - Advantage: it allows adaptive bit rates. The encoding quality can be varied based on the network throughput.
  - Pied Piper Middle-Out Compression Solution: (just kidding)

Live Video is Different and that Causes Problems

- The traffic pattern of the Watermelon video mentioned earlier:
  - A very steep initial rise, in a few minutes it reached more than 100 requests per second and continued increasing until the end of the video.
  - Then traffic dropped like a rock.
  - In other words: traffic is spiky.



- Live video is different than normal videos: it causes **spiky traffic patterns**.
  - Live videos are more engaging so tend to get watched **3x more** than normal videos.
  - Live videos appear at the top of the news feed so have a higher probability of being watched.
  - Notifications are sent to all the fans of each page so that's another group of people who might watch the video.
- Spiky traffic cause problems in the caching system and the load balancing system.

  - A lot of people may want to watch a live video at the same time. This is your classic Thundering Herd problem.
  - The spiky traffic pattern puts pressure on the caching system.
  - Video is segmented into one second files. Servers that cache these segments may overload when traffic spikes.
- **Global Load Balancing Problem**
  - Facebook has points of presence (PoPs) distributed around the world. Facebook traffic is globally distributed.
  - The challenge is preventing a spike from overloading a PoP.

Big Picture Architecture

This is how a live stream goes from one broadcaster to millions of viewers.

A broadcaster starts a live video on their phone.

The phone sends a RTMP stream to a Live Stream server.

The Live Stream server decodes the video and transcodes to multiple bit rates.

For each bit rate a set of one-second MPEG-DASH segments is continuously produced.

Segments are stored in a datacenter cache.

From the datacenter cache segments are sent to caches located in the points of presence (a PoP cache).

On the view side the viewer receives a Live Story.

The player on their device starts fetching segments from a PoP cache at a rate of one per second.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.