

# A Focus on Efficiency

*A whitepaper from Facebook, Ericsson and Qualcomm*

*September 16, 2013*

## TABLE OF CONTENTS

1	Executive Summary
22	Data Center Infrastructure
30	Building More Efficient Apps
42	Facebook for Every Phone
49	Qualcomm: Meeting the 1000x Challenge
53	Ericsson: Why Superior Network Performance Matters
68	Conclusion

In August 2013, Facebook, Ericsson, MediaTek, Nokia, Opera, Qualcomm and Samsung announced the launch of Internet.org, a global effort to make affordable Internet access available to the next five billion people.

As founding members of Internet.org, we believe it's possible to build infrastructure that will sustainably provide affordable access to basic Internet services in a way that enables everyone with a phone to get online. While the current global cost of delivering data is on the order of 100 times too expensive for this to be economically feasible, we believe that with an organized effort, it is reasonable to expect the overall efficiency of delivering data to increase by 100x in the next 5–10 years.

This effort will require two key innovations:

1. Bringing down the underlying costs of delivering data, and
2. Using less data by building more efficient apps

If the industry can achieve a 10x improvement in each of these areas—delivering data and building more efficient apps—then it becomes economically reasonable to offer free basic services to those who cannot afford them, and to begin sustainably delivering on the promise of connectivity as a human right.

To make this a reality, Internet.org partners, as well as the rest of the industry, need to work together to drive efficiency gains across platforms, devices, and operating systems. By creating more efficient technologies, we will be able to speed up the roll out of more sophisticated technologies that provide higher quality experiences

to more people in developing countries, while also enabling the industry to continue growing and investing in infrastructure development. As we work together toward this common goal, we aim to achieve shared learnings and advances that move the industry and society forward.

In the first half of this paper, we examine how Facebook has approached the challenge of building more efficient technologies in order to connect more than one billion users. We will examine methods used to bring down the underlying costs of delivering data, first by looking at how building Hip Hop for PHP and the Hip Hop Virtual Machine allowed us to run 500% more traffic on the same number of servers, and how those servers are built from the ground up to serve that traffic as efficiently as possible. We will also discuss the groundbreaking efficiency and sustainability gains we've achieved in our data centers and with the Open Compute Project.

After that, we will explore the challenges of creating more efficient apps that use less data. Efficiency gains achieved on the Android platform will be discussed, as well as improvements in image rendering. We'll also discuss Facebook for Every Phone, our apps built specifically for low-bandwidth phones. We cover the establishment of dedicated Facebook facilities for testing apps and power consumption across devices under different network conditions, in addition to the use of server side processing, image compression and a cafeteria approach for accessing fonts and languages for feature phones.

Many of the technologies we outline in this paper have been released as open source software for the community to use and improve. At Facebook, we are building solutions to unprecedented scaling and connectivity challenges that other companies will start to experience as more people from diverse geographies, network connections and devices share more types of content and make new connections. We believe the best way to move both our company and industry forward is to be open and share our learnings and mistakes so that we can collectively build better services for the world. In this section, we detail our open sourced PHP compiler and virtual machine. We'll also discuss some of our open source big data tools like Corona, Avatarnode and our improvements to Giraph. In the data center

section, we'll discuss our server and infrastructure specifications that are available to the world through the Open Compute Project.

The second half of this paper features contributions from Internet.org partners Qualcomm and Ericsson that look to the future. To ensure that telecommunications and technology leaders are ready for the next five billion people, building more efficient networks will be vital. In chapter 5, Qualcomm presents an overview of the “1000x initiative,” a plan to expand global wireless capacity by 1000 times. In chapter 6, Ericsson explores how achieving superior network performance lies at the heart of operator best practices today as global smartphone penetration achieves critical mass.

### *The Journey to Efficiency*

Optimizing for efficiency has always been central to the way Facebook builds infrastructure. In its early days, Facebook ran on a single server that cost \$85 a month and had to handle traffic for every Harvard student who used the service. With the limited financial resources available at the time, Facebook had to be capable of running with only a few people maintaining the service, even as more and more schools joined.

Facebook.com was first coded in PHP, which was perfect for the quick iterations that have since defined our style of product rollouts and release engineering. PHP is a “dynamically-typed” programming language, allowing for greater speed than a “statically-typed” language like C++. As a programming language, PHP is simple to learn, write, read and debug. New engineers at Facebook can contribute a lot faster with PHP than with other languages, producing a faster pace of innovation.

However, it soon became clear that if Facebook were to scale exponentially, PHP would be an obstacle to achieving this efficiently. Although PHP enabled speedy shipping of new products and changes to the service, its continued use would require exponentially more servers. A new approach was needed.

To avoid having to rewrite the entire site in a new programming language, our infrastructure team developed new ways to run PHP faster. In 2010 we built a tool called HipHop <[https://www.facebook.com/note.php?note\\_id=280583813919](https://www.facebook.com/note.php?note_id=280583813919)>, that transformed PHP source code into highly optimized C++ before it hit the servers, allowing fewer machines to be utilized. HipHop for PHP allows 50 percent more traffic to be processed on the same machines than original PHP. This significant performance improvement was realized through various optimization techniques in the HipHop compiler and the run-time system including almost-serialization-free APC, faster serialization and JSON encoding, less reference counting, more compact binary code, improved memory allocation and faster compilation.

Although HipHop provided significant gains in the performance of Facebook's code, its reliance on static compilation made optimizing code time consuming. The team also developed a separate HipHop interpreter (HPHP) that required significant effort to maintain. To overcome those constraints, the team soon refined the approach underlying HipHop and determined that greater efficiencies could be achieved by building a virtual machine that would convert PHP directly into native machine code. A small team was created to experiment with dynamic translation of PHP code into native machine code, and in 2011 the Hip Hop Virtual Machine (HHVM) was launched. HHVM is a new PHP execution engine based on the HipHop language runtime <[https://www.facebook.com/note.php?note\\_id=10150415177928920](https://www.facebook.com/note.php?note_id=10150415177928920)>. that uses a just-in-time compilation approach to achieve superior performance while maintaining the flexibility that PHP developers expect. HHVM (and before it HPHPC) has realized a 500 percent increase in throughput for Facebook compared with Zend PHP 5.2.

Both HipHop for PHP and HHVM have been open-sourced by Facebook so that other developers can use PHP to build a flexible product they can update and iterate quickly while still scaling fast. <<https://github.com/facebook/hiphop-php>>

In addition to building efficiency into the way Facebook writes and executes code, we've made tremendous advances in hardware and data center efficiency over the

past three years with the Open Compute Project. Through this initiative, Facebook has been able to build one of the most efficient data centers in the world from the ground up and make important strides towards our goal of reducing data delivery cost.

## *Reducing the cost of delivering data*

### DATA CENTER INFRASTRUCTURE

When Facebook began exploring whether to build its own data centers in 2009, the company had grown to 600 employees and was serving around 200 million people. Since then, Facebook has grown to 5,299 employees and 1.15 billion users as of June 2013. With this kind of exponential growth in mind, an early decision was made to build our own facilities from the ground up, with a central focus on sustainability and efficiency.

In April 2011, Facebook brought online our first data center based on our own designs, in Prineville, Oregon. When we measured its performance against our leased facilities at the time, we found that it was 38 percent more energy efficient, at a 24 percent lower cost.

The efficiency gains were made possible by three core areas of innovation: cooling, power transformations and a “vanity free” server design that we will discuss later in the paper.

In 2013, we brought our newest data center online, in Luleå, Sweden. A state-of-the-art facility, Luleå has achieved a power usage effectiveness (PUE) of 1.04—among the best in the world, and well below the EPD-defined best practice of 1.5. Luleå is cooled by 100 percent outdoor air, and runs on 100 percent renewable hydroelectric power. We’ve also been able to reduce the number of backup generators by 70 percent compared with our previous designs. The interior of the server hall, from the servers to the racks, is provisioned with 100 percent Open Compute Project (OCP) designs.

These efficiencies and innovations are necessary at Facebook's scale. Every day, there are more than 4.75 billion content items shared on Facebook (including status updates, wall posts, photos, videos and comments), more than 4.5 billion "Likes," and more than 10 billion messages sent. More than 250 billion photos have been uploaded to Facebook, and more than 350 million photos are uploaded every day on average. The technical complexity of delivering a personalized Facebook experience—which generates a completely personalized view of the world for every one of our 1.15 billion users, every time they log in—requires that we process tens of thousands of pieces of data, spread across hundreds of different servers, in a few tens of microseconds. This complexity means we need to be able to process 1000x more traffic inside our data centers (between servers and clusters) than the traffic in and out of them. And that load is continuing to increase.

To help monitor our complex infrastructure and find new opportunities to carve out efficiencies, Facebook has built out an extensive big data infrastructure housing more than 250 petabytes of data. We use our analytics infrastructure to improve our product by parsing results from the many A/B tests we run each day and even to power parts of major products like Messages, which runs on HBASE.

Facebook's data infrastructure team has created a new platform that allows the running of a Hadoop cluster across multiple data centers by creating multiple namespaces that then create many logical clusters on top of the same bigger physical cluster. Name spaces can be divided, but they all have a common dataset that can span multiple data centers. This allows the teams to move data between all of Facebook's data centers.

Nearly every team at Facebook depends on custom-built data infrastructure for warehousing and analytics. There are about 1,000 people across the company, both technical and non-technical, who use these technologies every day. Over half a petabyte of new data arrives in the warehouse every 24 hours, and ad-hoc queries, data pipelines, and custom MapReduce jobs process this raw data around the clock to generate more meaningful features and aggregations.

Given our unique scalability challenges and processing needs—our data infrastructure crunches more than 10 PB of data a day - our team has to ensure that our systems are prepared to handle growth. Our data warehouse has grown by 4000x in the past four years, and is continuing to expand.

In response, we have devoted significant effort to improving virtually every aspect of Hadoop. Apache Hadoop was initially employed as the foundation of this infrastructure, and served Facebook well for several years. But by early 2011, the limits of that system began to be recognized. Because of our unprecedented scale, we've had to innovate on top of Hadoop and MapReduce and have created several new tools, including Corona, AvatarNode, Giraph, Presto and Morse.

#### CORONA

In late 2012 Facebook developed Corona<<https://www.facebook.com/notes/facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920>>, a more efficient scheduling framework that separates cluster resource management from job coordination. Corona lets us achieve low latency for small jobs, gives us the option to upgrade without disruption, and allows us to improve cluster utilization and scale more efficiently overall. Corona does this by introducing a cluster manager that tracks the nodes in the cluster and the amount of free resources. Because this manager doesn't need to track or monitor the job's progress, it can spend its resources making fast scheduling decisions while individual job trackers handle the rest. This design choice goes a long way to enabling Corona to manage more jobs and achieve better cluster utilization.

#### AVATARNODE

Optimizing HDFS is crucial to keeping our systems efficient and reliable. The way HDFS works is that clients perform filesystem metadata operations through a single server called the Namenode, and send and retrieve filesystem data by communicating with a pool of Datanodes. Data is replicated on multiple Datanodes, so the loss of a single Datanode should never be fatal to the cluster or cause data loss.

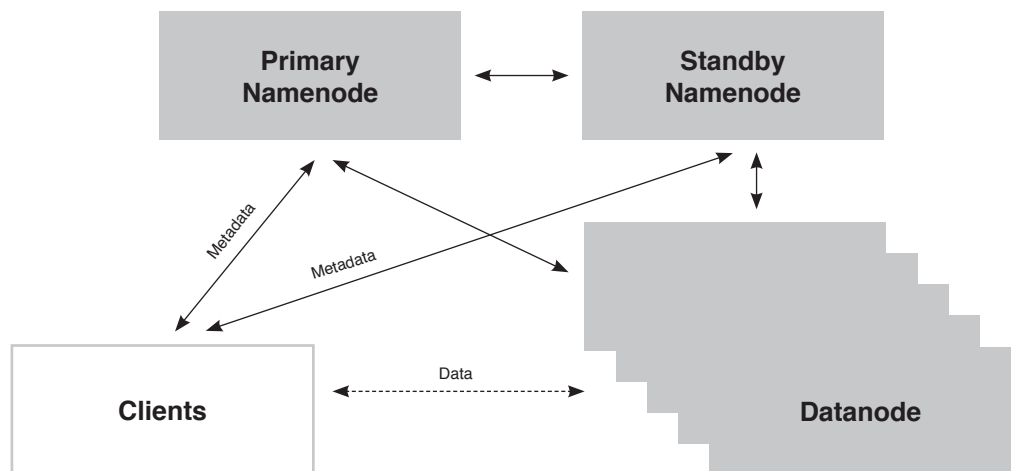
However, metadata operations go through the Namenode, so if the Namenode is unavailable, clients can't read from or write to HDFS. To solve this single-point-of-failure problem, Facebook built a system called AvatarNode. The AvatarNode



runs on our largest Hadoop data warehouse clusters and offers a highly available Namenode with manual failover.

AvatarNode, which is now available through open source, works by wrapping the existing Namenode code in a Zookeeper layer. It is running on Facebook's most demanding production workloads and plays an important role in increasing the reliability and efficiency of our HDFS clusters.

