PacketZoom
Technology Overview

Table of Contents

Executive Summary

This white paper provides a comprehensive overview of the PacketZoom Mobile Expresslane platform. We cover platform architecture, functionality, configuration, and how to deploy the service. Our target audiences are DevOps, engineers, and non-technical managers who would like to learn more about Mobile Application Performance Management (Mobile APM) and Optimization.

Industry Background

Some thirty years ago, as the Internet became more popular, CDNs (Content Delivery Networks) were necessary to allow websites to scale and deliver content faster, and in a more reliable way. The entire web performance ecosystem, including web performance measurement tools, were designed to optimize the desktop web experience and they delivered good results overall. The wired web infrastructure, including client technologies, have matured greatly across that time and most of the developed world population now has relatively fast and reliable Internet service.

In 2007, mobile Internet usage was significantly affected by the introduction of the first iPhone. The AT&T network did not scale appropriately and users were quick to complain about poor experience and long waiting times. While capacity planning could have helped prevent these issues, they immediately made apparent the relative weakness of wireless communication, as well as the protocol then in use. The protocol was not designed to work over slow networks with high packet-loss conditions and frequent disconnects.

In 2008, another industry started to bloom, which has become known as APM (Application Performance Management). Companies like AppDynamics and New Relic started to offer platforms that could monitor the bottlenecks, errors and issues occurring within web applications and databases. For eCommerce sites, APM can monitor the flows of actual transactions, from their website to the backend application and database.

Then in early 2010, when app stores became super popular, and millions of apps were suddenly available to instantly download, developers quickly realized that the tools used to develop Internet applications were not adequate for mobile apps. Web Performance and Web Performance Measurement companies that tried porting their web solutions to mobile apps failed. Even today, mobile app developers are in an inferior position compared to web developers in relation to understanding and troubleshooting issues.

In response, the traditional APM companies have extended their functionality from being backend applications to mobile applications; in addition, another crop of companies has emerged, including Apteligent, that are building APM platforms from the ground up, specifically for mobile.

Finally, in early 2017, the CDN and APM industries have started to converge. On the one end, there are CDNs trying to solve the problems with mobile application performance. At the other, there are mobile APM solutions trying to solve the same problems from an APM perspective.

However, no company in the world yet takes the best of both to offer a compelling new solution that leverages CDN global infrastructure and mobile APM built from the ground up, specifically for mobile applications.

Not until now, that is. PacketZoom is the first and only solution that provides advanced mobile APM functionality on a global scale. And we go further than that: because we can also do optimization. Hence, our Mobile platform is the only mobile APMO (Application Performance Monitoring + Optimization) solution in the world today that fixes performance problems related to the network. Thus, we can monitor application performance in the mobile last mile, *and* fix any problems that arise.

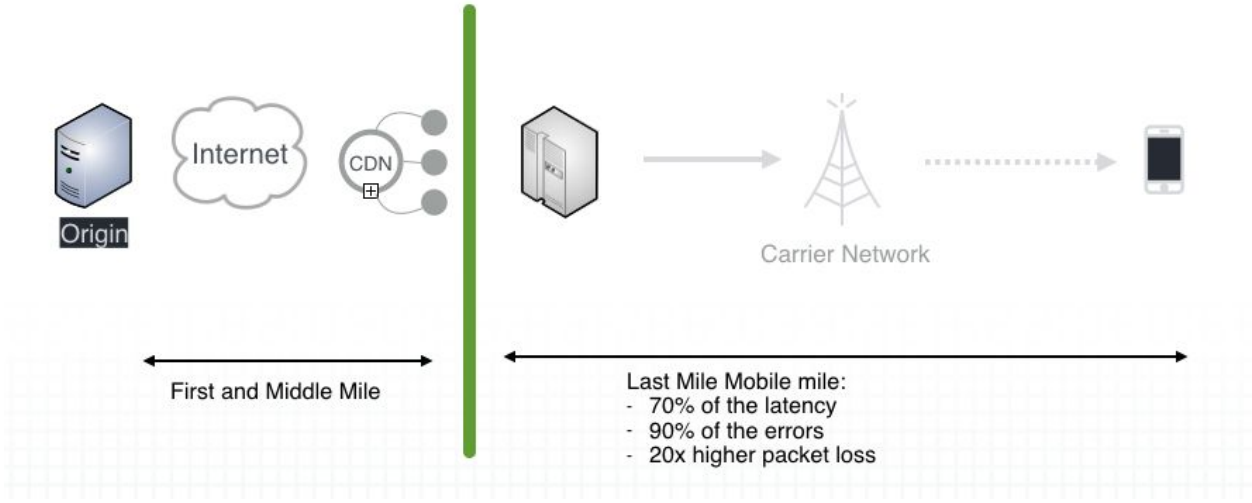
Problem Definition

Unlike mobile websites, where the development and operation teams are alerted of production issues and can quickly act to handle them, mobile applications are more of a black box. Once the application is released to the app store, the developer has little visibility and control. Oftentimes, the team that worked so hard to develop a new game or shopping application only realizes something is wrong when it is too late - through user reviews that badly affect the ranking of the app in the app store.

High Level Solutions

To gain greater visibility, increased control, and improve performance, mobile app developers are using solutions that were not necessarily designed to support their use case, but rather for a more generic web development use case. In many instances, these “new solutions” are only new names with the right buzzwords included. It is difficult for a mobile application developer to assess these technologies, and they can find themselves compromising on less than ideal solutions.

Unlike the cloud (i.e. the web), which is well connected through a "Fixed Connection," the last mobile mile relies on radio transmission, which introduces new challenges of speed and reliability. Due to the nature of these issues, and the fact that they are rooted in the wireless part of the network, they are more difficult to discover and troubleshoot. Resolving these issues requires a completely new way of thinking.



