

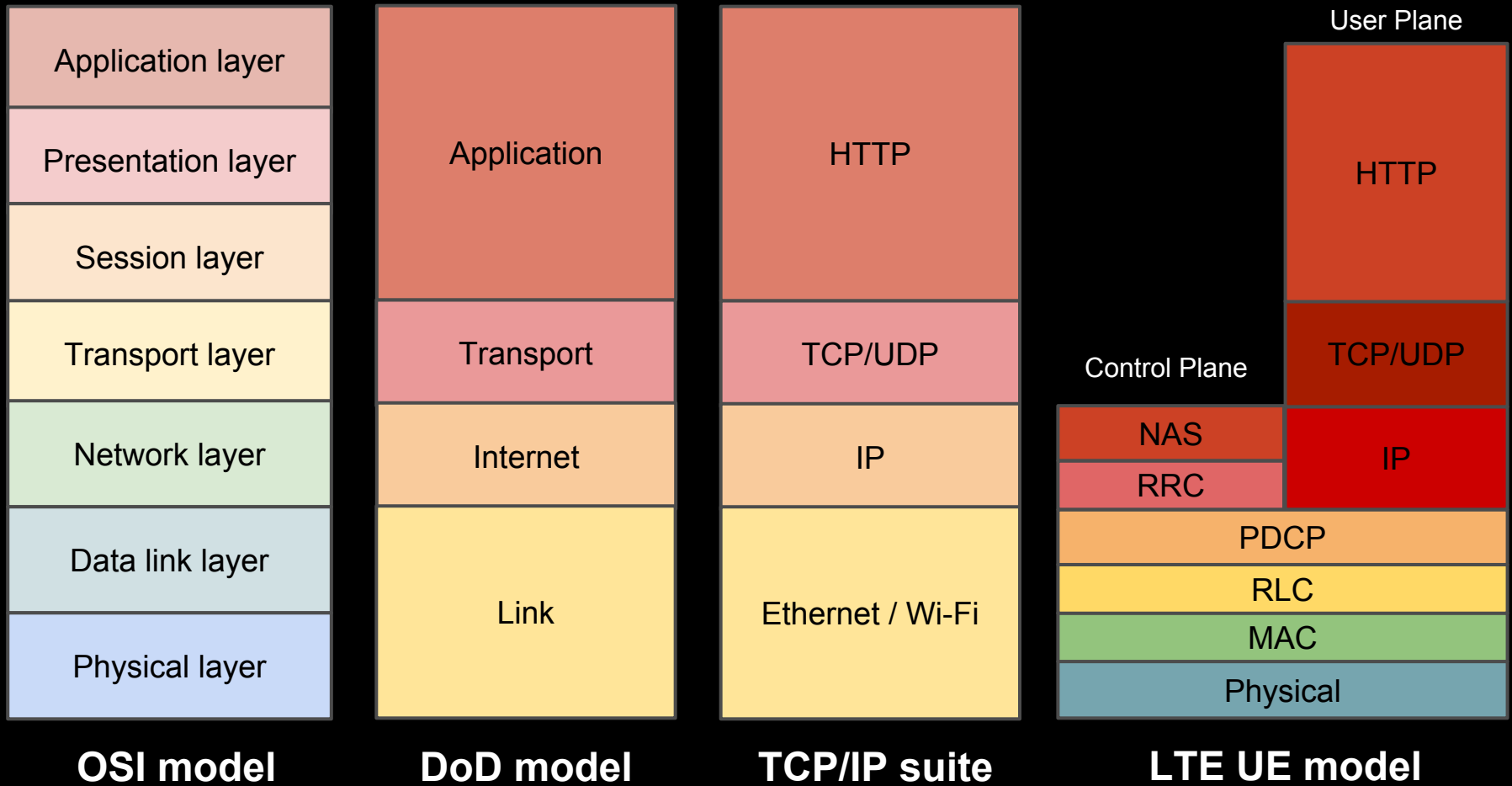
Accessing Cellular Network Efficiently Without the Battery Drain

Lee, Chang Hwan, SK Telecom
lipisoft at gmail dot com
Kandroid: Lipi(리피)

