PacketZoom Powers Mobility for Today's Changing Organizations

Table of Contents

Introduction: Every Business is Becoming More Mobile and Connected

As organizations become more connected and dynamic, mobile applications are on the rise. According to Gartner, 40 percent of sales organizations will rely primarily on mobile digital technology for their sales force automation initiatives[1].

Businesses of all kinds are embracing mobile apps. It's a great opportunity for developers that can deliver the superior performance and user experience their end-customers want.

But today's marketplace is competitive. Poor last-mile performance can introduce latency and put a drag on any mobile app. That's important, because 79 percent of users say they would retry a mobile app only once or twice if it failed to work the first time.

To succeed, DevOps, product managers, and mobile developers need a better way to boost the reliability and performance of their mobile apps. PacketZoom was designed to do just that.

A Simple Solution to the Mobile Performance Challenge

PacketZoom enables developers and project managers to unlock the full potential of their mobile apps and deliver a superior user experience—anywhere. This mobile app networking platform makes mobile app content delivery and API access faster and more reliable—simply and cost-effectively.
PacketZoom was built for simple, easy operation, with no changes required to existing code, configurations or infrastructure. Developers install the PacketZoom SDK in their app and set up a regex to specify which of their requests need to be accelerated. The chosen HTTP and HTTPS requests are automatically routed through PacketZoom's worldwide presence of servers. A PacketZoom client software development kit (SDK) downloads content over the WiFi and mobile networks using the PacketZoom protocol, an innovative networking protocol designed specifically for mobile apps.

The PacketZoom Protocol Stack: Created for Mobility

At the heart of the solution is the PacketZoom protocol stack. Written on top of User Datagram Protocol (UDP),it operates completely in user space, with no special kernel facilities required. This means that it can work on all mobile devices in the world today.

The PacketZoom protocol stack supports two key capabilities: server discovery and session init, as well as transport of user data to the client. Let's take a closer look at how it enhances performance throughout both steps.

---

[1] http://www.gartner.com/smarterwithgartner/the-customer-experience-in-2020/

Step 1: Server Discovery and Session Init

Each PacketZoom session spans the lifetime of a mobile app session. At the start of the session, the PacketZoom SDK uses one network round trip to discover the best PacketZoom servers in the user's region, establish a session and get the latest configuration. Expediting these processes helps save time, enhance performance, optimize configurations, and enhance security.

Save Time By Avoiding DNS Lookups on the Client Network

Once a PacketZoom server is bound to an app client session, all the PacketZoom requests go to the same server. This saves at least one round trip for each and every connection the app would have made with systems—at least once per domain name, even if the DNS caching is working well. With the HTTP practice of "sharding" domains, this saves a lot of network "dead" time for most apps.

Discover the Best Performing PacketZoom Servers Closest to the User

PacketZoom seeks out and discovers the best-performing PacketZoom server that's closest to a user at a given moment, enabling the server to gracefully failover and load-balance in real time. Any new client session discovers the latest list and state of PacketZoom servers automatically.

Deliver the Latest Configuration

The PacketZoom dashboard provides the latest configuration as defined by the app developer. This allows almost every configuration change made by the owner (via PacketZoom dashboard) to propagate to end-users in minutes, with no DNS or TTL issues.

Optimize Security

Exchanging encryption keys ensures that every session uses new keys for encryption, ensuring forward secrecy.

Step 2: Transporting User Data from PacketZoom Server to the Client

Once the session init is successfully completed, the SDK is ready to route selected HTTP/S traffic through the PacketZoom stack. Selected HTTP/S calls traverse the mobile last mile over the PacketZoom protocol to the nearest PacketZoom server. The PacketZoom server backend works like a caching proxy for static content and as a transparent proxy for non-cacheable dynamic content.

The PacketZoom transport protocol plays a key role in enhancing performance over the mobile last mile between the PacketZoom SDK in the app and the PacketZoom server endpoint. The result is fewer congestion issues and smarter buffer handling for better performance.

Context-Aware Congestion and Loss Handling
The PacketZoom server backend utilizes our extensive database of network conditions based on users' context. It considers network type, signal strength, carrier, location, device type, time of day, and other variables. These factors enable us to extrapolate network properties such as expected throughput, packet loss,and round-trip times, then use that insight to ramp up throughput.
The same knowledge base also lets us make better decisions about back-off algorithms from packet drops during an ongoing session. Estimating whether a timeout is due to packet loss, due to congestion or due to media-loss is challenging, so we employ our in-depth knowledge about mobile networks worldwide to help make the call.

Smarter Buffer Handling
PacketZoom uses proprietary techniques in the protocol and on the receiving side that obsoletes the concept of limited receive windows. This also means that the packets for a particular transfer can arrive wildly out of order, yet still be useful to the client. We use this to turn techniques like SACK on its head. PacketZoom packets are rich in information and facilitate the use of a technique we call "intelligent hole filling".

With PacketZoom, apps require fewer retransmit round trips than would have been required in the TCP world. It's possible because the solution is based on native apps running on mobile devices, not a web browser.

Integrating and Using PacketZoom with Mobile Apps

PacketZoom works seamlessly with developers' mobile apps through a simple SDK integration with absolutely zero server-side changes or DNS modifications. It requires just a few minutes of developer time to put the goodness of PacketZoom to work.
The PacketZoom client stack captures outgoing HTTP/S requests at the client library level (NSURLSession, HTTPURLConnection, OkHTTP etc). From here, the SDK selectively chooses which URLs to proxy through the PacketZoom server infrastructure. The client stack automatically discovers the closest and best performing PacketZoom servers to connect to in real time at the start of app session.

Discovering the Optimal Server for Performance

At the start of the app session, the PacketZoom SDK uses a UDP "multicast" of probe packets to query PacketZoom servers in the same time zone about the best server to use for the proxy. This server discovery process bypasses any requirement for DNS—and the issues it can create for mobile clients.

PacketZoom doesn't use an actual IP multicast mechanism, but simply employs UDP packets sent to multiple IPs in the region.

Dependable and Scalable

To give end-customers the reliable mobility experience they expect, PacketZoom service includes multiple layers of protection for customer data, including a robust fail-forward design and intelligent load balancing, backed by a global network of redundant servers.

Fail Forward Design

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

Intelligent Discovery of PacketZoom Servers

To minimize failures, each multicast session init process automatically discovers the best and fastest 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 completely bypasses one of the traditional Achilles Heels of traditional CDN technologies.

Global Redundant Servers on Multiple Cloud Infrastructures

A flexible software stack that can operate on most any cloud server, PacketZoom can utilize multiple cloud services simultaneously to host our service in all parts of the world. All regions are serviced from at least two different cloud service providers. This redundancy makes PacketZoom service resistant to outages in network or server equipment in individual data center infrastructures.

Scalable to Fit Changing Needs

Flexibility is important for mobile apps because usage can grow and change rapidly. The PacketZoom stack has been proven to scale directly with the number of clients. When more clients connect to PacketZoom servers, any newly-added servers are immediately and automatically discovered by the very 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. There's no involvement of DNS Time to Live (TTLs) required.

Ready to Learn More?

You've invested a lot of time, money, and hard work to build a great mobile app. It's time to ensure that it delivers the performance and dependability your users expect—every time. Getting started is easy. Integrating PacketZoom takes just minutes, requires no changes to your app or infrastructure, and requires no additional hardware or software.

To get started, visit us at www.packetzoom.com.