The last mobile mile is known to account for:

- Over 70% of the connection latency
- 10-20x increase in packet loss
- Over 90% of network errors

Based on PacketZoom Mobile Observatory (data collected from millions of real devices around the world) the following are typical latency, packet loss and error rate numbers:

COUNTRY	Disconnects	Packetloss	Latency (ms)
Argentina	9.10%	2.20%	812
Australia	6.20%	2.10%	367
Brazil	10.00%	3.00%	413
Canada	5.00%	1.10%	410
Chile	9.20%	1.60%	460
China	9.50%	5.50%	464
France	6.70%	1.10%	293
Germany	11.90%	1.80%	395
India	7.30%	1.50%	806
Indonesia	12.10%	2.50%	663
Japan	3.50%	2.20%	375
Malaysia	9.90%	3.20%	463
Mexico	8.00%	1.60%	414
Netherlands	4.20%	1.80%	206
Russia	13.30%	2.20%	370
South Korea	6.40%	3.00%	508
Spain	6.50%	1.30%	352
Taiwan	5.00%	3.00%	298
United Kingdom	5.20%	1.30%	231
United States	5.60%	1.30%	458

* Based on PacketZoom Mobile Observatory June 2017

While CDN's still represent an important part of the Internet infrastructure ecosystem with servers positioned at the "Edge of the Internet", they are mostly solving network issues related to the first and middle mile, and cannot do much to resolve last mile issues.

Similar arguments can be made in the mobile APM space, with solutions that worked well for the web connected world, but don't provide similar benefits to developers building mobile applications. Given the variability of mobile network conditions, and the unique use case in which mobile users move around and are exposed to changing conditions even in a single session using the same network, the only viable approach for collecting accurate measurements is to be present on the end user device (i.e. via an SDK).

Some companies (e.g. New Relic, Apteligent) have taken the right technical approach and developed a Mobile APM solution leading with an SDK. These solutions have become fairly popular, indicating a strong demand; however, they tend to suffer from the following issues:

1. They are not optimized for mobile apps, thus consume unnecessary bandwidth;
2. They leverage the app users to collect the data (i.e. low costs for the user), but overcharge the app owner;

3. They provide alerts related to performance issues, but don't offer advice or tools to fix the problems.

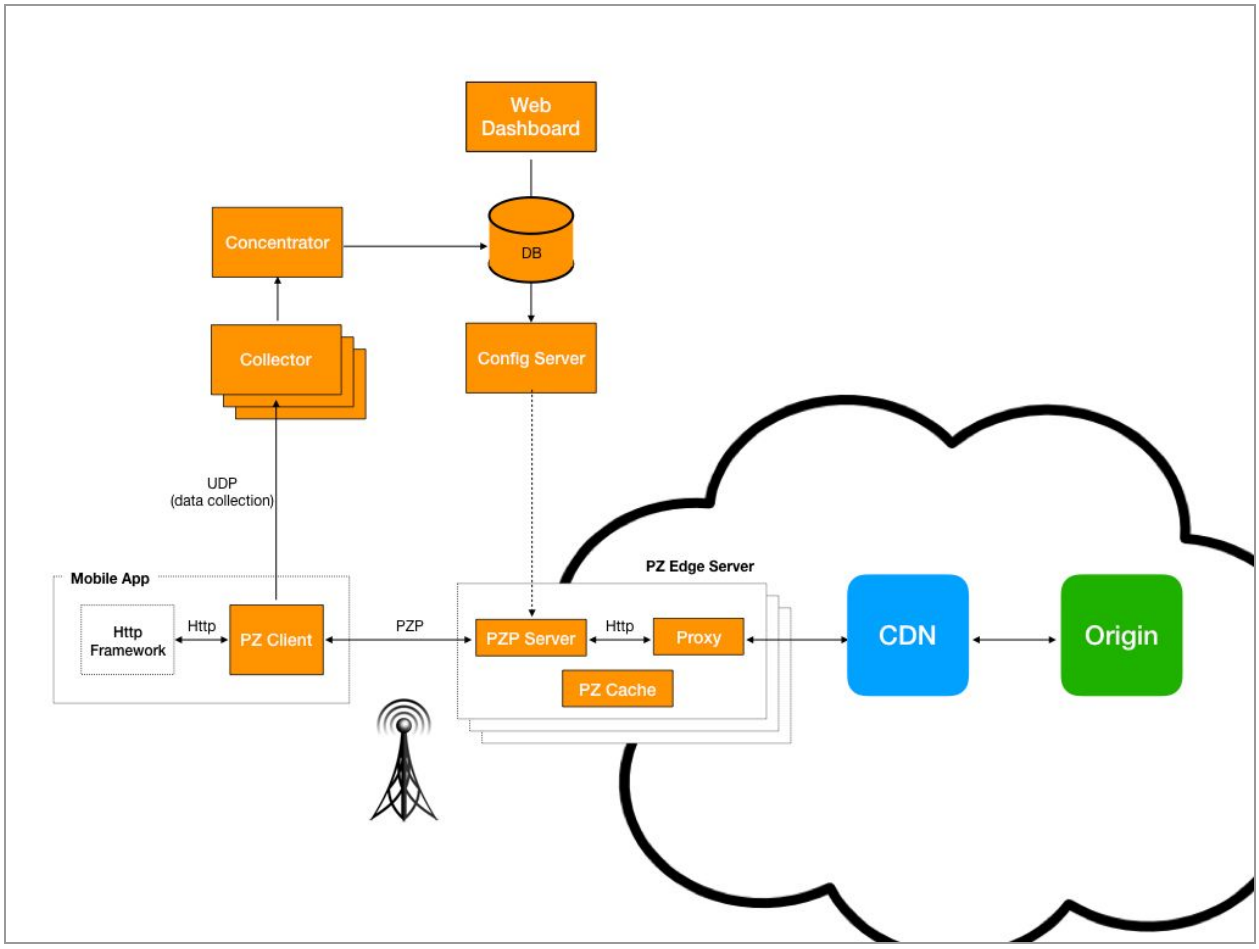
Solution Details

The PacketZoom Mobile APM (Application Performance Monitoring) & Optimization platform offers an end-to-end solution to analyze, detect, and resolve mobile app networking issues in real time without even changing the app code.