### *Simplified HDFS Achitecture: Highly Available Namenode*



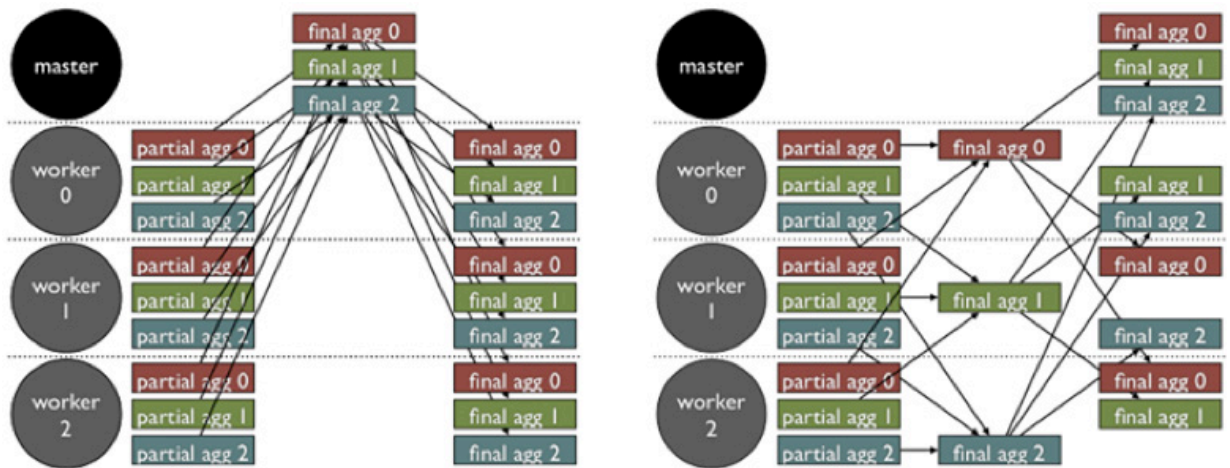
#### GIRAPH

Facebook's social graph comprises more than a trillion edges between people and their friends, Pages they like, places they've visited, apps they use and countless other connections.

Analyzing these real-world graphs at this scale with the software available last year was impossible. We needed a programming framework to express a wide range of graph algorithms in a simple way and scale them to massive datasets. With some homemade improvements, Apache Giraph became our solution.

We ended up choosing Giraph because it directly interfaces with our internal version of HDFS and talks directly to Hive. And since Giraph runs as a MapReduce job, we can leverage our existing Corona infrastructure stack. Giraph also performed faster than other frameworks, and its graph-based API supports a wide array of graph applications in a way that is easy to understand.

We made several modifications to Giraph to allow it to scale to our needs. The first was to make Giraph allow loading vertex data and edges from separate sources. Each worker can read an arbitrary subset of the edges, which are then distributed so that each vertex has all its outgoing edges. This new model also encourages reusing datasets while operating on the same graph. Also, since Facebook's data warehouse is stored in Hive, one of our early requirements was efficient reading from/writing to Hive tables. Since Hive does not support direct querying of its tables, and HCatalog didn't support Facebook's internal Hadoop implementation, we created HiveIO, which provides a Hadoop I/O format style API that can be used to talk to Hive in a MapReduce job. Our Giraph applications use HiveIO to read graphs in both edge and vertex oriented inputs up to four times faster than Hive. We also added multithreading to loading the graph and implemented shared aggregators.



While these tools have had particular impact on our abilities to carve out efficiencies in our data management and analysis, we've created whole other suites of tools that help us manage our load. Two other examples are Presto, an interactive query processing engine, and Morse, which performs real-time data scraping. We will write more about these two technologies in upcoming blog posts.

Today Facebook faces scaling challenges that more and more services will experience as they expand globally. As more people share more content, more frequently, the industry needs to ensure that we're also creating efficient ways to transfer people's content from data centers onto their devices.

Making the delivery of data faster and more efficient is crucial to ensuring access to more people in more diverse regions of the world. And as the number of people who use our services and other like it continues to grow, building more efficient apps to parse and present data is the other major key to delivering affordable Internet to all.

### *Building More Efficient Apps*

There are currently five billion people using mobile phones - only one billion of which are smartphones. As Facebook continues to grow, we're focused on reaching the entire spectrum of mobile users and making it easy and affordable to access the Internet from their phones. Currently, the vast majority of the prices people pay for data plans go directly towards covering the tens of billions of dollars spent each year building the global infrastructure to deliver the Internet. Unless this infrastructure becomes more efficient, the industry cannot sustainably serve everyone.

One of the biggest hurdles to achieving efficiency is the challenge of supporting the enormous diversity of devices and networks in use around the world. We've already begun addressing this with the Facebook for Android app.

Here are some of the specific challenges that Facebook is working to solve:

DIVERSITY OF DEVICES AND OPERATING SYSTEMS

The Android operating system is exceedingly popular and continues to power a wide variety of smartphones around the world. However, there are several different versions of the OS that continue to be in active use every day by hundreds of millions of people around the world.

In order to support these users, Facebook must build for devices with wide ranging capabilities and differences in feature sets. These features may be visual and stylistic changes that affect the aesthetics of an application or improvements in the garbage collection algorithms of the Java Virtual Machine that significantly influence the stability and performance of the application.

Android devices also come in many different physical styles and complex permutations of hardware. Facebook's apps must be tested on these. The number of cores in a CPU, the amount of RAM on the device, the presence of external storage such as an SD card, the size and resolution of the display, the capacity of the battery and the type of cellular radio on the device are just a few of the things that can change not just from one manufacturer to another, but even between different models produced by the same manufacturer. Each one of these characteristics plays an important role in how we build our apps to optimize for stability, performance and efficiency.

GEOGRAPHIC DIVERSITY

Our goal is to deliver the best experience customized for each of the 1.15 billion people using Facebook. This means building the service in around 70 different languages and showing the most relevant content no matter how often the app is used.

NETWORK CONNECTIVITY

One of the main goals behind Internet.org is to provide affordable access to basic services on the Internet. Even for people who have some form of connectivity today, the type and quality of experience available varies wildly. Countries like Korea are on the leading edge of cellular connectivity and already rolling out 4.5G networks whereas countries like India are just starting to roll out 3G and 4G networks widely. There are also vast parts of the world where the best connection

available is an EDGE connection. Affordability of cellular data plans also varies widely—in some countries, people pay a flat rate for a seemingly unlimited amount of data while in the majority of the world, access is purchased as it's used on the go, on a daily basis.

This generates a lot of unpredictability in the quality of a user's data connection at any given moment. And when we combine the need to optimize for operating system, device, and level of network connectivity, optimizing for a high-quality and reliable experience becomes a huge challenge.

To solve for efficient data delivery and consumption—and preservation of battery life—we're taking several approaches.

#### TESTING TO ENSURE CONSISTENT NETWORK CONNECTIVITY

Facebook's engineers can't be everywhere at once, so to test the service across all locations, devices, and operating systems, a system called Air Traffic Control has been implemented to help engineers simulate different network conditions right inside Facebook's offices. Through this infrastructure, employees have the ability to control the connection that their device has to the Internet. Aspects that can be controlled include bandwidth, latency, packet loss, corrupted packets, and packet ordering. This allows engineers to do things like:

- Simulate mobile radio technologies over Wi-Fi. (e.g. 2G, EDGE, 3G, 4G)
- Simulate conditions in various countries (e.g. what the Facebook app feels like to users in India)
- Simulate various levels of network capacity and congestion including peak network hour

This system has helped Facebook to detect issues that affect users on slower networks, and has other issues to be solved, such as dropped connections.

## CONSUMING LESS DATA

Mobile users today have a wide variety of data plans. From multi-gigabyte/month plans popular in the US to pay-per-MB plans popular in developing nations to family shared data plans, the range of experiences with respect to mobile data is wider than ever before. This presents a unique set of challenges for Facebook, given that it is a visually oriented experience with constantly updating content and big photos. Our goal is to provide a good product, but scaled down in a way that allows it to work well in a network constrained and price sensitive environment. Methods of achieving this have included varying image sizes, aggressive prefetching when Wi-Fi is available, reordering stories in low data situations, and long-lived, multi-process caches.

One significant way to reduce data consumption is by optimizing image resolutions and formats. Facebook is the largest image-sharing site on the Internet, and images represent the largest source of data usage on Facebook. As a result, impact achieved in this area can materially reduce the overall data consumption.

When building Facebook Home, we started to measure how many new stories we expected users to see, multiplied by the bytes per photo, and recognized that we shouldn't just silently download this much data. While many apps will continually download data in response to user actions (scrolling through images, for example), this can lead to users being unpleasantly surprised by their data usage—not a great experience to leave people with.

Our first solution was to provide URLs corresponding to multiple versions of the photo associated with the story we were trying to display <[https://www.facebook.com/note.php?note\\_id=76191543919](https://www.facebook.com/note.php?note_id=76191543919)>. Facebook's photo storage system is intelligent enough to only actually store and persist those photos when they're asked for, so we added the ability to download a different resolution photo based on whether a device is on Wi-Fi or a cellular connection. This helps control cellular data usage.

But images aren't the only driver of data usage. The text of a story, its image URLs, names of people in a story and other content all drive data usage, as well as drain battery life. Consequently, the rate at which these items are fetched in the background was also adjusted, in order to minimize data and battery usage.

In order to make these things work, a reliable and scalable image cache had to be developed on Android. We needed to be intelligent about how to evict images from this cache, to avoid re-downloads wherever possible, and not use undue memory to operate quickly. We also decided to make the cache work across multiple processes, so that Facebook for Android and Home could reuse the same images and avoid making resource-intensive network connections too often.

We also needed to solve for the case of someone wanting to view hundreds of stories each day, but never using Wi-Fi and still wanting to use only a small amount of data. In general, we put a lot of effort into not using huge amounts of data no matter what the situation. With Home, a virtual cap was established on the amount of data we're willing to let the app use, regardless of the number of stories, size of photos, or battery state of a device, which lets us protect a device's data usage against interactions that might otherwise use data, but don't seem harmful.

We also want to be sure to design an experience that would degrade gracefully in a low-data situation.

To make image rendering more efficient, we first looked at image sizes and resolutions. In order to preserve image fidelity, the same image size was always sent that was originally uploaded, often 960x960px or even larger. On a desktop, this makes sense, as a wide, high quality image is necessary to fill up much of the screen. But on a phone with a 3" screen with only 320px on the long side, sending such a large image would only increase image-loading times without actually providing additional quality

To account for this, these constraints are now communicated to the server in the request and the server calculates the best size to respond with. Ideally the exact size requested could be sent, but with the number of different screen sizes this is untenable. In order to increase the cache hit rate, there are constraints to allow only certain image sizes, and increasing this cache hit rate results in dramatically faster image loads for the user, as these images do not need to be retrieved from storage.

In addition to optimizing for image size, we also looked at tackling data usage through the widespread adoption of WebP throughout the application. WebP is a new image format developed by Google, and initially released in 2010. This is intended to replace both PNGs and JPEGs, two of the most common and prevalent image formats on the internet, providing substantial reduction in image size at an equivalent image quality. A more efficient image format means that users can download and view images faster, while saving on bandwidth costs.

The general trend is that as photo size and quality increase, WebP's advantage over JPEG diminishes. While WebP achieves a tremendous size savings on 180x540px photos and below, it doesn't deliver as large a savings on larger photos. Because WebP is a new image format, there are multiple hurdles around support as well as efficiency we need to overcome in order to truly reap the benefits of WebP.

Maintaining caching performance is another challenge. Since all users on desktop, and some users on mobile are still fetching JPEG, the cache in Facebook's CDN is more likely today to have a JPEG in its cache rather than a WebP. This incurs a penalty in having to fetch the image from storage. Over the long term, as we convert more to WebP, we expect the penalty to decrease. At this point, most of our images are converted into WebP for the Android application, and our goal is to roll out WebP to other platforms as well. When the images are converted to WebP, this will save over 20% of total network traffic, without loss of quality.



To continue to innovate and improve data usage efficiency we will need to:

- Educate engineers and equip them with necessary tools/infrastructure
- Cache images on device and using SD card where available
- Set a data budget and stick to it

### *Optimizing Battery Life*

Having fresh, real-time data from your friends in the form of new stories, messages, and notifications is core to the Facebook experience. With Home, Facebook was introduced to a new challenge—how to manage power usage of our Android apps in a world where significant actions (panning, data refresh, etc) happen without user input. Working to solve this problem has identified new ways to optimize background fetching and foreground interaction power costs, while creating a testing framework to measure power usage and test for regressions.

Tackling background power usage was an obvious first step in improving the power drain of the experience as a whole. In general, it's critical that while the app is backgrounded, it doesn't drain significant amount of power, so getting to that point was an important precondition for releasing the app, and not something that the team wanted to fix at a later date. While a solution like GCM instead of periodic refreshes seemed like a good idea to start, it turned out to be an inadequate solution for this particular use case. Given the goal of starting with periodic refreshes, one of the first improvements we made was to vary the interval at which new stories are fetched based on whether the device is connected to Wi-Fi or cellular networks at any given time. Wi-Fi refreshes are substantially more power efficient than cellular connections, so it is possible to err towards refreshing data more often without paying as high of a price. This also has important implications for data usage.

The high cost of waking up a radio for a story fetch comes from network inactivity timers on the device, which keeps the radio in high-energy state for fixed period of

time regardless of the size of data packets to be transferred. These timers vary by network operators, so the same device type can have different radio wakeup costs across different network operators. Because of this, we needed to minimize the number of radio wakeups and coalesce as much network traffic as possible, while maintaining the feeling of recency. Additionally, the range of Android devices have dramatically different power draws for similar actions. For instance, the battery drain for a waking up the device and fetching a new story range from 0.02% to 0.1% of the total battery, even among popular ICS+ devices. Consequently, we implemented a way to control how often these devices wake up based on their power profile.

In addition to background power usage, we also optimized for active usage. With Home, Facebook worked with GPU vendors to tune the workload so that the power draw kept the Application Processor in optimal power mode while not compromising the experience on the device. This made it so when the home screen is active, Home ensures that the Application Processor and the GPU stays in optimal power mode.

Even if apps are optimized for background and active use, it can be difficult to measure power optimization; even a single change can cause a regression in power. To measure power across builds, we created a power measurement testing lab, where we measured power across various tests scenarios. Each time a commit was pushed to our codebase, the code was compiled and loaded it onto one of our test devices by the setup. The device was connected to build slave through adb over Wi-Fi. Connecting the device through USB causes the device to not go into a complete power collapse state, thus preventing the catching of potential regressions in the code which would prevent the device from going to sleep.

When a regression was identified by this framework, the offending diff was simply discarded so that our baseline power could be maintained going forward. This allowed us to continue moving fast while avoiding regressions of a difficult-to-observe key metric. As a result, the progress achieved in background fetching optimizations and foreground interactions was preserved while new features could

continue being added to Home. As time goes on, our goal is to make further improvements to power usage.

### *Facebook for Every Phone*

In addition to optimizing our Android apps, we're also building feature phone apps designed to run on low bandwidth with Facebook for Every Phone, a product of the Snaptu team. Snaptu was acquired in 2011 to bring its technology stack to Facebook users who cannot afford smartphones. While the smartphone market is growing, it will be a long time before they replace all feature phones. Even today when smartphones outsell feature-phones, there is a huge installed base of over four billion feature-phones. The forecast is that only in 2015 the world will reach the feature-smartphone break-even point, in terms of installed-base. Even then, the decline of feature-phones may take some time.

The Snaptu technology behind "Facebook for Every Phone" is being used by over 100 million users every month. The system is designed to meet the needs of people who cannot afford an expensive phone or expensive data plan, but still want to be connected to their friends on Facebook. While connecting everyone regardless of device is driven by Facebook's mission, we have also proven that we can build a profitable business from it.

The system was originally designed to optimally serve users who had:

1. A narrow band network pipe (2G network)
2. Phones with limited processing power

The limited processing power on the client required moving the heavy-lifting computation to the server-side where computation resources are abundant. Server efficiency is key to enable the system to scale economically to millions of users. Common narrow band network conditions was another leading consideration in the design of the client-server architecture, and a key design goal was related to the device fragmentation.

It is challenging to solve all the problems that originate from this fragmentation in parallel with solving the application-level problems, so we elected to separate these concerns into two distinct system components—gateway and application server. This breakdown of the problem domain enables different people and teams to focus on a specific and smaller set of problems, and solve them one at a time rather than having to solve them all at once.

The Facebook for Every Phone client-server protocol is designed around 2G network conditions. To work around high-latency low-bandwidth connection between the client and the server, we use a somewhat unusual binary protocol. The protocol attempts to minimize transfers between the client and the server, and maximize reuse of the elements that were sent over to the client through the use of caching.

Facebook for Every Phone has a unique remotely managed persistent cache, in which the server has an index of exactly what files are available on the client. Because the server knows exactly what is in the client cache, the server can send the entire screen down at once, without the client needing to request particular images. This saves us a round trip and data.

Server side cache management also gives us several more advantages; it allows us to change up the caching algorithm on the fly—and lets us manually tune our algorithms on a per-screen basis. Some of these considerations include:

#### TRANSLATIONS

Facebook for Every Phone uses the Facebook translation system, but instead of packing all the possible translations onto the client device, Facebook for Every Phone is loading the right translation for every user during the screen preparation time. This results in a smaller downloadable client and immediate fixes.

#### DEVICE DETECTION

Identifying the correct device allows the system to optimize the feature set it applies for that session, and allows us to compensate for lack of other features. By building more efficient apps—both through strengthening Facebook for Every Phone which caters directly to the feature phone market, and by experimenting

and pushing new boundaries with our most advanced Android apps—we are learning how to bring effective user experiences to everyone across geographies, networks, and devices.

### *Looking Ahead*

In order to bring affordable Internet to more people around the world, we want to make sure our apps are optimized for all devices and networks. To make this happen, we need to continue to push the limits of efficiency in data delivery and usage. Facebook has already invested more than \$1 billion to connect people in the developing world over the past few years, and we plan to do more.

However, this is just a start—and there is much more to be done. In chapters 5 and 6 of this paper, we feature contributions from Qualcomm and Ericsson that explore the need to achieve more efficient network technologies as a way of preparing the telecommunications and technology industries for a future in which billions more people are online in developing countries.

For Qualcomm, technical innovation and industry collaboration are essential to achieving the “1000x Challenge”—a world in which there is sufficient global wireless capacity to meet demand that is 1000 times greater than today. A combination of massive research and development, extensive deployment of new small cells and the provision of significantly more spectrum will provide a solid foundation for achieving this challenge.

It will not be easy to achieve this challenge. But if the telecommunications industry can achieve superior network performance, then this will produce enormous benefits for them—building customer satisfaction and loyalty over the long-term and driving new growth and opportunities. Based on a long history of collaborating across the industry, Ericsson identifies a number of best practices to inform the approach of operators towards infrastructure development including developing cutting-edge products to improve user experience and network efficiency, innovating in network design to reduce bottlenecks to smartphone performance and automating tasks to reduce operational costs.

Beyond the strategies and technical approaches championed by our partners Qualcomm and Ericsson, there are many other approaches to driving down the cost of delivering network coverage that utilize both licensed and unlicensed spectrum bands in networks that include both traditional mobile networks as well as wireless ISP-centric networks that support “nomadic” data-centric use cases. We will share more details on these “non-traditional” mobile networks in a future update.

With the learnings contained in this report as a starting point, our goal is to work with the entire industry to generate fresh ideas and shared insights into the challenges and opportunities of connecting the next five billion people. We look forward to receiving feedback from all those with an interest, and working together to advance the goals of Internet.org. Please visit our website to learn more.

When Facebook began thinking about building its own data centers in 2009, both the company and number of people using the site were growing exponentially. With this growth in mind, a decision was made to build our own facilities from the ground up to ensure long-term sustainability and efficiency.

In April 2011, we turned up our first data center based on our own designs in Prineville, Oregon, driving a 38% increase in energy efficiency and 24% decrease in cost.

All three of our data centers—Prineville; Forest City, NC; and Luleå, Sweden — defy the gold standard of 1.9 PUE with PUEs ranging from 1.04 to 1.09. We've been able to achieve these gains by using 100% outside air for cooling, by taking a “vanity-free” approach to our hardware design, and by making heavy use of automation in our operations.

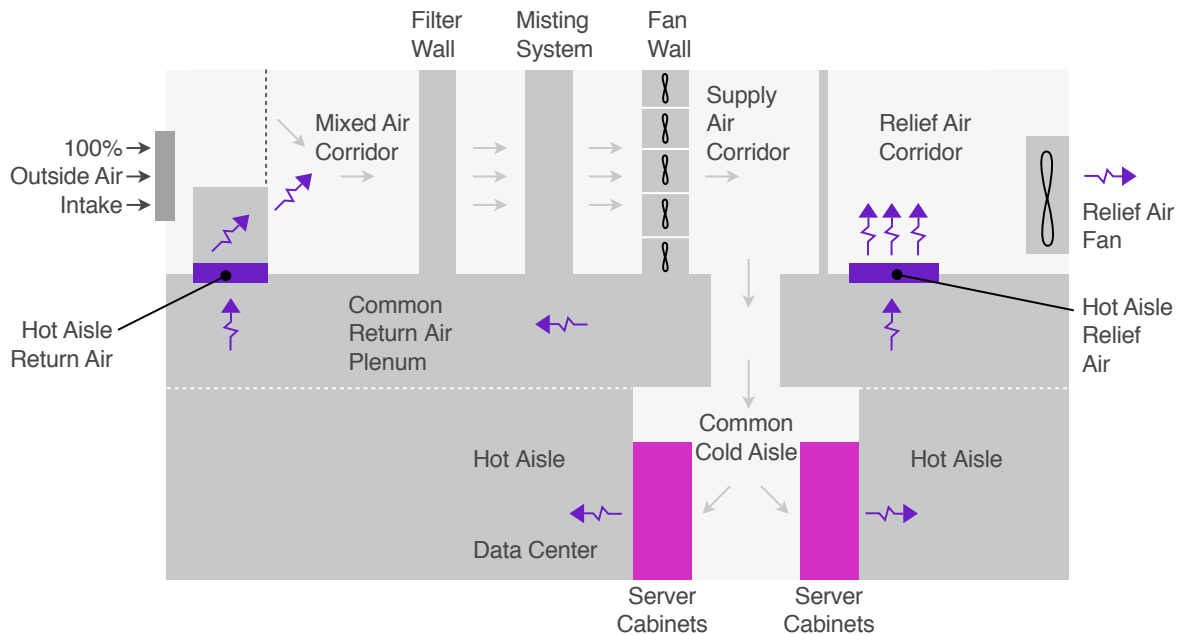
We designed and built our servers, software, power supplies, and data centers in tandem from start to finish to achieve maximum efficiency. The efficiency gains we've achieved have been made possible by three core areas of innovation: cooling, power management, and “vanity free” server design.

OUTDOOR AIR COOLING

Traditional data centers employ chillers or cooling towers to regulate the temperature of their servers. This system can be highly inefficient and detrimental to the environment, so in Facebook data centers we use an evaporative cooling system that brings in outside air and then lowers the temperature of that air by adding humidity.

Facebook uses the outside air as a first stage of cooling, which is also known as outside air economization. Outside air enters the data center, gets filtered and directed down to the servers, and is then either re-circulated or exhausted back outside. This system is used all year round, and allows us to forgo chemical water treatments and dumping cooler tower water.

When the outside air needs extra cooling, Facebook data centers employ either a direct ECH misting system or the use of wetted media. This phase drops the temperature of the air significantly by changing liquid water to water vapor in the direct path of supply air to the data hall. In our Prineville data center, direct evaporative cooling operates 6% a year on average, and will be less than 3% for Luleå because of its cooler climate. For other times of the year, outside air economization is used.





For example, our Luleå facility is cooled by bringing in colder outside air approximately 97% of the year. During low-humidity times of year, the evaporative cooling system will function as a humidifier to meet server humidity requirements. During the colder months, a portion of the excess heat from the servers is used to heat the internal office space.

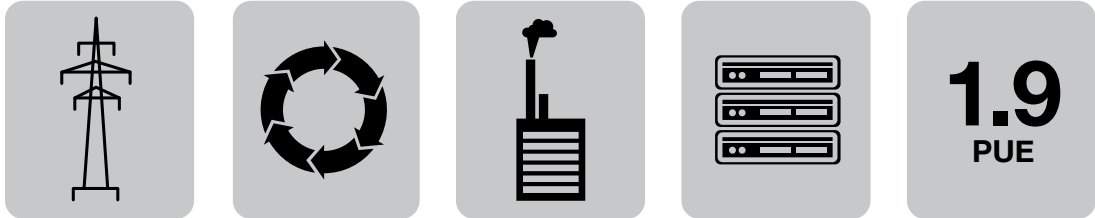
Optimizing for cooling is even built into our servers—they are designed to work in a hotter and more humid environment. They are also taller than the average server, which means we can use fewer, bigger fans to cool them. To save energy, we also slowed the fans down so they now make up 2-4% of the energy consumption in a typical server, compared with the typical 10-20%.

#### POWER MANAGEMENT

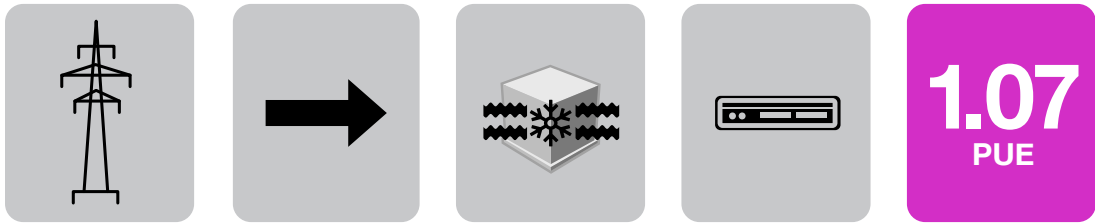
We rethought every part of our power management system to optimize for efficient power consumption, such as cutting out a stage of power transformers, using a higher voltage throughout the facility, and removing everything that doesn't directly benefit efficiency.

All our data centers have an uninterruptible power supply to bridge the gap between a power outage and the moment backup generators kick in. The Luleå data center uses a new, patent-pending system for UPS that reduces electricity usage by up to 12%. And given the robustness of the utility grids in Luleå, we have been able to reduce our number of backup generators by 70%.

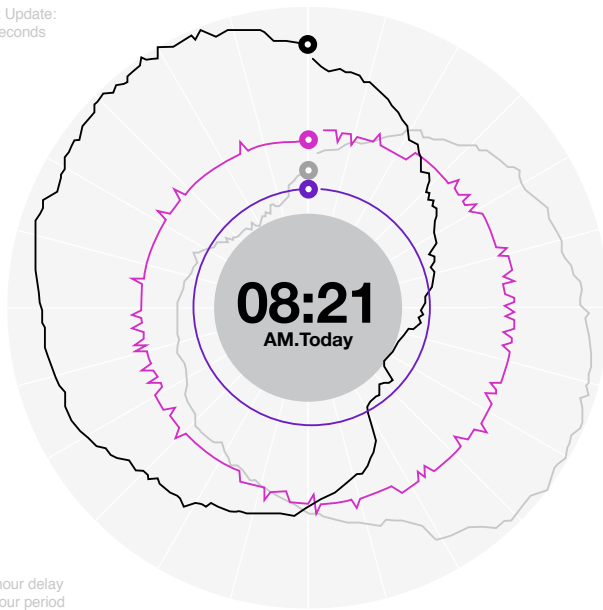
### Industry Standard



### Open Compute Project



Next Update:  
45 seconds



Power Usage  
Effectiveness  
(PUE)

1.07

Water Usage  
Effectiveness  
(WUE)

0.17

Humidity  
(Outdoors)

77%

Temperature  
(Outdoors)

58°

F/14.2°C

VANITY-FREE SERVER DESIGN

The Open Compute Project design philosophy—which we like to call “vanity free”—eliminates anything from the designs that isn’t necessary to the function of that device.