The PacketZoom (PZ) platform combines two essential products:

1. **Mobile IQ** offers real time insight into mobile app performance, as well as control functions to manage the app behaviour without making code changes.
2. **Mobile Expresslane** is a modern mobile networking stack designed for apps to speed up content downloads, curtail network errors, and reduce infrastructure costs.

Service Architecture



Main Service Components

Client Side:

- **HTTP Framework**: Third party or OS-provided HTTP framework (e.g.NSURLSession, OkHttp etc.);
- **PZ Interceptor**: Part of PZ SDK which can hook into OS-level or 3rd party HTTP framework in order to intercept HTTP/S requests;
- **PZ Client**: A core piece of PZ SDK that implements a custom UDP transport to communicate with PZ cloud in order to:
    - Serve requests
    - Collect performance metrics
    - Utilize local cache

- Control A/B tests
- Provide dynamic config params and more.

Server Side:

- **PZ Server**: Edge servers, which the client interacts with using UDP protocol, plays a key part in the last mile optimization;
- **PZ Cache**: server-side cache - similar to CDN;
- **Proxy**: runs on same box as Edge servers - transparent HTTP/s proxy which delivers content from origin servers, can keep persistent connection to origin and compress origin response for better bandwidth utilization;
- **Collector** - Collects performance information from each active SDK using a single UDP packet, validates the packet and forwards it to the concentrator;
- **Concentrator** - Processes data collected from multiple collectors and dispatches it to redshift and pipelineDB.

Service Enablement

In order to be able to start using the PacketZoom mobile networking platform, the developer should follow a straightforward integration process, which includes:

(a) Mobile SDK integration;
(b) App account setup (through the PacketZoom dashboard), the service for an app.

Once the SDK is functioning inside one or two devices, the developer can schedule functional sanity testing to ensure that the SDK does not interfere with app functionality. Since PacketZoom SDK is used by millions of devices daily, this step should be minor.

Next, the developer should schedule performance testing, meaning the SDK should be installed on a larger scale and tested by a large number of devices for proper A/B testing. We recommend at least 1,000 devices, effectively meaning a new app release to a limited user base.

SDK Integration

PacketZoom SDK is compatible with multiple development platforms. Detailed developer's instructions are available here, or via this short video.

Account Configuration

Setting up a PZ account is quick and easy, and involves only a handful of steps, as described in the PacketZoom Account Setup video.

Performing A/B Testing

The PacketZoom platform includes a built-in module for performance A/B testing.

There are two ways to segmentize PZ and Transmission Control Protocol (TCP) traffic:

1) By device - devices are selected on a round-robin basis to get the PacketZoom service;
2) By session - user sessions are selected on a round-robin basis to get the PacketZoom service.

Once setup is complete, the system serves only the percentage of devices/sessions defined in the PZ dashboard, which can be easily changed at any time without code changes. This includes termination of A/B testing and a move to the desired state (100% PZ or TCP).

Service Operation from A to Z

Real Time Performance Analytics

Once the PacketZoom SDK is integrated and deployed with an app, it is immediately ready to capture HTTP connection and session information. This functionality is offered free of charge to developers interested in understanding the performance of their app.

The data is collected on the device and packaged so that it can be transmitted to the closest collector in a single round trip. This avoids unnecessary bandwidth consumption from the application. In the background, as part of the data pipelining process, data is fetched from all the collectors to the concentrator and made available for the Mobile IQ portal to present. The entire process takes under one minute, so the data presented on screen is very near to real time.

HTTP Connection Optimization

When opening HTTP connections using a standard network library, developers have to specify connection parameters, such as timeout and buffer sizes. These parameters will be used to set up the connection in real time. Unfortunately there is no way for the developer to know the network that his/her code will be executed on, so developers often end up using default

parameters. This tends to result in suboptimal connections, for example, over a slow 3G network, the default timeout will be too short, but over a fast WiFi network, it will take too long.

PacketZoom's big network data offers developers the chance to override default connection parameters in real time and fine tune the connection perfectly for the underlying network. Customers who are interested in this free functionality can simply turn on the "HTTP Optimization" feature, and the PacketZoom SDK will immediately begin to select the optimal connection settings for that user/session, providing a network experience specifically customized to each user, and their unique network conditions in that moment.

Real Time App Control

The PacketZoom platform was designed to allow real time control of live mobile apps. Each time that a configuration change takes place on the dashboard, the changes are stored in the database and almost immediately pushed to the config server.

Each time a PacketZoom powered app is initiated (i.e. activated or moved from the background to the foreground on the phone), it fetches the most recent config file from the server and pairs up with it. This means that each new user session has the most recent configuration, and allows developers to change A/B testing parameters, start/stop services etc. without pushing out a new version of their application, or having to make any code changes themselves.

Last Mile Optimization

Server Discovery and Session-Init

To minimize failures, each multicast session-init process automatically discovers the optimal server available for that particular user in that moment in time. This eliminates the need for potentially unreliable, laggy DNS lookups from the user's local environment. Load balancing therefore completely bypasses one of the traditional Achilles' heels of traditional CDN technologies: discovering the optimal server for best performance.