A good example of this philosophy in practice can be found in the removal of the plastic bezel from the front of our first web server designs. What we found was that not only was that bezel unnecessary from a materials standpoint—it was a non-functional piece of plastic on every server that would need to be commissioned and then recycled at end of life—but the bezel was also impeding air flow through the server, meaning the server fans needed to consume more energy to cool the device. Removing these bezels from the front of our servers reduced fan power consumption by 25 Watts per server, when compared with other web server models at the time.

FUTURE WORK

When we built our first data center in Prinville, Facebook made an unprecedented decision to “open source” all the specifications behind that infrastructure, founding the Open Compute Project in the process. Openness is part of Facebook’s culture, and it was our belief that opening the technology would spur advances that we wouldn’t have discovered if we’d kept this technology secret. Other people are now building on and making improvements to our original designs.

By opening the specifications for our servers and datacenters, the Open Compute Project believes that innovation will accelerate, equipment efficiency and quality will increase, and costs will decline. Individuals and organizations across the industry are beginning to build on and make improvements to our original designs, and everyone is benefiting, including companies that manufacture their own servers.

Some of the latest, most impactful OCP projects we’re working on are:

<b>Network</b>	Interconnect	SDN	 Hardware	 Operating System	
<b>Storage</b>	HDFS	Swift	Posix	Open Vault	Cold Storage
<b>Server</b>	Linux	Open Stack	IPMI	2-socket Intel/AMD	Group Hug + ARM SOCs
<b>Rack</b>	Co-lo		Open Rack		
<b>Data Center</b>	Co-lo	Greenfield		Cold Storage	

NETWORK SWITCH

Over the last two years, Facebook has been able to leverage existing open source technologies and work with the Open Compute community to build new software and hardware to create an open source data center. However, we were still connecting our data centers to the network with black box proprietary switches. OCP recently expanded its focus to include networking, collaborating on the development of an OS-agnostic switch that is designed for deployment at scale and will allow consumers to modify or replace the software that runs on them.

A wide variety of organizations — including Big Switch Networks, Broadcom, Cumulus Networks, Facebook, Intel, Netronome, OpenDaylight, the Open Networking Foundation, and VMware — are planning to participate.

An open, disaggregated switch should accelerate innovative development of networking hardware; help software-defined networking evolve; and ultimately provide the freedom that consumers of these technologies need to build flexible, scalable, and efficient infrastructures.

COLD STORAGE

Cold data storage is increasingly in demand as more people share more content that needs to be stored, like old photos that are no longer accessed regularly but

still need to be available. However, there's a lot of progress to be made in developing a system with high capacity at low cost.

The Open Compute specification for cold storage is designed as a bulk load fast archive. The typical use case is a series of sequential writes, but random reads. We've found that a Shingled Magnetic Recording (SMR) HDD with spin-down capability is currently the most suitable and cost-effective technology for cold storage. For this to work, a separate infrastructure dedicated to cold storage needs to be designed and deployed. The Open Compute specifications for this design are here < [http://www.opencompute.org/wp/wp-content/uploads/2013/01/Open\\_Compute\\_Project\\_Cold\\_Storage\\_Specification\\_v0.5.pdf](http://www.opencompute.org/wp/wp-content/uploads/2013/01/Open_Compute_Project_Cold_Storage_Specification_v0.5.pdf)>.

#### DISAGGREGATED RACK

Much of the hardware we build and consume as an industry is highly monolithic —our processors are inextricably linked to our motherboards, which are in turn linked to specific networking technology, and so on. This leads to inefficient system configurations that can't keep up with evolving software and in turn waste energy and material.

To get to a place where hardware can allow for rapid innovations in software, we need to disaggregate some of these components from each other so we can build systems that are customized to fit the workloads they run. This also allows components to be replaced or updated independently of each other, saving money and resources. Our first steps toward this kind of rack disaggregation are:

- Silicon photonics: Intel is contributing designs for its upcoming silicon photonics technology, which will enable 100 Gbps interconnects, which is enough bandwidth to serve multiple processor generations. This technology's low latency allows components that previously needed to be bound to the same motherboard to be spread out within a rack.
- “Group Hug” Facebook is contributing a specification for a new common slot architecture for motherboards that can be used to produce boards that are completely vendor-neutral and will last through multiple processor generations.

- New SOCs: AMD, Applied Micro, Calxeda, and Intel have all announced support for the Group Hug board, and Applied Micro and Intel have already built mechanical demos of their new designs.

The goal of these innovations is to enable data center operators to build systems that better fit the workloads they need to run and to upgrade through multiple generations of processors without having to replace the motherboards or the in-rack networking. This should enable significant gains in utilization and efficiencies across construction and operations.

This chapter explores some of the challenges and potential solutions for delivering more efficient mobile applications designed to meet the unique needs and infrastructure of users in developing countries. This examination focuses on lessons learned during the development of Facebook's apps for the Android mobile platform.

When it comes to mobile app development, Facebook is driven by a core objective—to ensure a consistently high quality experience for all our users. When people share messages, content or updates with their friends, family and loved ones, they expect to have these delivered to them as quickly and efficiently as possible, wherever they are in the world. This is the opportunity of mobile today—to share and consume all the important moment in life today as they actually happen.

However, delivering on this vision is not as straightforward as it seems. Below are just some of the factors that make this an incredibly complex challenge for us.

## *The Challenges*

### ANDROID IS INCREDIBLY DIVERSE

The Android operating system is incredibly popular and continues to power smartphones across the world. However, there are still multiple versions of the OS that continue to be in active use every day.

What this means is that Facebook users are using many different devices whose operating systems have wide ranging capabilities and differences in feature sets. These features may be visual and stylistic changes that affect the aesthetics of an application or they may be improvements in the garbage collection algorithms of the Java Virtual Machine that significantly influence the stability and performance of the application.

FACEBOOK USERS ARE EVERYWHERE

As the largest social network on the planet, Facebook connects more than 1.15 billion people as of June 2013. To deliver a truly global experience for the Facebook community, the service must be capable of being reliably accessed from around the world at all times. Delivering relevant content is also important; to that end, Facebook is available in approximately 70 languages today.

PHONES COME IN ALL SHAPES AND SIZES

One of the reason Android smartphones are so popular is that they come in an incredibly wide variety of shapes and sizes. Phone manufacturers are constantly trying new ways to differentiate and innovate in an incredibly competitive market. Although this provides greater choice for users, this also places greater demands on app developers, requiring them to test their apps on a complex and constantly shifting permutation of hardware. The number of cores in a CPU, the amount of RAM on the device, the presence of external storage such as an SD card, the size and resolution of the display, the capacity of the battery and the type of cellular radio on the device are just some of the things that can change not just from one manufacturer to another, but even between different models produced by the same manufacturer. Needless to say, each one of those on device resources plays an important role in the stability, performance and efficiency of the Facebook app.

NETWORK CONNECTIVITY IS UNEVENLY DISTRIBUTED

One of the main goals behind Internet.org is to provide affordable access to basic services on the Internet. Even for users who have some form of connectivity today, the type and quality of connectivity available varies wildly. Countries like Korea are on the bleeding edge of cellular connectivity and already rolling out 4.5G networks whereas countries like the India are just starting to roll out 3G and 4G networks widely and then there are vast parts of the world where the best connection you can get is an EDGE connection. Affordability of cellular data plans also varies widely. In some countries, people pay a flat rate for seemingly unlimited amount of data while in the vast majority of the world, access is purchased on the go, on a daily basis. As a result, the quality of a user's data connection can be unpredictable at any given moment—and app developers have a special responsibility to ensure they are efficient and judicious in how they use someone's data budget.



On their own, any one of these factors can pose significant challenges to delivering on our mission of providing a high quality experience. When combined, they become an unprecedented challenge.

The following section discusses some of the technical solutions that Facebook has explored to overcome the challenges described above—in particular, ways that our team has worked to use less data and minimize battery drain on Android.

## *The Solutions*

### HANDLING NETWORK CONNECTIVITY

One of the ways Facebook ensures that applications work well under realistic network conditions is by extensively testing them in these conditions. However, it is clearly impractical to dispatch engineers all over the world every time a new version of the app is developed, and to test their work directly in the local environment.

Instead, Facebook has implemented a system known as ‘Air Traffic Control’ internally that helps engineers simulate different network conditions inside Facebook’s offices. With this infrastructure, employees have the ability to control all aspects of the connection that their device has to the internet, including bandwidth, latency, packet loss, corrupted packets, and packet ordering. This allows employees to do things like:

- Simulate mobile radio technologies over Wi-Fi. (e.g. 2G, EDGE, 3G, 4G)
- Simulate conditions in various countries. (e.g. what the Facebook app feels like to users in India)
- Simulate problematic service. (e.g. slow DNS, intermittent connection, blocked ports, firewalls)
- Simulate various levels of network capacity and congestion including peak network hour

Air Traffic Control has been invaluable in detecting issues that affect users on slower networks, and has allowed the graceful handling of errors such as frequently dropped connections.

## *Consuming Less Data*

Facebook has achieved significant expertise in reducing data consumption by optimizing image resolutions and formats.

As well as image optimization, we've also pursued two techniques to reduce the data burden of its apps— offloading large data transactions to non-cellular connections (such as Wi-Fi) and locally caching data on devices in advance.

## *Images*

Facebook is the largest image sharing site on the internet. Every day, over 350 million photos are uploaded to Facebook on average. As a result, images represent the largest source of data usage on Facebook, and impact achieved in this area can materially reduce the overall data consumption.

### RESOLUTION

The first thing we did when tackling images was to look at image sizes and resolutions.

In order to preserve image fidelity, we always sent the same image size that was originally uploaded, often 960x960px or even larger. On a desktop, this makes sense, as a wide, high quality image is necessary to fill up much of the screen. On a phone with a 3" screen with only 320px on the long side, sending such a large image would only increase image loading times without actually providing additional quality. Furthermore, many images are not loaded to fill out the entire screen, but only a smaller area they are constrained to.

As a result, these constraints are now communicated to the server in the request and the server calculates the best size to respond with. But why not just send

the exact size requested, why allow the server the flexibility? Unfortunately, the number of different screen sizes is staggering. In order to increase the cache hit rate, there are constraints to allow only certain image sizes. Increasing this cache hit rate results in dramatically faster image loads for the user, as we do not need to retrieve these images from storage.

#### BACKGROUND ON WEBP

We also looked at tackling data usage through the widespread adoption of WebP throughout the application.

WebP is a new image format developed by Google, and initially released in 2010. This is intended to replace both PNGs and JPEGs, two of the most common and prevalent image formats on the internet, providing substantial reduction in image size at an equivalent image quality. A more efficient image format means that users can download and view images faster, while saving on bandwidth costs.

#### IMAGE STUDY

We first wanted to quantify the savings for serving WebP over JPEG for Facebook. The general trend is that as the photo size and quality increase, WebP's advantage over JPEG diminishes. While WebP achieves a tremendous size savings on 180x540px photos and below, but are unable to deliver as large a savings on larger photos. As a result, we saw that the overall savings were not as high as 25-30%. We believe our differences in savings compared to the Google study are due primarily to a different of measuring quality.

In order to measure image quality, doing a pixel-by-pixel comparison does not closely correlate with perceptual quality, or whether people can meaningfully distinguish two images. As a result, multiple metrics were developed, including Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM), and Multi-Scale Structural Similarity (MS-SSIM).

At Facebook, we currently use MS-SSIM to measure image quality, because this most closely quantifies users' perception of image quality. In comparison to SSIM, MS-SSIM measures the image at multiple scales, which will measure not only

local differences in the image but more holistic differences across the image as well. While MS-SSIM and SSIM are heavily correlated, the WebP format seems to perform disproportionately poorer for MS-SSIM than for SSIM. To achieve the same MS-SSIM score, WebP has to have a higher SSIM on average than JPEG. We believe this accounted primarily for the discrepancy in the results.

#### CONVERTING TO WEBP AT SCALE

WebP is a new image format, especially compared to JPEG (25 years old) and PNG (18 years). As a result, there are multiple hurdles around support as well as efficiency we need to overcome in order to truly reap the benefits of WebP. Many browsers do not support WebP and are unable to display it. On desktop, currently only Chrome and Opera support WebP. Even so, this is insufficient, as we have to also consider users will want to download these images and use them elsewhere in programs that may not yet support WebP. As a result, it is important for us to understand when these images will be used, as well as how they are used. On Android, only Ice Cream Sandwich and above (Android 4.0+) support WebP. However, the users most constrained by bandwidth and limited data plans tend to more typically be on Gingerbread (Android 2.3) phones. As a result, the users who need it the most are typically unable to benefit from WebP.

We are still exploring options to address this, including building WebP support for Gingerbread. We send WebP to these devices over the wire, thus saving on the data bill for the user, as well as being able to send it faster. On the client side, our driver transcodes this into JPEG before saving it to the image cache. From this point onwards, the device sees the JPEG image it is able to process, but we were able to reap the savings from it being sent over the wire as a WebP.

Getting this to perform quickly and invisibly to the user is a tremendous challenge, as Gingerbread phones have some of the slowest hardware among Android devices today. Since decoding the WebP is 2/3 of the time in this flow, we are currently experimenting with simply decoding to a byte array, and using that to display the initial image. This would allow us to asynchronously write to the cache, and delay that until after the user has seen the image.

Maintaining caching performance is another challenge. Since all users on desktop, and some users on mobile are still fetching JPEG, the cache in our CDN is more likely today to have a JPEG in its cache rather than a WebP. This incurs a penalty in having to fetch the image from storage. Over the long term, as we have converted more to WebP, we expect the penalty to decrease.

We transcode all JPGs to WebP on the fly, creating load on the server. In order for WebP to achieve the substantial savings in bandwidth, we must spend a tremendous amount of resources to decode the JPEG, resize, and then encode a WebP. Compared to JPEG, encoding WebP requires 7x the time. Factoring in other parts of the image pipeline, such as decoding and resizing, WebP takes 5x as long as generating the same JPEG.

At this point, most of our images are converted into WebP for the Android application, and we aim to roll out WebP to other platforms as well. When the images are converted to WebP, we will have saved over 20% of total network traffic, without loss of quality.

### *Offloading to Wi-Fi*

While many apps will continually download data in response to user actions (scrolling through images, for example), this can lead to users being unpleasantly surprised by their data usage—not a great experience to leave people with.

One method Facebook has used to avoid surprises is applied when downloading new News Feed stories to display. When this occurs, the app also downloads URLs corresponding to multiple versions of the photo associated with the story. Facebook's photo storage system is intelligent enough to only actually store and persist those photos when they're asked for, allowing it to add the ability to download a different resolution photo based on whether a device is on Wi-Fi or a cellular connection. This helps control cellular data usage.

In addition, our apps watch for transitions to a Wi-Fi connection. When one is spotted, Facebook begins aggressively prefetching and caching images. This means that a device builds up an inventory of photos that it can rely on when data is no

longer plentiful. Images aren't the only source of data usage, however—the text of a story, its image urls, names of people in a story and other assets all cost data, as well as battery life. Consequently, the rate at which these are fetched in the background is adjusted, in order to conserve these smaller sources of usage. These mechanisms are currently used in Home and are expected to be brought to other Facebook applications soon.

In order to make these things work, Facebook needed to develop and use a reliable and scalable image cache on Android, including the ability to intelligently evict images from this cache, avoid re-downloads wherever possible, and avoid undue memory use to operate quickly. We also decided to make the cache work across multiple processes, so that Facebook for Android and Home could reuse the same images and avoid going to network too often. By aggressively prefetching stories and images on Wi-Fi and caching them on your phone, a user may actually download very little in the time they are on a cellular connection.

We have also added the ability to offload this cache to the phone's SD card. Using the SD card frees up precious space on a phone's internal storage, which is usually pretty low on many devices that are being sold in emerging markets.

Sometimes all of these measures together still aren't enough to download great images for all the stories you want to see. What happens if you want to use a really small amount of data, but view hundreds of stories each day without touching Wi-Fi? To support this use case, we created a virtual cap on the amount of data we're willing to let Home use, regardless of the number of stories, size of photos, or battery state of a device. This allows us to protect a device's data usage against interactions that might otherwise use data, but don't seem harmful. We've chosen an initial data cap for certain devices and markets for Home, but a different usage level can be selected under Settings -> Data Use and Image Quality (defaults to Medium).

Given our goal to respect data usage constraints, we also needed to design an experience that would degrade gracefully in a low-data situation. We decided that when a device is at the data cap, we'll just download and show a couple of recent

stories. After those stories, we'll show older stories that we have images for, but that haven't been seen, then finally older stories with images that have already been seen on the device. Over time, as more data becomes available, these newer stories will get images and move to the front of cover feed automatically. Consequently, stories aren't ever lost—they're just postponed until we have enough data to show them properly.

These are both capabilities that we are testing with the Home app and soon expect to bring to the main Facebook app as well.

#### BATTERY LIFE

Every mobile user dreads that moment when it's not quite the end of the day and their phone is almost out of power. Our goal is to avoid this happening to Facebook users. The challenge is how to balance having fresh, real-time connections with friends via stories, photos, messages and notifications without draining the battery—especially since a number of the features used to provide a fast and responsive user experience also require us to anticipate a user's needs via background processing.

Once again, with both Home and the main Facebook app, we've worked to optimize background fetching, foreground interaction power costs and creating a testing framework to measure and track battery usage. Much of this work has been done in close collaboration with our partners at Qualcomm, whose tools, engineers and expertise have proven invaluable.

Reducing background power usage was an obvious first step in improving the power consumption of Facebook's apps. In general, it's critical that devices that aren't doing anything don't drain significant amounts of power. One of the first improvements we achieved was to vary the interval at which new stories are fetched based on whether the device is connected to Wi-Fi or cellular networks at any given time. Wi-Fi refreshes are substantially more power efficient than cellular connections, so it is possible to err towards refreshing data more often without paying as high of a price. This also has the added benefit of reducing the impact on people's data plans and bills.

The high cost of waking a phone's radio to fetch data comes from network inactivity timers on the device. These timers keep the radio in high energy state for fixed period of time regardless of the size of data packets to be transferred. Also, these timers vary by network operators. Thus, the same phone can have very different impact on battery performance across different network operators. Consequently, it is critical to wake the radio as seldom as possible and send as much network traffic each time as possible, while maintaining the feeling of freshness and recency in the UI. One way we've tackled this is by pre-fetching multiple images at a time that your friends have posted instead of waking up the radio separately for each image fetch.

Another problem with optimizing power draw on Android is that many of the devices have dramatically different power drain for the same event. For instance, the battery drain for a waking up the device and fetching a new story range from 0.02% to 0.1% of the total battery, even between popular ICS+ devices. Consequently, we implemented a way to control how often these devices wake up based on their power profile.

However, background power usage isn't the only thing we optimize for. We find that people engaged with Facebook so much that we wanted to be sure that active usage was battery-conscious as well. While the home screen in Home is active, we ensure that that the GPU stays in optimal power state and does not cause the silicon to run in 'Turbo' or high power mode. Facebook worked with GPU vendors to tune the workload such that the power draw remains optimal while not compromising the experience on the device.

As another example, Chat Heads is a new feature that is displayed to the user while other apps are foregrounded. Instead of GPU composing the Chat Heads and requiring an extra pass (which is costly from power perspective), the composition is done by the back-end hardware which is more power efficient. A number of other tweaks were made to the workload so that the composition of all layers was done using power optimal hardware. This has helped us to deliver the right experiences without compromising on battery life.



Unfortunately, the tricky part of optimizing power is that the effects cannot always be easily detected. If the power draw for an application is highly optimized but a bugfix badly impacts this, we may not be capable of detecting this initially. By the time the problem is noticed, it may be difficult to figure out its cause. Did we add an extra hardware layer in the UI? Did we stop backing off on fetching errors and consequently fetched too often in bad network connections? After living through a few surprising regressions, we realized that we needed a solution that works well with Facebook's culture of moving fast, but also ensures the best power experience is delivered to the Facebook community.

To measure power across builds, a power measurement testing lab was created at Facebook where power can be measured across different test scenarios. Each time a commit was pushed to our codebase, we compiled the code and automatically loaded it onto one of our test devices. We then connected the device to a build slave and communicated with the device over Wifi through adb (connecting a USB to the device causes the device to not go into a complete power collapse state, and so it prevented us from catching potential regressions in the code which would prevent the device from going to sleep). The device was connected to an external power source through a power measurement board that our Hardware team quickly hacked up for us. The power measurement board was connected to the build slave through USB through which we collected the power measurement. We also setup a dedicated Wi-Fi Access Point so that the test devices connected for power measurement would not be disturbed by continuous stream of broadcast packets that might be getting sent by other devices. This also gave us a control environment which would not be influenced by devices connecting and disconnecting through the day. This ensured that the idle current while the device was connected to the Wi-Fi networks was as close to 0 as possible.

When a regressions was identified by this framework, we simply backed out the offending diff so that we could maintain our baseline power going forward. This allowed us to continue moving fast while avoiding regressions of a difficult-to-observe key metric. As a result, the progress achieved in background fetching optimizations and foreground interactions was preserved while new features could continue being added to Home. As time goes on, we plan to make even more

improvements to power usage so that users can benefit from new features without having to worry about battery life.

### *Looking Ahead*

This chapter has described a number of efforts and improvements we've already made to improve the data and battery efficiency of our apps. We still have many more ideas for ways to make these better. Technologies like Bluetooth, Wi-Fi Direct, LTE Direct and other peer-to-peer mesh networking technologies hold a lot of promise in creating new and more efficient models for data sharing that don't compromise the quality of user experience. Partners like Qualcomm continue to innovate aggressively on enabling computing on extremely low power—helping your battery to last even longer.

At Facebook, we have a saying that our journey as a company is 1% done. That statement could not be truer when it comes to making sure we can provide the highest quality access to your online services in the most efficient way. We've done a lot to make our apps more efficient, but there's a lot more to do.

In addition to optimizing our Android apps, we're also building native apps for feature phones designed to run on low bandwidth with Facebook for Every Phone, a product of the Snaptu team. Facebook for Every Phone is being used by over 100 million users every month and is designed to meet the needs of people who cannot afford an expensive phone or expensive data plan, but still want to be connected to their friends on Facebook.

Snaptu was originally built in 2007 by an Israeli company with the same name. Facebook acquired the company in 2011 and re-purposed its technology stack to serve Facebook users.

This chapter shares some of the lessons we've learned over the last couple of years while developing Facebook for Every Phone, and that today are helping us deliver more efficient mobile experiences for more people in developing countries.

### *Why Feature Phones Matter*

Snaptu was built to serve people who cannot afford smartphones. Although smartphones will ultimately come to replace feature-phones, this transition has not been quick. Today there is a huge installed base of over 4 billion feature-phones, and the forecast is that the world will only reach the feature-smartphone break-

even point (in terms of installed-base) in 2015. Even after the break-even point, the decline of feature-phones may take some time.

Facebook wants to connect all people around the world regardless of the type of phone they have, smart or feature. This ambition stems from our mission, but we've proven that we can also build a successful business from it - Facebook for Every Phone became a profitable service despite the doubts regarding the monetization prospects of its demographics.

### *Design Goals and Philosophy*

This section provides the principles and the high-level design goals of the Facebook for Every Phone system.

### *Size Matters*

The system was originally designed to optimally serve users who had:

1. A narrow band network pipe (2G network).
2. Phones with limited processing power (min requirement 500kb of RAM).

The limited processing power on feature phones required that we utilize server-side acceleration, moving the heavy-lifting computation to Facebook servers where computation resources are abundant. Server efficiency is key to enable the system to scale economically to millions of users.

The narrow band network required bandwidth to be a leading consideration in the design of the client-server protocol.

### *Device Fragmentation*

A key design goal was related to the device fragmentation. The following Cartesian product explodes into thousands:

**{Screen resolutions} × {Color depths} × {Portrait | Landscape} ×  
{Input: T9 / Qwerty / Touch /... } × {Vendors} × {Generation}**

It is very hard to solve all the problems that originate from this fragmentation in parallel with solving the application-level problems. We decided to separate these concerns into two distinct system components:

- Gateway—this component is addressing the device fragmentation problem. The Gateway creates an abstraction of a “canonical device” on top of which the application logic is being built.
- Application Server—this component hosts the application-level logic, which is agnostic to a specific device.

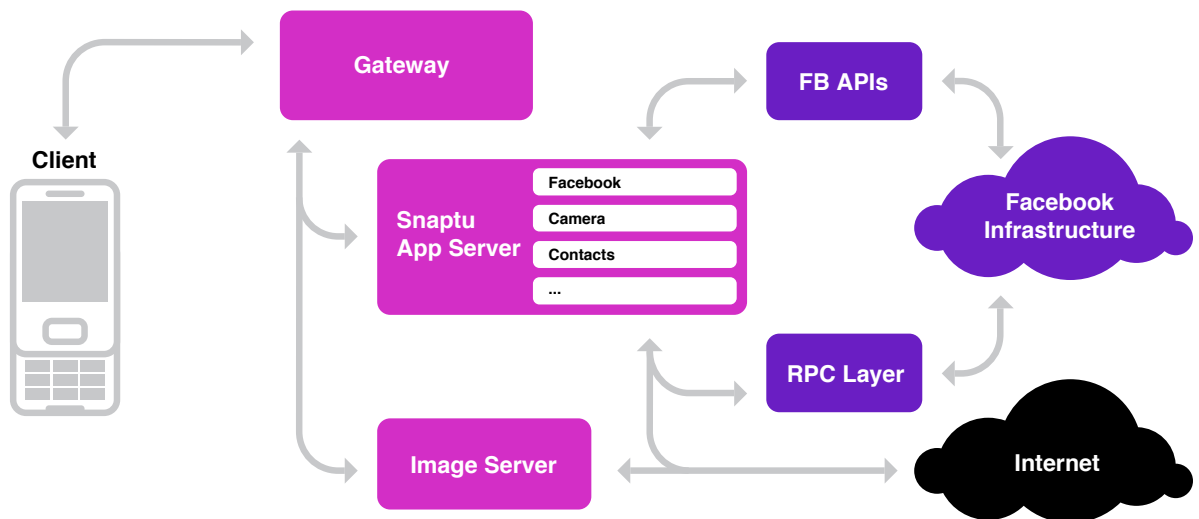
This breakdown of the problem domain enables different teams within Facebook to focus on a specific and smaller set of problems, and solve them one at a time rather than having to solve them all at once.

Another principle we heavily used to address the fragmentation problem was to move decisions to the server-side. Testing our software on all device combinations was unmanageable to us. However, we wanted to retain the option to fix problems in our client software, even after we ship it. To do this, we moved all the decisions we sensibly could to the server-side. This allowed us to fix problems in long-released software by addressing them on the server-side, without releasing new clients or requiring users to upgrade. Problems reported by users were quickly fixed within hours or days of being reported.

## Architecture

The following diagram summarizes how the Snaptu system is being deployed at Facebook:

### Snaptu @ Facebook



As mentioned earlier, the Gateway is the component that is responsible for device adaptation and canonicalization. The App Server contains the application-level logic. The image server is a service that transcodes images to a format the Client can render. The App Server connects to FB back-end services and to the Internet.