"The Secret Sauce" (PZ Protocol)

Since PacketZoom collects billions of data points about each network, this big data can be used to make better transport decisions, such as the optimal throughput per user connection. Below are the main advantages that PZ's protocol offers users over TCP. Those without a networking background can more quickly learn about the benefits of the PZ protocol through this short "How it Works" video.

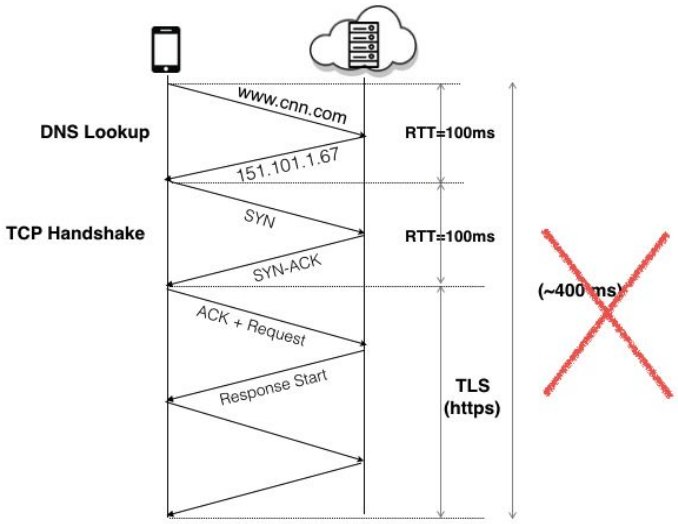
No Attacks on DNS

So, now let's get into how the PZ protocol actually works. As soon as the app starts, the SDK tries to establish a session with our servers. Initially, the SDK has a list of root servers available, and based on the user's timezone, a multicast init request is made to these root servers. The root servers then respond with a list of closest/optimal servers, along with validating the client and establishing a session. The SDK retains this list, so that in the future it can directly connect with these servers, allowing us to do **dynamic load balancing** of servers.

Basically, we can add a new server and within five minutes, start serving traffic without involving any DNS changes. This means that not only can we save round trips in getting the IP address, but crucially, we are also safe from any attacks on DNS.

Skipping a Lengthy TCP Handshake

In order to establish a secure HTTP connection, TCP has to make four round trips to the server. Assuming that the average latency over a wireless connection is 100 milliseconds, this means wasting 400 milliseconds before downloading even the first byte. PacketZoom protocol eliminates the need for this time-consuming start, and thus is able to accomplish the initial handshake between the client and server in much quicker time.

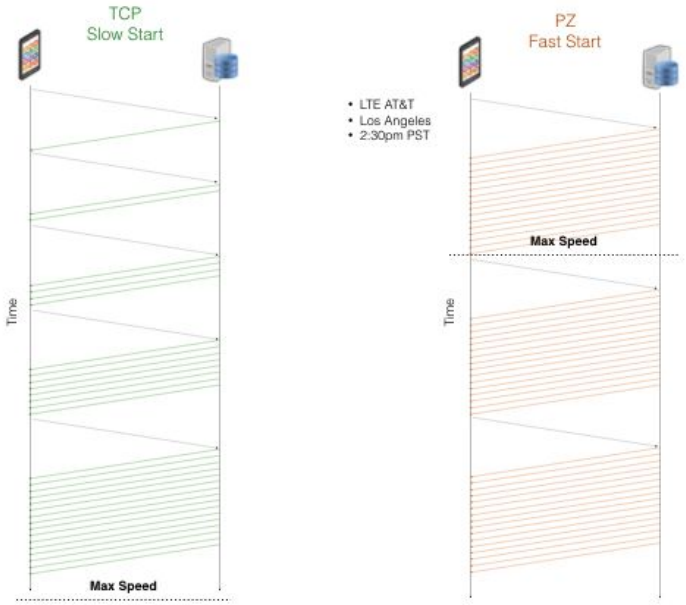


Avoiding Slow-start

Slow-start is part of the congestion control strategy used by TCP, the data transmission protocol deployed by many Internet applications. **Slow-start** is used in conjunction with other algorithms to avoid sending more data than the network is capable of transmitting and avoid

causing network congestion. When used over a high latency connection, TCP slow-start has a significant impact on the user experience. In many cases, it means waiting for a significant period of time.

To avoid the need for guessing the maximum bandwidth available, PacketZoom collects and leverages historical data based on the carrier network, time and location. When starting a new connection, PacketZoom protocol starts faster, reducing the number of round trips it takes to reach ideal throughput.