### Client-server protocol

The Facebook for Every Phone client-server protocol is designed around the bottleneck of 2G networks. To work around high-latency low-bandwidth connection between the client and the server, a somewhat unusual binary protocol. The protocol attempts to minimize transfers between the client and the server, and maximize reuse of the elements that were sent over to the client through the use of caching.

Images are processed on the server side and being resized to the exact width and height needed by each client. The image quality may be somewhat reduced to

increase compression. Being cut to size and ready to use, the client can render the images without having to resize them.

Screens are sent to the client in broken down in parts (pages). The client's UI becomes responsive upon receiving the first part. This saves the users the need to wait for long screens to load in their entirety. Images are loaded by the client only when they are needed (as the user scrolls). However, the server-side services eagerly fetch the images and resize them to the client's needs. This allows for rapid calling of images when the client needs to render them.

### *Screen Diffs*

In the majority of cases, screens do not change very much. This allows for an optimization to send only changes/updates to screens that were sent earlier. We call this optimization “screen diffs”, that is, after establishing the baseline, we try to send to the client only the differences. The server mirrors the client state and keeps track of the screens that were sent to the client throughout a session. In the frequent case a screen is requested again, a diff is sent down to the client rather than the full content of the screen. When the client flushes a screen from its cache, the server is being notified to also eliminate its own copy.

### *Persistent Caching*

Facebook for Every Phone has a unique remotely managed persistent cache, in which the server has an index of exactly what files are available on the client. Because the server knows exactly what is in the client cache, the server can send the entire screen down at once, without the client needing to request particular images. This saves us a round trip, and data.

Server side cache management also gives us several more advantages; it allows us to change up the caching algorithm on the fly—and lets us manually tune our algorithms on a per-screen basis. Because we have only a limited amount of screens, manually tuning to common usage patterns can bring our eviction policy closer to a clairvoyant one.

## *Font Rendering*

We debated whether to use the device's native fonts or to use our own. There are pros and cons to both approaches, but for us the device fragmentation consideration was key. If we were to go with the device's fonts, we would have a greater element of surprise when we run the app on a device. Further, some devices were lacking the language support we needed. We decided to build a font rendering engine that depends only on the fonts we provide, that is, we went device-independent. The leading benefit for us was the ability to test our app using an emulator and know that what we see is what we're really going to get on the client.

The font engine we've built is a client-server implementation. The server takes TrueType font files, rasterizes the characters that the client needs, and sends them to the client. The client uses the rasterized characters and renders them. This approach helped us remove dependency on the device and achieve a PDF-like fidelity with our screen rendering.

## *Translations*

Translations are important to us since we believe people should be able to use Facebook in their native language. Facebook for Every Phone uses the Facebook translation system. However, instead of packing all the possible translations onto the client device, the app is loading the right translation for every user during the screen preparation time. This results in a smaller downloadable client. This allows for a convenient operational flexibility—translations could be fixed at any time and they take effect immediately for the affected users.

## *Device Detection*

Identifying the correct device has positive impact on the user experience. It allows the system to optimize the feature set it applies for that session, and allows to compensate for lack of other features and can also be used to serve the correct download package to the device, so it can have the correct security signature.



We use a host of methods to detect the exact type of the device being used: Maintaining a list of known devices and their variants of the User-Agent HTTP header, for example, has proven to produce good results. Augmenting the results of the User-Agent header with information from the publicly available WURFL (Wireless Universal Resource File), and run time tests like size of screen, allows us to improve these results in an ongoing process.

Qualcomm is the world's largest licensor of wireless technology and the world's largest manufacturer of chips for wireless devices. Our chips support licensed and unlicensed technologies and as many frequency bands as possible. We strive to develop new technologies, add them into our chips, and support every new band, as quickly as possible.

Our goal is to ensure that the wireless industry is able to meet the "1000x Challenge"—to expand wireless capacity by 1000 times. If usage doubles for ten years, it will be 1000 times today's. Qualcomm, the industry and policymakers must work together on many fronts, in parallel, to meet the 1000x Challenge. The combination of massive research and development, extensive deployment of new licensed small cells, and far more spectrum provides a good path to meet the 1000x Challenge.

First, we are developing new technologies to help meet the 1000x Challenge such as: Carrier Aggregation and Supplemental Downlink, for more capacity and faster data speeds; LTE-Broadcast to deliver content where many people want to see the same thing; LTE-Direct to enable communications when cell networks go down; 802.11ac and ad for faster unlicensed services; DSRC, to enable cars to communicate to avoid collisions; and, broadband for airplanes.

Second, to create 1000x more capacity, cell base stations need to be much closer to devices. Licensed small cells, integrated into wireless networks, will help meet the 1000x Challenge.

Third, meeting the 1000x Challenge will also require far more spectrum. The industry needs more clear, exclusive use licensed spectrum. Clearing new bands by a date certain and auctioning them for exclusive use is the industry's top priority. For unlicensed, wide contiguous bands, adjacent to an existing unlicensed band, are ideal.

Other government bands are equally important. We're also focused on other government bands that are not used nationwide, 24/7, but will not become clear in a reasonable time. One example is 3.5 GHz, which is allocated to the government in the US. This band would be ideal for licensed small cells. Qualcomm and others have proposed Authorized Shared Access (ASA) to enable use of a band by an operator when and where it's not used by the government. A database would ensure that government operations are fully protected from interference, and the operator can provide a predictable quality of service when it can use the spectrum. ASA can provide access to bands that would otherwise be unavailable for many years, without requiring any new technology or devices.

We're working very constructively with policymakers on this initiative, which is another important aspect of meeting the 1000x Challenge.

### *Meeting the 1000x Challenge*

Qualcomm's chips support licensed technologies, 2G, 3G, and 4G; unlicensed Wi-Fi, Bluetooth, and NFC; and, GPS. Our chips support as many frequency bands as possible because there are now approximately 40 bands worldwide for LTE alone. We strive to develop new technologies, add them into our chips, and support every new band, as quickly as possible. We work with virtually every wireless carrier and manufacturer in the world. Qualcomm constantly innovates and works with our many partners to deploy our innovations swiftly.

When we think about a new band, we always ask what technology is best suited technically for it, and what policies will enable the industry to start using it rapidly and broadly.

Qualcomm's goal is to ensure that the industry meets what we call the "1000x Challenge"— to expand wireless capacity by 1000 times. Wireless data usage is doubling each year, and if that trend continues, in ten years, the usage will be 1000 times today's. Qualcomm, the industry, and policymakers must work together on many fronts, in parallel, to meet the 1000x Challenge. The combination of massive research and development, extensive deployment of new small cells, and allocation of far more spectrum provides a good path to meet the 1000x Challenge.

Starting with research and development, Qualcomm and our many partners are working as quickly as possible on a variety of research and development initiatives to help meet the 1000x Challenge. These initiatives include:

- Carrier Aggregation and Supplemental Downlink to bond together separate bands for more capacity and faster data speeds for consumers;
- LTE-Broadcast for multi-casting of video and data in places where many people want to see the same content;
- LTE-Direct to allow first responders and others to communicate device-to-device even if the cell network is down;
- 802.11ac and ad for faster Wi-Fi and other unlicensed applications;
- DSRC, which enables cars to communicate with one another to avoid collisions; and, a next-generation system to provide broadband for airplane passengers.

In addition, creating 1000x more capacity will require locating cellular base stations much closer to devices by integrating licensed small cells into cell networks. A small cell has the connectivity of a base station, but at much lower power. You can put it indoors, where so much wireless traffic originates. Software will inte-

grate it into a wireless network to create a hetnet—a heterogeneous network with cells of different sizes.

Finally, meeting the 1000x Challenge will require more spectrum—far more spectrum. We need more clear, exclusive use licensed spectrum, such as the 600 MHz spectrum from the voluntary incentive auction. Clearing new bands by a date certain in a reasonable time, and auctioning them for exclusive use, is the industry's top priority. For unlicensed, wide contiguous bands, being adjacent to an existing unlicensed band, is ideal.

But, there is another category of spectrum that is quite important: other government bands that are not used nationwide, on a 24/7 basis, but will not become clear in a reasonable time, such as 3.5 GHz in the US and elsewhere and 2.3 GHz in Europe. 3.5 GHz would be ideal for licensed small cells, operating at low power and minimizing any impact on government operations. A small cell can operate at 3.5 GHz because its signal need not travel far, but it requires licensed spectrum to avoid interference.

Qualcomm and others have proposed what we call Authorized Shared Access (or ASA) to enable commercial use of a band such as 3.5 GHz when and where it is not used by the government. ASA is binary—either an operator or the government would use the spectrum at any given time and location. A database would ensure that government operations are fully protected from interference and when the operator uses the spectrum, it can provide a predictable quality of service. ASA can provide access to bands that would otherwise be unavailable for many years, without requiring any new technology for devices or networks.

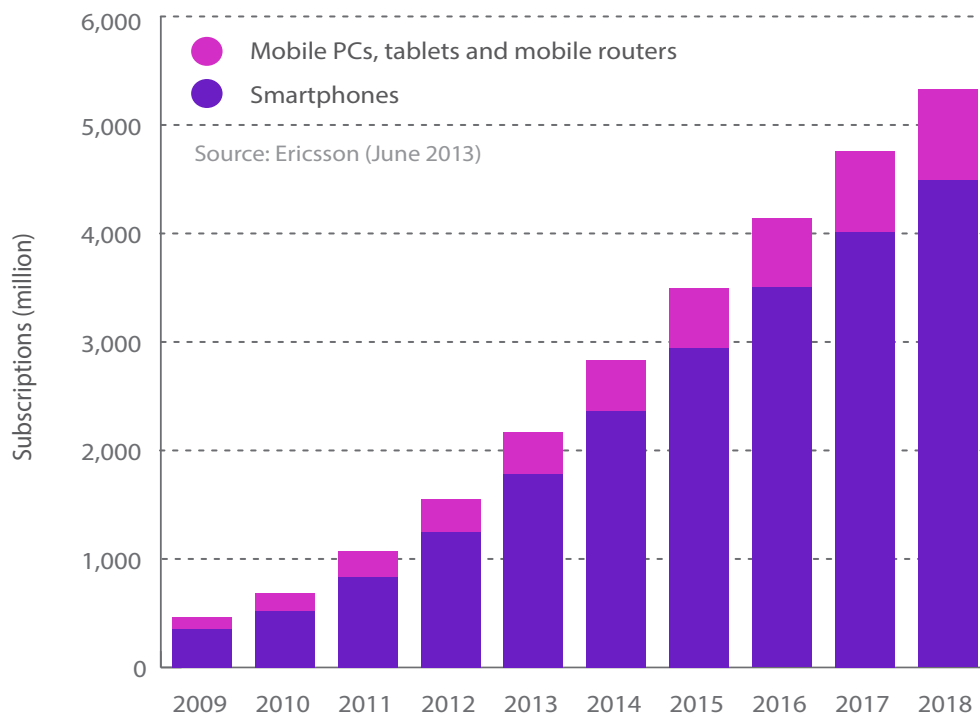
ASA can unlock spectrum bands that would otherwise be unavailable for many years. Doing so and ensuring that small cells can be deployed on such spectrum is vital to meeting the 1000x challenge.

## *Understanding and Meeting Customer Expectations Will Reward Operators with Superior Mobile Broadband Networks*

### SMART MOBILE DEVICES DRIVE NEW USER BEHAVIORS AND EXPECTATIONS

There is no doubt that smartphones and smart mobile devices such as tablets are rapidly becoming the main driving force for mobile broadband growth. Globally, the number of smart mobile devices in use has grown there is no doubt that smartphones and smart mobile exponentially, with some 1.2 billion users estimated by devices such as tablets are rapidly becoming the main the end of 2012. unabated growth is foreseen in the next driving force for mobile broadband growth. Globally, five years, bringing the user number to over 3.7 billion the number of smart mobile devices in use has grown by 2017 1. the bulk of that growth has been—and will exponentially, with some 1.2 billion users estimated by continue to be—fueled by smartphones, as shown in Figure 1.

Figure 1. Smartphone, PC, mobile routers and tablet subscriptions with cellular connection. 2009–2018



The global momentum for smartphone uptake is very strong across all regions, with smartphones already making up around 40 percent of all mobile phones sold in the first half of 2012. Despite this fast-growing the global momentum for smartphone uptake is very pace, the growth potential for smartphones remains strong across all regions, with smartphones already enormous, with only 15 percent of the world's mobile making up around 40 percent of all mobile phones phone subscribers currently using these devices. sold in the first half of 2012. Despite this fast-growing Ericsson estimates that the proliferation of smartphones, pace, the growth potential for smartphones remains mobile pcs and tablets will contribute to an expected enormous, with only 15 percent of the world's mobile 15-fold growth in global mobile data traffic in the next five years, mainly driven by video.<sup>1</sup>

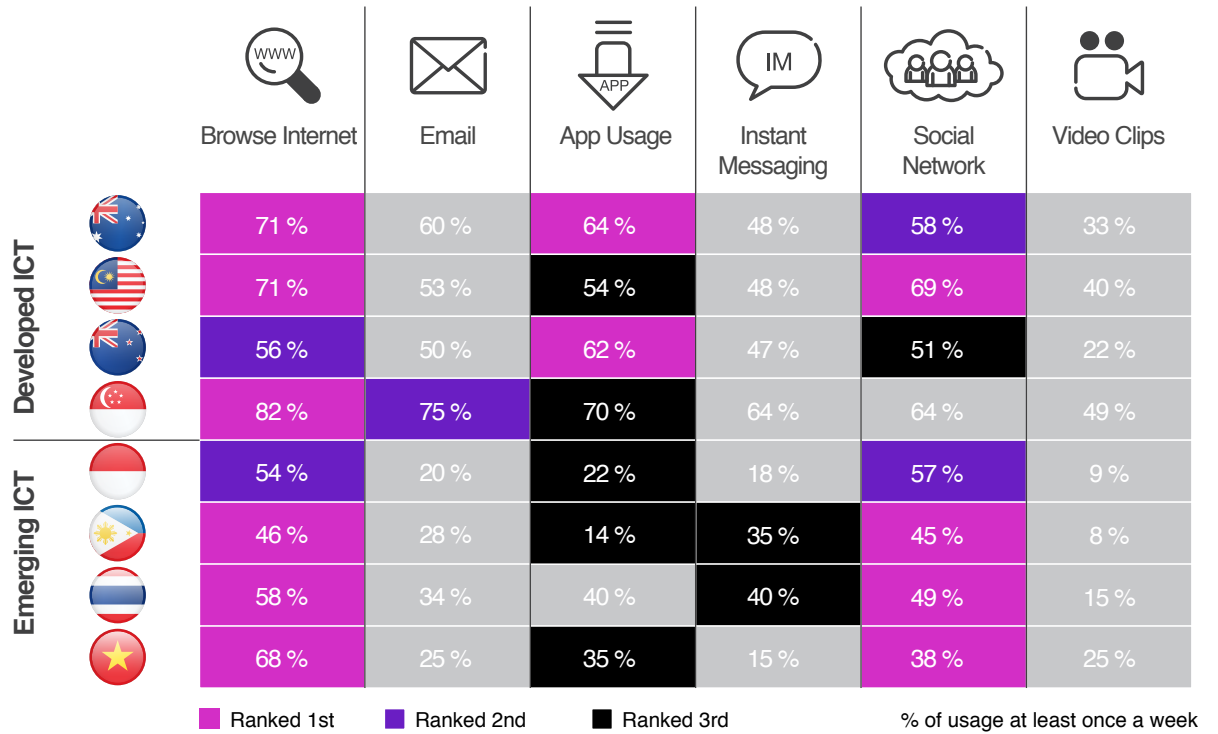
### *Smartphone Users Demand Connectivity Anywhere, Anytime*

Smartphones and other smart mobile devices have enabled new usage behaviours as increasing proportions of services accessed from these devices are, in fact, cloud-based. insights from ericsson consumerLab analysis on smartphone usage experience in Western europe reveal that regardless of the type of service and content being accessed, smartphone users want seamless connectivity all the time and, although users may not think about or be aware of 'the cloud', they still need services that are dependent on a good internet connection<sup>2</sup>. This changing behaviour brings new demands for mobile operators wishing to cater for today's consumer needs. Unlike laptops, smart mobile devices are used for shorter periods of time but much more frequently during the day, creating new usage patterns that need to be addressed by mobile network service providers. popular cloud-based services include email, social networking like facebook, video such asyoutube, and a growing array of dedicated applications. as shown in Figure 2, demand for such services is driving smartphone adoption and usage in both developed and emerging markets across Southeast asia and oceania<sup>3</sup>.

The mobile internet experience is evidently becoming more pervasive and instrumental to the lives of enterprise and consumer users alike. consequently, operators need to listen carefully to changing user needs to capture this ever growing market

opportunity. as figure 3 shows, almost 40 percent of smartphone users start their day by checking their devices, even before getting out of bed. once the day starts,

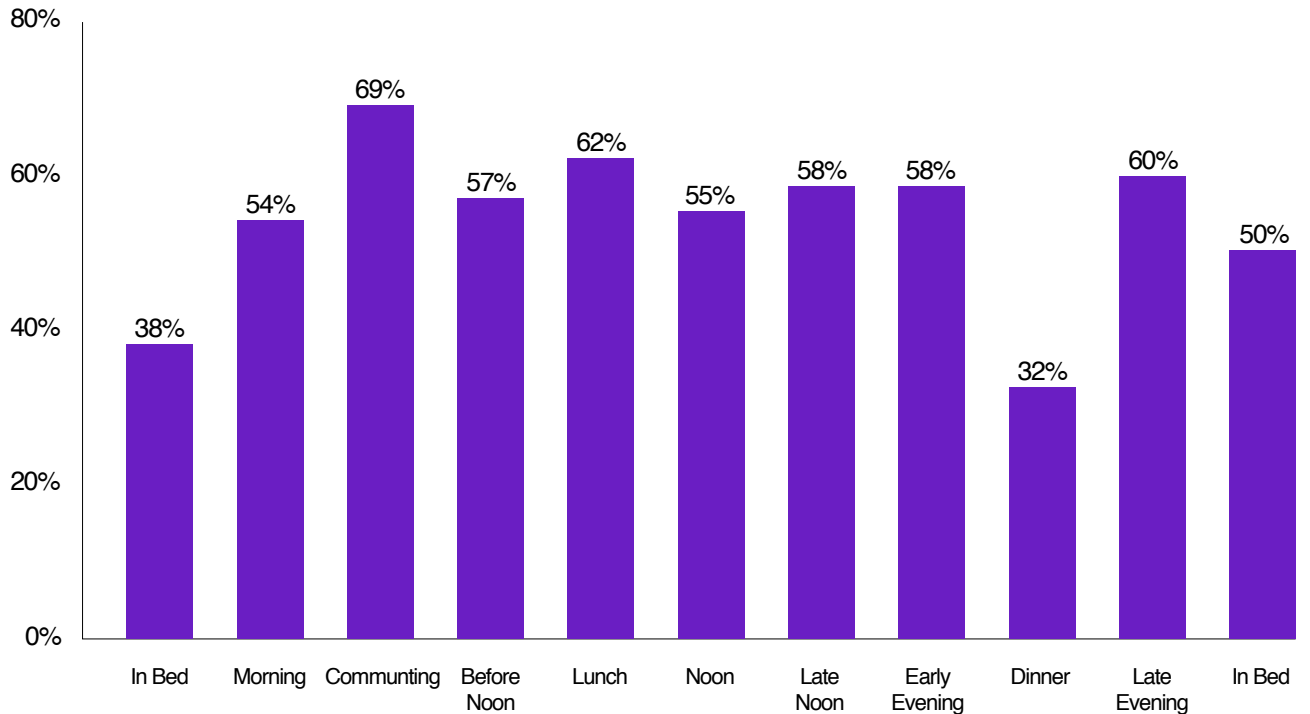
Figure 2. Key Drivers for Smartphone Usage



Source: Ericsson consumerLab, 2012



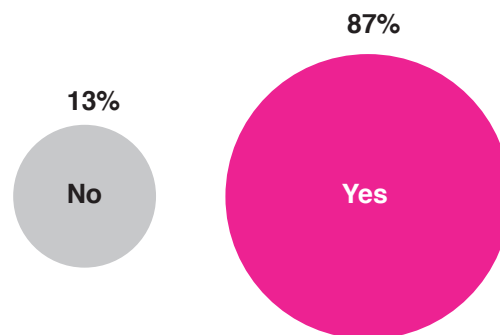
Figure 3. Non-Voice usage of Smartphones



Source: Ericsson consumerLab, 2011

usage patterns are rather consistent, peaking during the daily commute with a temporary decrease around dinner time <sup>4</sup>.

Ericsson ConsumerLab has found that close to nine in every ten smartphone users experience problems when using their devices<sup>2</sup>. Overall, half of the surveyed users experience issues on a weekly basis and the number of issues grows in line with device usage. Notwithstanding the fact that most of the problems are considered minor, they are occurring far too often and are perceived as irritating or even frustrating by users.



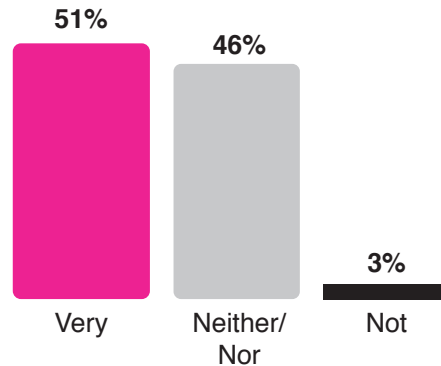
### *Do you experience issues with your smartphone?*

Among the problems faced, the most frustrating ones—by far—are internet connection issues, app crashes and slow speeds; this resonates well with users' need to constantly be connected and have a great experience while using their smartphones. With this in mind, how can operators come up with smartphone and tablet-ready quality networks that are up to the challenge posed by these new user demands?

### *Quality Networks Matter*

Just as smartphones have changed user behavior and new internet habits have developed, the concept and expectations of network quality have also evolved. Ericsson ConsumerLab has found that only 51 percent of smartphone users are very satisfied with their operator's network, leaving the other half in a position where their experience could definitely be improved<sup>2</sup>.

### *How satisfied are you with your operator's network?*



#### WHAT IS NETWORK QUALITY?

Base: Smartphone users in Finland and Netherlands

users, even in very different markets, tend to have fairly common expectations of the characteristics of a good mobile broadband experience. These include not only the earlier mentioned requirements of speed and coverage, but also a number of implicit qualities, namely:

- Convenience—users want to carry out tasks with little effort or difficulty. Devices and apps play an important part here, but so do the network and its speed.
- Immediacy—Latency and slow speeds must be minimized. Some users, when encountering a problem will either wait or try again, but as many as 40 percent of them will put their phones away, resulting in reduced usage and, over time, a potentially negative sentiment towards mobile broadband services
- Simplicity—although this falls largely under the terminal manufacturer domain, operators must carefully select the range of devices they want to offer. Smartphones, tablets and other mobile devices in general must be easy to use.
- Reliability—from the customer’s perspective, quality is defined as “having access to the network anytime, anywhere”; it is the service provider who is responsible for delivering this experience.

Operators must deliver upon all of the above aspects to achieve a good quality perception by their customers. In addition, these characteristics must also be consistently met because when problems are experienced more than once a week, user satisfaction is likely to decline<sup>2</sup>.

#### MANAGING USER PERCEPTIONS IS KEY

Ericsson ConsumerLab has found that customers tend to blame the network and the operator for most of their communication troubles and, although some of these issues can be caused by the device or application provider, most of the earlier mentioned problems causing major user frustrations do fall within the operator’s domain<sup>2</sup>.

Service providers have a high degree of control over network speeds, coverage and—to a certain extent—device battery life since poor coverage tends to drain batteries rapidly. It is always in the operator’s best interest to take the lead and work with other players in the ecosystem to deliver a high-quality user experience, thus improving customer perceptions.

A SUPERIOR NETWORK DRIVES CUSTOMER SATISFACTION

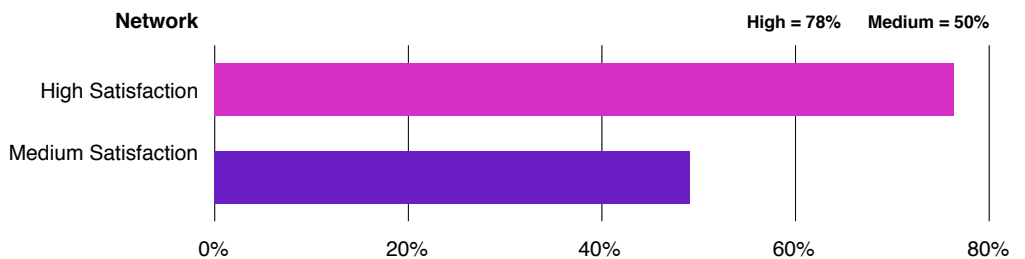
Near-ubiquitous coverage and a fast and reliable connection are the most important drivers of smartphone user satisfaction, and therefore must-haves for superior networks that stand out from competition<sup>1</sup>. Given that only half of users are very satisfied with their operator's network, there is substantial potential to generate competitive advantage through enhancement of network performance. Providing an excellent network with great coverage and fast data speeds where apps run smoothly is absolutely essential in order to achieve high customer satisfaction and increased revenues.

Other factors, such as price plans and device choice, are considered of somewhat lesser importance to users. It is worth noting that voice quality, while important, does not rank at the top when it comes to customer concerns today. Mobile users simply expect good voice quality in their network<sup>2</sup>.

SATISFIED USERS RECOMMEND THEIR NETWORKS

Network performance drives customer satisfaction and this translates into a profound effect on loyalty as users' word-of-mouth and recommendations turn into powerful brand promotion tools for operators. Customers with high satisfaction are found to be very loyal to their operators, with three-quarters planning to stay<sup>2</sup>. On the other hand, those customers with low satisfaction display a 50-50 chance of staying, suggesting that by just moving users from a low to high satisfaction level would dramatically reduce churn. For mobile operators to keep costs under control, it is even more critical to retain existing customers than to acquire new ones.

*How likely are you to stay with your current network provider?*



Research has shown that most network operators are yet to offer the ‘wow factor’ that will entice their customers to generate positive recommendations. the Net promoter Score (NpS) or likelihood of brand promotion by customers—calculated by plotting the percentage of users who would act as either detractors or promoters of a brand—shows that highly satisfied customers can increase an operator’s NpS

### *Operators Reap Benefits from Superior Networks —Telstra, Australia*

Having a high-quality and superior network can definitely pay off, australia’s largest operator, telstra has found. Not only was telstra ranked highest in overall network quality according to the J.D. power and associates 2012 australian Wireless Network Quality Study<sup>5</sup>, it also consistently delivered the country’s top mobile network speeds<sup>6</sup>.

Telstra’s superior-performance network has resulted in superior mobile business results, well ahead of the competition. for the financial year ending June 30, 2012 for mobiles, the operator enjoyed strong year-on-year revenue growth (8.5 percent), market-leading profitability (36 percent eBitDa), growing market share (signing up 1.6 million new customers), and increased customer satisfaction (customer complaints reduced by 26 percent)—all achieved in a highly-mature market with over 130 percent mobile subscription penetration.