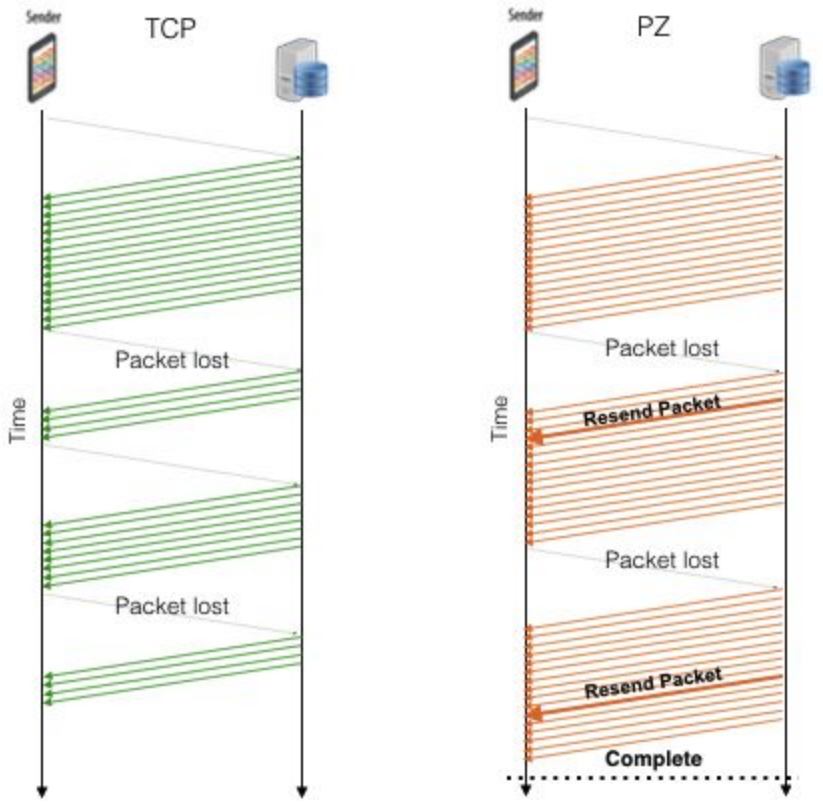


Better Packet Loss Recovery

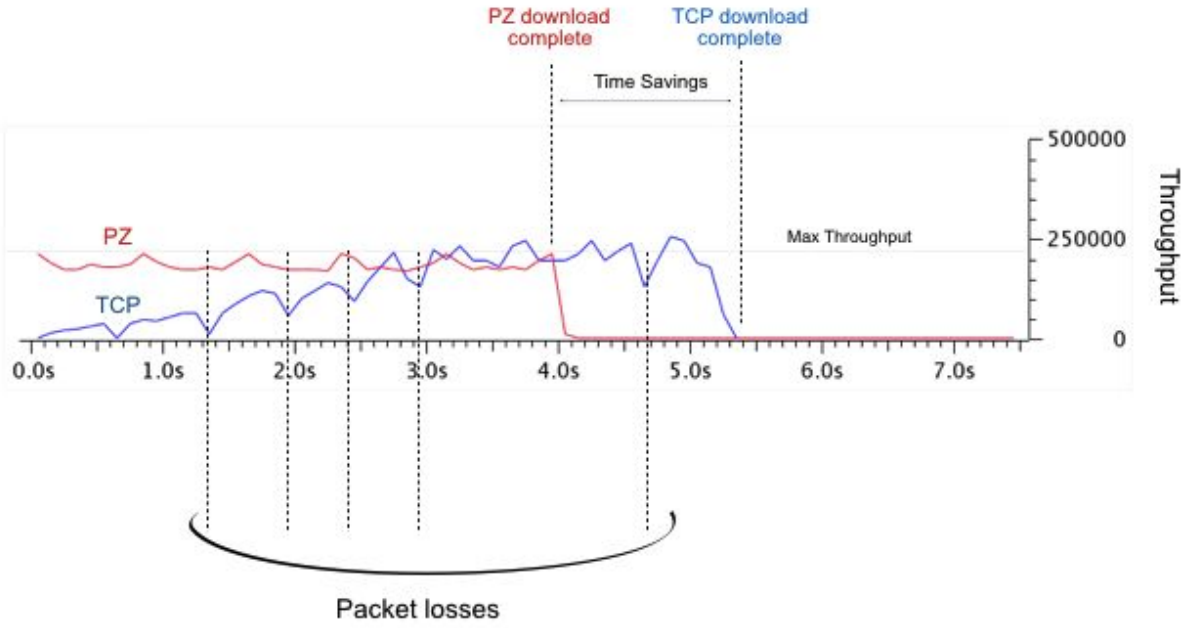
Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. In traditional non-mobile networks, packet loss is typically caused by network congestion, and is measured as a percentage of packets lost against packets sent.

Over three decades ago when TCP was first designed, packet loss was a rare event and the only possible cause was a congested network. When TCP detects packet loss, it "backs off" (i.e. slows down) the transmission rate to allow the congestion to vanish. However, with the surging use of wireless networks in recent years, other more common causes have become widespread, all related to environmental issues.

PacketZoom is able to detect the real cause of packet loss and respond accordingly. If the cause is unrelated to network congestion, it simply resends the lost packet in the next round trip and keeps the transmission rate the same.



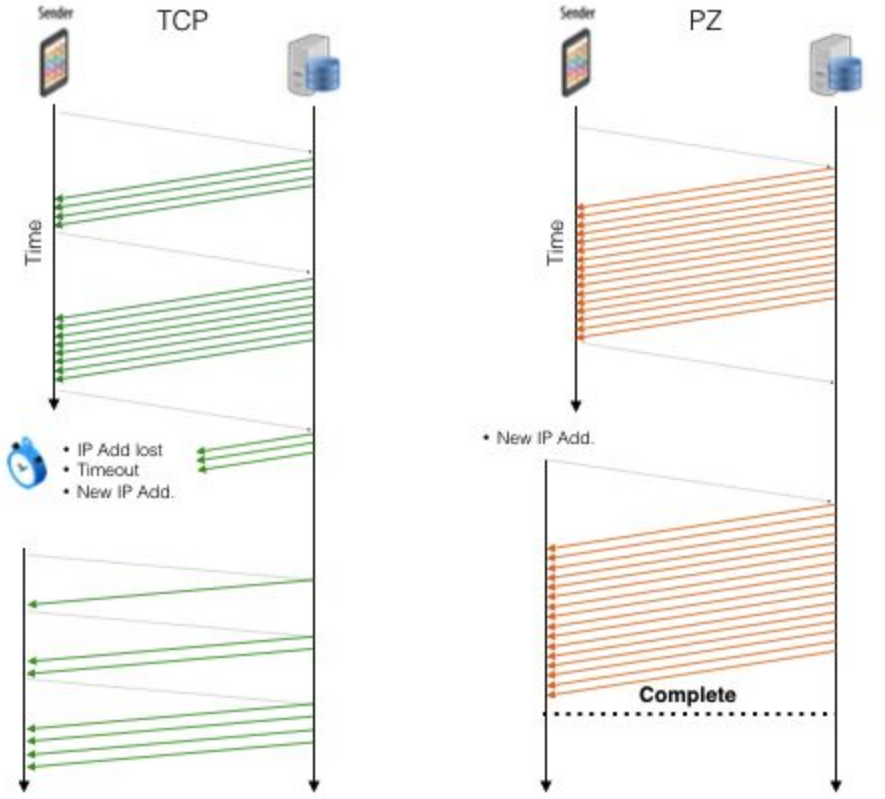
The example *tcpdump* below reveals how TCP and PZP use the available bandwidth while downloading a 4MB file. The blue line (TCP) starts slow and suffers from packet loss backoffs during the download process. The red line (PacketZoom) starts quickly and keeps constant maximum throughput across the download, allowing the download to complete more quickly.