### *Unwavering Focus on Network Quality and Coverage —Vivo, Brazil*

Telefonica’s arm in Brazil, Vivo, has for years focused its strategy on network quality and coverage according to the case study Vivo focuses on network coverage and quality to increase customer retention and loyalty from research firm informa telecoms & media<sup>7</sup>. this has translated into sustainable growth for the operator, mainly in the high-arpu postpaid segment. in the fourth quarter of 2011, Vivo’s share of Brazil’s postpaid market grew to a market-leading 33 percent, while its overall arpu carried a 10 percent premium over that of its closest competitor.

Furthermore, Vivo’s churn rates have remained consistently low (around 3 percent), despite the fact that it has achieved outstanding growth in the prepaid segment as well.

## *Attaining Superior Performing Networks*

In the transition from feature phones to smartphones and smart mobile devices, user perceptions of a service provider's network quality have never been more critical. In building quality networks, operators must ensure their networks are truly smart, simple, scalable, and deliver superior performance.

### SMART

Smart networks allow operators to gain awareness of users, their preferences and subscriptions, as well as the devices, apps and content being used—and to act on it. They allow operators to differentiate and to grow business. Smart networks also help operators save in real terms, as they optimize network resources. Smartness goes beyond network infrastructure to encompass support systems, providing the flexibility needed to respond to changing user behaviors and emerging business models. To achieve this, operators need solutions that are smart end-to-end; and the right partnerships are key in this process.

### SIMPLE

New business opportunities can emerge from anywhere and are sometimes unpredictable. Operators need to be agile and able to seize opportunities on the go; this requires turning the complex into simplicity. Converged platforms, with simple architecture and few interfaces, centralization, standardization and automation, will keep operational and capital expenditures and operational efforts down while enabling innovation and service differentiation.

### SCALABLE

Mobile broadband spawns changes in markets, businesses and technology. With mobile broadband demand projected to grow by 60 percent annually for the next five years, operators need to create scalable solutions that allow business to grow in line with—or even ahead of—the demand curve. Four growth dimensions constitute network scalability: bandwidth, devices, signaling and analytics. In short, networks must be able to ramp up quickly to meet exponential demand without service deterioration.

## SUPERIOR PERFORMANCE

Broadband is the highway to the cloud. its performance determines the user experience of all things digital and, importantly, it plays a pivotal role in operators' ability to monetize that experience. user experience cannot be tested in a lab, since true superior network performance in speed, latency, resiliency and capacity is what influences user satisfaction. Network design, tuning and assurance are critical to realizing superior- performing networks.

### *Achieving True Superior Performance*

To achieve superior customer satisfaction, operators need to develop and implement strategies that result in best-in-class network quality, speed and coverage. ericsson recommends that operators develop strategies based on the following five disciplines:

## CUTTING-EDGE PRODUCTS

Superior products with advanced features and algorithms ensuring great user experience even at the most extreme situations are central to high- performance networks, and benchmarking various ict suppliers gives a good indication of what the best solutions are. at the core of the mobile broadband ecosystem is the network infrastructure strategy, which needs to be complemented with a device strategy. high-end devices drive network capability while low-end ones drive capacity. it is important to keep the terminal offerings fresh in order to improve user experience and network efficiency while meeting demands from all segments.

## NETWORK DESIGN

The worldwide surge in smartphones and smart mobile devices has placed new and ever-changing demands on mobile networks and systems. to deliver true in-service performance, network design should be viewed by operators as a constantly evolving process. an optimum network design will ensure that typical bottlenecks which limit smartphone performance such as back-haul under-dimensioning and unnecessary inter-technology handovers that push smartphone users to slower networks are kept under control.

SWIFT IMPLEMENTATION

Swift implementation through automation is fueled by the need to decrease operational costs by automating simple and repetitive tasks, and by the need to reduce margins of error by automating complex tasks.

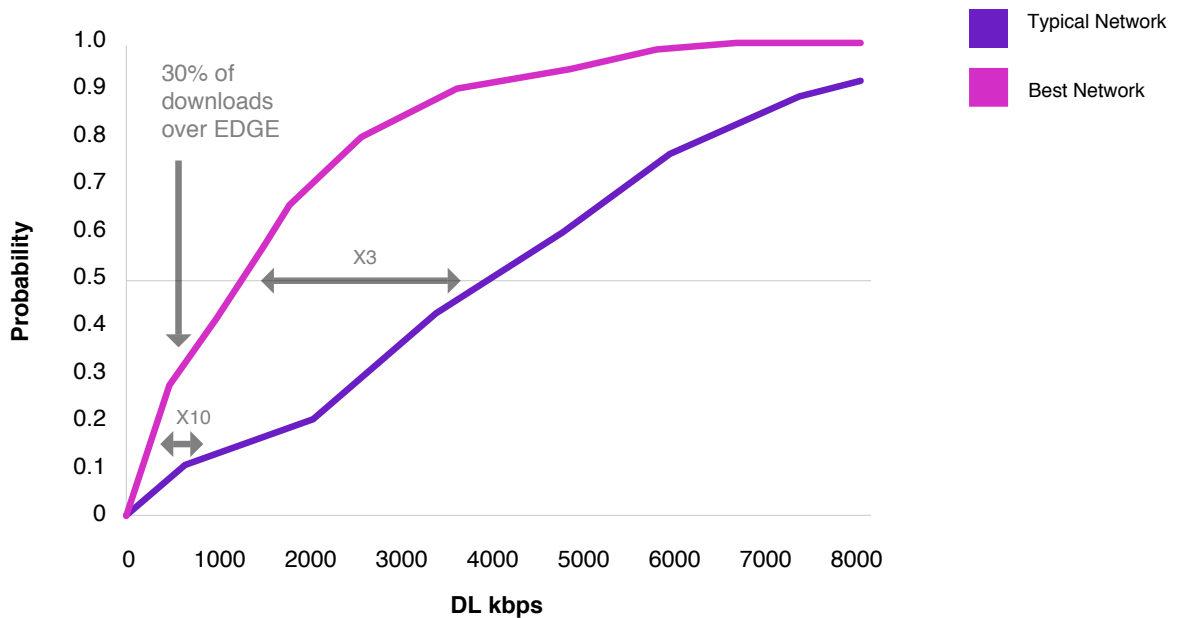
TUNING TO MAXIMIZE CURRENT ASSETS

operators can secure optimum network performance and end-user experience by continuously tuning their networks to ensure adequate capacity and coverage to meet customer needs. a network that maintains its competitive advantage can then be developed with the same product setup, leading to improved performance with minimum additional capital expenditures. in tuning networks, operators can simplify tasks by focusing on selected and relevant key performance indicators (kpis), collect measurements on those kpis, and translate measurements to actionable improvements.

SERVICE ASSURANCE

Total service assurance means not just taking care of customer issues but proactively delivering better service quality. While service quality management has traditionally relied on aggregated service kpis, ericsson customer experience assur-

Figure 5. A Superior Network makes a Difference for Operators





ance 8 enables the drill-down of customer experience to an individual level. This gives greater flexibility and insight in tracking specific issues while service affecting faults can be prioritized based on customer impact—an impact that includes value metrics derived from other information systems.

### *Conclusion*

Mobile broadband users rely on a good connection to the internet, and this dependency is becoming even more critical now with the rapid uptake of smartphones, where connection to the cloud and access to apps are demanded anytime and anywhere. users are also becoming increasingly sophisticated, with stronger opinions on what—in their own view—are the causes of their user experience issues, prompting operators to place a greater focus on improving factors that affect user experience.

Customers are highly sensitive to the quality of their mobile networks. the good news is operators can significantly enhance customer perception on network quality by addressing the top three problems: internet connection issues, app and service crashes, and slow network speeds. even though not all of these issues are within their direct control, mobile network operators need to realize that often, customers will blame the networks for most of their problems. as a result, carriers must take the lead and engage with other players in the ecosystem—such as device manufacturers, ict infrastructure suppliers and application developers—in order to deliver a true superior user experience.

Indeed, good network coverage and a fast, reliable connection are mandatory. they command higher user spend and are the strongest drivers of mobile broadband customer satisfaction. a strong correlation exists between highly satisfied customers and their likelihood to stay with their current service providers—and furthermore to positively recommend them to others. a network with superior performance is a strategic asset that operators must create, proactively monitor and sustain to attract new users and, perhaps more importantly, keep existing users happy.

In a very competitive connected world, *the network is one of the key differentiators*. Operators with value propositions that include an optimum combination of

high-end and mass-market devices, clear service differentiation strategies and the ability to deliver networks that are truly smart, simple, scalable and with superior performance, will become market leaders, as they turn customers into loyal brand promoters.

## *References*

1. Ericsson 2012, traffic and market report, June 2012, retrieved 10 July 2012, <<http://www.ericsson.com/traffic-market-report>>
2. Ericsson ConsumerLab 2012, Study on Smartphone usage experience
3. Ericsson ConsumerLab analytical platform 2012
4. Ericsson consumerLab analytical platform 2011
5. J.D.power and associates, the mcGraw-hill companies, inc 2012, 2012 australian Wireless Network Quality Study, retrieved 10 July 2012, <<http://www.jdpower.com/content/press-release/Z6zhvai/2012-australian-wireless-network-quality-study.htm>>
6. Telstra Corporation Limited 2012, Next G mobile Network Works Better in more places, retrieved 10 July 2012, <<http://www.telstra.com.au/mobile-phones/coverage-networks/network-information/nextg/>>
7. Informa telecoms & media 2012, case study: Vivo focuses on network coverage and quality to increase customer retention and loyalty
8. Ericsson 2012, Ericsson customer experience assurance, retrieved 30 July 2012, <<http://www.ericsson.com/ourportfolio/products/customer-experience-assurance>>

## *Further Reading*

### CONSUMERLAB RESEARCH

From Apps To Everyday Situations

[http://www.ericsson.com/res/docs/2011/silicon\\_valley\\_brochure\\_letter.pdf](http://www.ericsson.com/res/docs/2011/silicon_valley_brochure_letter.pdf)

Emerging App Culture

[http://www.ericsson.com/res/docs/2012/ericsson\\_emerging\\_app\\_culture.pdf](http://www.ericsson.com/res/docs/2012/ericsson_emerging_app_culture.pdf)  
[http://www.ericsson.com/res/docs/2012/oce\\_design\\_optimal\\_consumer\\_experience.pdf](http://www.ericsson.com/res/docs/2012/oce_design_optimal_consumer_experience.pdf)

Smarter Mobile Broadband

[http://www.ericsson.com/res/thecompany/docs/publications/business-review/2012/issue1/Smarter\\_mobile\\_Broadband.pdf](http://www.ericsson.com/res/thecompany/docs/publications/business-review/2012/issue1/Smarter_mobile_Broadband.pdf)

### ERICSSON TRAFFIC AND MARKET REPORT

Ericssontraffic And Market Report

<http://www.ericsson.com/traffic-market-report>

### ERICSSON PAPERS

Profitable Prepaid Smartphones

[http://www.ericsson.com/res/region\\_raSo/docs/2012/ericsson\\_prepaid\\_paper\\_june.pdf](http://www.ericsson.com/res/region_raSo/docs/2012/ericsson_prepaid_paper_june.pdf)

Why Smartphones Need Smart Networks

[http://www.ericsson.com/res/docs/2011/111014\\_smartphone\\_brochure.pdf](http://www.ericsson.com/res/docs/2011/111014_smartphone_brochure.pdf)

**CONCLUSION** Making affordable Internet access a reality for the next 5 billion people depends on the industry achieving a dramatic improvement in the overall efficiency of delivering data. Over the next decade, we believe this is a realistic prospect. By bringing down the underlying costs of data, and building more efficient apps that use less data, we believe we can increase the efficiency of delivering data by 100x—opening up an entirely new world of connected technologies, experiences and opportunities to people in developing countries.

Achieving this mission is clearly beyond the capabilities of any one company, or any one technical approach. It will depend on telecommunications and technology leaders working together to drive new innovations across platforms, devices, operating systems and infrastructure. In turn, this will require a new level of industry collaboration, focused on sharing knowledge, developing and implementing best practices, and building and sharing new tools, systems and technologies. This is the purpose of Internet.org.

With this paper we hope to provide a small contribution to the work of countless companies, entrepreneurs and innovators working to drive new gains in efficiency. By understanding some of the efforts that have already been made towards this end, and the lessons learned along the way, we hope to inspire new thinking and approaches. We don't have all the answers, but we want to work together with the entire industry to ask the right questions and we're committed to solving this challenge. Affordable Internet access is transformative for people and communities everywhere—and it starts with a focus on efficiency.

#### SOURCES

Avatarnode:

<https://www.facebook.com/notes/facebook-engineering/under-the-hood-hadoop-distributed-filesystem-reliability-with-namenode-and-avata/10150888759153920>

Corona:

<https://www.facebook.com/notes/facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920>

Giraph:

<https://www.facebook.com/notes/facebook-engineering/scaling-apache-giraph-to-a-trillion-edges/10151617006153920>

HipHop:

<https://www.facebook.com/notes/facebook-engineering/hiphop-for-php-more-optimizations-for-efficient-servers/10150121348198920>

HHVM:

[https://www.facebook.com/note.php?note\\_id=10150415177928920](https://www.facebook.com/note.php?note_id=10150415177928920)

HHVM:

<https://www.facebook.com/notes/facebook-engineering/speeding-up-php-based-development-with-hiphop-vm/10151170460698920>

OCP/Data centers

Water efficiency:

<http://www.opencompute.org/2012/08/09/water-efficiency-at-facebooks-prineville-data-center/>

OCP Summit IV:

<http://www.opencompute.org/2013/01/16/ocp-summit-iv-breaking-up-the-monolith/>

Cooling:

<http://www.opencompute.org/2012/11/14/cooling-an-ocp-data-center-in-a-hot-and-humid-climate/>

Hardware:

<http://www.opencompute.org/2012/10/24/deploying-ocp-hardware-in-a-co-located-facility/>

Prineville: <http://www.opencompute.org/2011/11/17/learning-lessons-at-the-prineville-data-center/>