Connection Loss Handling

Another assumption made by the original architects of TCP was the use of IP address to identify the client. Unfortunately a user's IP address tends to get lost when using wireless networks, leading to the need for the assignment of a new IP address. This can happen when entering a deadzone (walking into an elevator, driving through a tunnel, etc.) or simply when swapping networks (WiFi to cell, or vice versa). IP loss is a problem with TCP for various reasons: having to wait for timeout before re-starting a device; not being able to resume an existing download; losing time by having to start over, and so on.

PacketZoom, however, has added another unique identifier to the client so that when the IP address changes, we can still identify the device and thus quickly resume the session. This feature both helps reduce network errors, and improves API reliability.



Accelerating Static Content

Similarly to CDN services, PacketZoom Mobile Expresslane includes server side caching functionality. Server side caching does two things: (1) it speeds up content delivery; and (2) it reduces the number of calls to the CDN origin, thus reducing associated costs.

In addition to server side caching, PacketZoom offers in-app caching for commonly used files on a per user basis. Due to device storage limitations, this is a smaller cache, but it offers even greater benefits when there is a cache hit, as the content is served locally and this saves 100% of the network overhead for the selected asset.

PacketZoom's main benefit when serving static content is rooted in the reduced number of client-server round trips that are required to download files. As explained in the "Avoiding Slow Start" and "Handling Packet Loss" sections, thanks to its big networks data and more intelligent transport decision-making, PacketZoom can download the same amount of content using significantly fewer round trips and as such, deliver content much more quickly, and with less server overhead, than traditional methods.

Accelerating Dynamic Requests

PacketZoom excels at accelerating dynamic requests, thus providing a huge benefit in the areas of performance and cost savings:

- Persistent Connections - A key advantage of using PacketZoom for dynamic requests is that the origin server has to manage far fewer connections, coming only from our servers, versus connections from million of devices to and from one's customer base;
- Lower RTT (Round Trip Time) - Slower connections from mobile networks use up server resources for longer periods of time, thus adding to load problems. In the case of PacketZoom, all connections come from our servers, which are on a wired connection;
- Continuity - A failed attempt in HTTP would lead to a retry and the server has to do the same processing again; by contrast, the PacketZoom origin server does the processing only once, and our servers take care of the rest, resulting in fewer disconnects and decreased server load due to retries of failed connections.

Accelerating Third Party Domains

The PacketZoom SDK can intercept any HTTP traffic in an instrumented app and can route selected content (using predefined filters) to the PacketZoom server via the PacketZoom protocol. This efficient design allows PacketZoom to accelerate third party content without additional integration/provisioning. This means that application owners can accelerate SDK traffic (which they currently have no control over) in the same way that they can accelerate their own content.

Prioritizing Application Requests by Domain

TCP/HTTP is limited when it comes to allowing developers to define request priorities. Developers can order their requests, but can only hope they will be executed in the order they are sent. There is no guarantee this will be the case. However, PacketZoom protocol was designed to help developers gain more control over their mobile apps. Domains can be assigned with priorities in order to ensure the most important requests will get high priority. A popular use case would be prioritizing or deprioritizing ads to find the right balance between quality of user experience and monetization.

Blocking Domains

While a rare incident, third party services can sometimes suffer from outages or poor performance. To handle these cases, developers are using programmatic "kill switches" that allow them to immediately shut down service. When using PacketZoom, developers can temporarily block a domain, even if they forget to implement a kill switch.

Rewriting URLS

Another way to manage developer errors or production issues is via request rewrite. Such a feature could also be extremely useful for testing across CDNs without releasing new code. This can be accomplished from the PacketZoom Mobile IQ dashboard.

Service Scalability and Reliability

To provide end customers with a more reliable mobile experience, the PacketZoom service includes multiple layers of redundancy, including a robust fail-forward design and intelligent load balancing, backed by a global network of servers.

Global Servers on Multiple "Clouds"

As a flexible software stack that can operate on almost any cloud server, PacketZoom can utilize multiple cloud services simultaneously to host our service worldwide. All regions are serviced from at least two different cloud service providers. This redundancy makes PacketZoom service resistant to the kinds of outages in network or server equipment that is typical within single data center deployments.

Scalability to Fit Changing Needs

Flexibility is so important for mobile apps because usage can grow and change rapidly. The PacketZoom stack has been proven to scale directly in relation to the number of clients served. When more clients connect to PZ servers, any newly-added servers are immediately and automatically discovered by the next client to connect.

This allows us to rapidly scale the server infrastructure in any part of the world by simply installing our software on new servers and pushing a config button. As the PacketZoom service handles server selection/connection itself using a known list, there is no involvement of DNS Time to Live (TTLs) required, allowing rapid scaling in near real time.

Fail Forward Design

The multicast session-init process must return a successful response before the PacketZoom client stack attempts to reroute any user requests. Even in the event of a catastrophic failure of the PacketZoom infrastructure, user requests keep going to their original destination, whether they are using a customer's server or a CDN.

Service Security

PacketZoom Infrastructure

PacketZoom uses heterogeneous infrastructure to support the PacketZoom service. The architecture has been designed to maintain high availability and security across our platform. Our servers are located in PoPs (points of presence) around the world and built with high levels of redundancy in order to eliminate downtime: should one PoP fail, for whatever reason, another can take over its load.

Each server is independent and does not maintain state with respect to other servers, thus enhancing the durability of the service. Server communication is handled via secure SSH transactions and self-signed TLS certificate. Authentication is achieved via public key cryptography with absolutely no use of user/password combinations. All servers are locked down with a minimally exposed footprint.

How Does PacketZoom Cache Encrypt (SSL) Content?

On receiving an HTTP(s) request from the client over PacketZoom Protocol, our proxy servers make a new HTTP(s) request to the origin server. On receiving the response, our proxy server (behaving as a HTTP(s) client) decrypts the response and caches it (if caching rules apply).

Does PacketZoom Keep Private Session/User Info?

PacketZoom does not store private information unless it is part of the request/response and we respect the caching header rules. More specifically, PacketZoom saves the URL in our databases for analytical purposes only, and we cache the response on our servers (based on the caching rules) solely for performance improvement purposes. We do not keep any other data related to the request.

How Does PacketZoom Handle HTTP(s) Traffic?

The PacketZoom SDK intercepts secure traffic request on the client before any network activity is initiated. The request is then transmitted to PacketZoom servers using our proprietary protocol. PacketZoom servers then forward the request to the origin server over a standard HTTP(s) connection.

Does PacketZoom Store SSL Certificates to Manage Secure Traffic?

Certificates are used by traditional web browser/server models where the browser needs to confirm that the public key received actually belongs to the server it is attempting to

communicate with, and not by a malicious server trying to impersonate it. To avoid this, the browser confirms the identity of the server with a trusted third party. Since PacketZoom protocol was designed specifically for mobile apps and is shipped as an SDK, the server public key is already securely stored on the device.

How Secure is the PacketZoom Protocol (PZP)?

PacketZoom is a proprietary secure protocol that is designed around the following three principles:

1. Data Encryption
   By default, all data transmitted using PZP is encrypted using the CHACHA20 encryption algorithm (used by Google in TLS). This means that middleman systems cannot intercept PacketZoom content, manipulate images, etc.

2. DDoS Prevention
   The PacketZoom protocol includes built-in mechanisms to protect against Distributed DoS attacks.

3. Prevention of Replay Attacks
   Encryption keys used in PZP are ephemeral and refreshed frequently within a given session.

Disaster Recovery

The PacketZoom support infrastructure is built on AWS, which guarantees high availability and auto recovery if necessary. Production services take advantage of the stateless nature of the PZ service and can be built and rebuilt on a heterogeneous menu of cloud and dedicated service providers.

Business Benefits

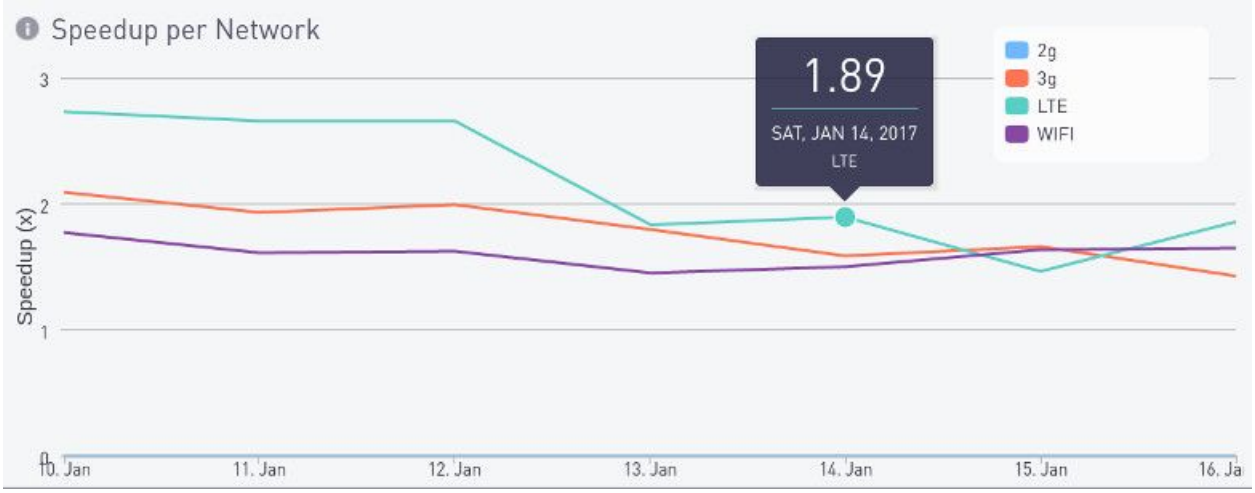
Higher User Engagement

Be it playing a favorite game or utilizing a shopping app, thanks to better network technology, PacketZoom users are able to accomplish more during their session. Customers such as Goat (sneakers), News Republic (news) and Sephora (makeup) reported 10-12% higher user engagement measured by:

- Average number of images downloaded per session
- Average number of API calls per session (product search, find hotels, flights, etc.)
- Average number of page views (news)
- Lower percentage of game abandonment

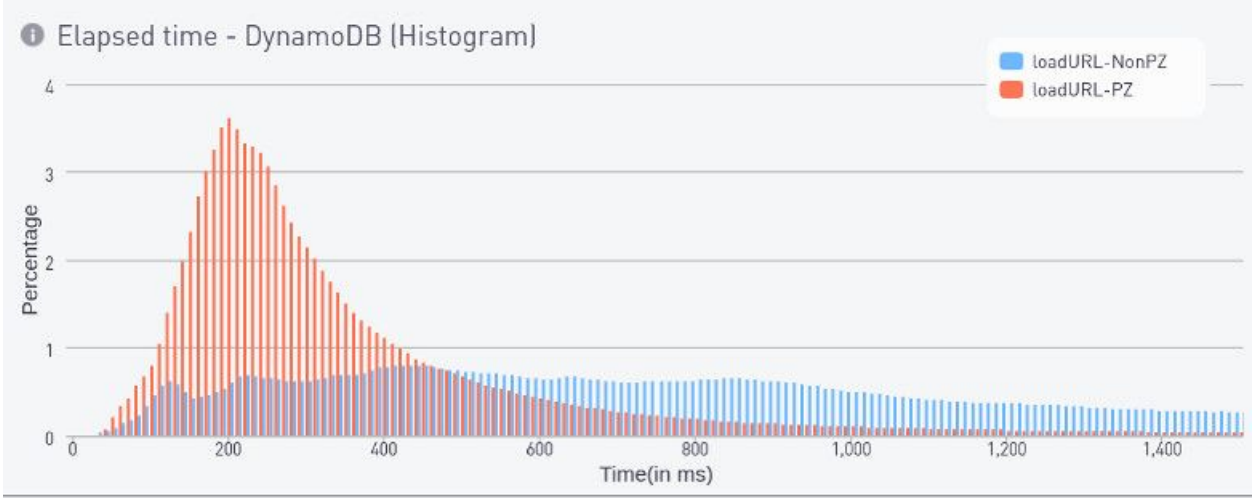
Faster Content Download

Thanks to PacketZoom's smart and modern protocol, quicker handshake process, smart-start and our better packet loss recovery, static content such as images, videos, and game bundles download 2x-3x faster.



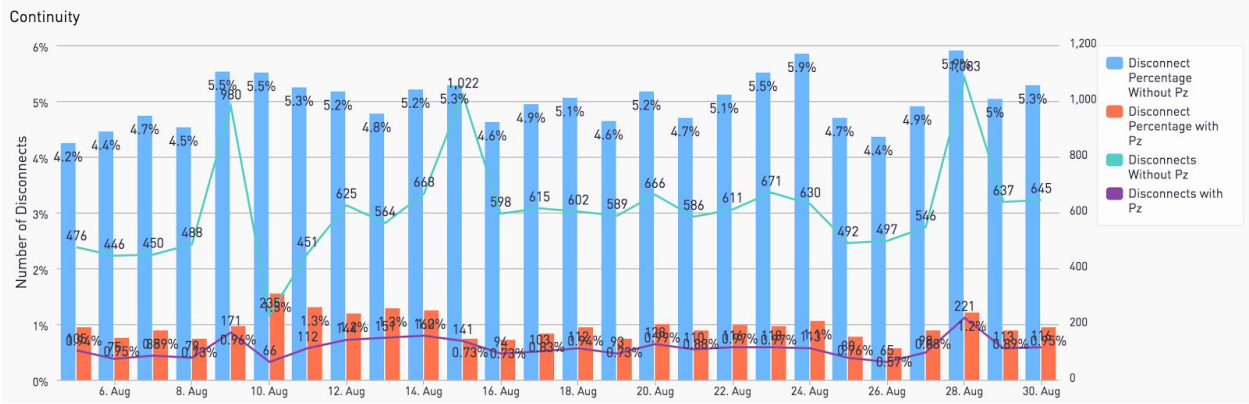
Faster API Response Time

Thanks to our smart-start functionality and better packet loss recovery, the PacketZoom protocol is able to retrieve large API responses from remote servers 50-70% faster.



Fewer Network Errors = Reliable API

Thanks to the fact that PacketZoom protocol does not use the IP Address as the only way to identify a client device, we can significantly reduce the number of user disconnects.



In Conclusion

PacketZoom's revolutionary mobile app networking solution allows mobile app developers to put their focus on building great apps. We provide a turnkey solution, which successfully boosts user engagement and solves quality of experience issues resulting from poor service across last mile mobile networks.

The PacketZoom Mobile Expresslane platform helps mobile app developers and publishers to: (1) understand their app performance in real time; (2) control the app behaviour without deploying code changes; and (3) accelerate app performance using a modern networking stack.

PacketZoom is currently serving tens of millions of daily users. Customers using the PacketZoom solution report an upsurge in user engagement and a significant reduction in user abandonment.

There is a huge amount of activity happening in the rapidly growing market of CDN, DPM and APM and various big players are teaming up to offer CDN and DPM solutions, including Cisco-AppDynamics, Akamai-Soasta and Instart Logic-BlueTriangle.

However, no company in the world right now is leveraging CDN global infrastructure and APM built from the ground up, expressly for mobile application, other than us. PacketZoom is the only solution out there that offers advanced mobile APM functionality worldwide. APM is more sophisticated than DPM as it can analyze and monitor applications down to a single line of code. APM monitors interdependencies within the application, mappings, transaction flows, and more.

And remember, we can also do optimization.

PacketZoom's Mobile Expresslane platform both monitors and fixes problems related to the application, and again, we are the only mobile APM solution in the world doing it.

Call to Action

The PacketZoom solution was designed with ease of integration and straightforward use in mind. As long as you are using standard HTTP libraries, integrating our SDK should take a matter of minutes.
What are you waiting for? Start using PacketZoom today for better performance at a lower cost.