Contents

- Wired vs Cellular
- Cellular Network
- LTE Protocol Stack
- LTE RRC state
- AT&T ARO
- Monitor Battery status and Connectivity
- Case Study
 - Remove unnecessary traffic
 - TCP FIN Retry
 - Redirect to Mobile Web page
- Additional Readings

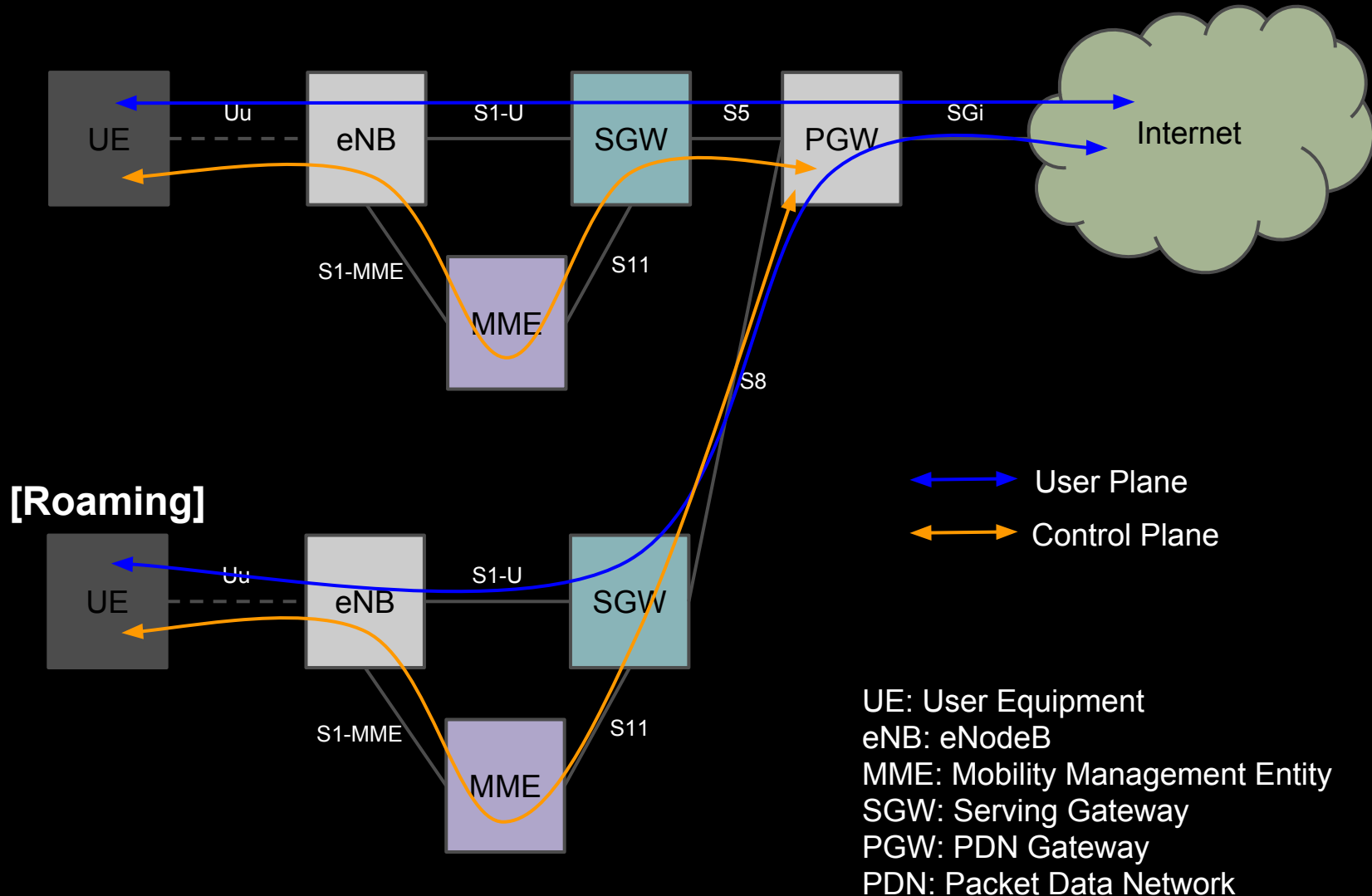
Wired vs Cellular(Protocol Stack)



PDCP: Packet Data Convergence Protocol
RLC: Radio Link Control
MAC: Medium Access Control

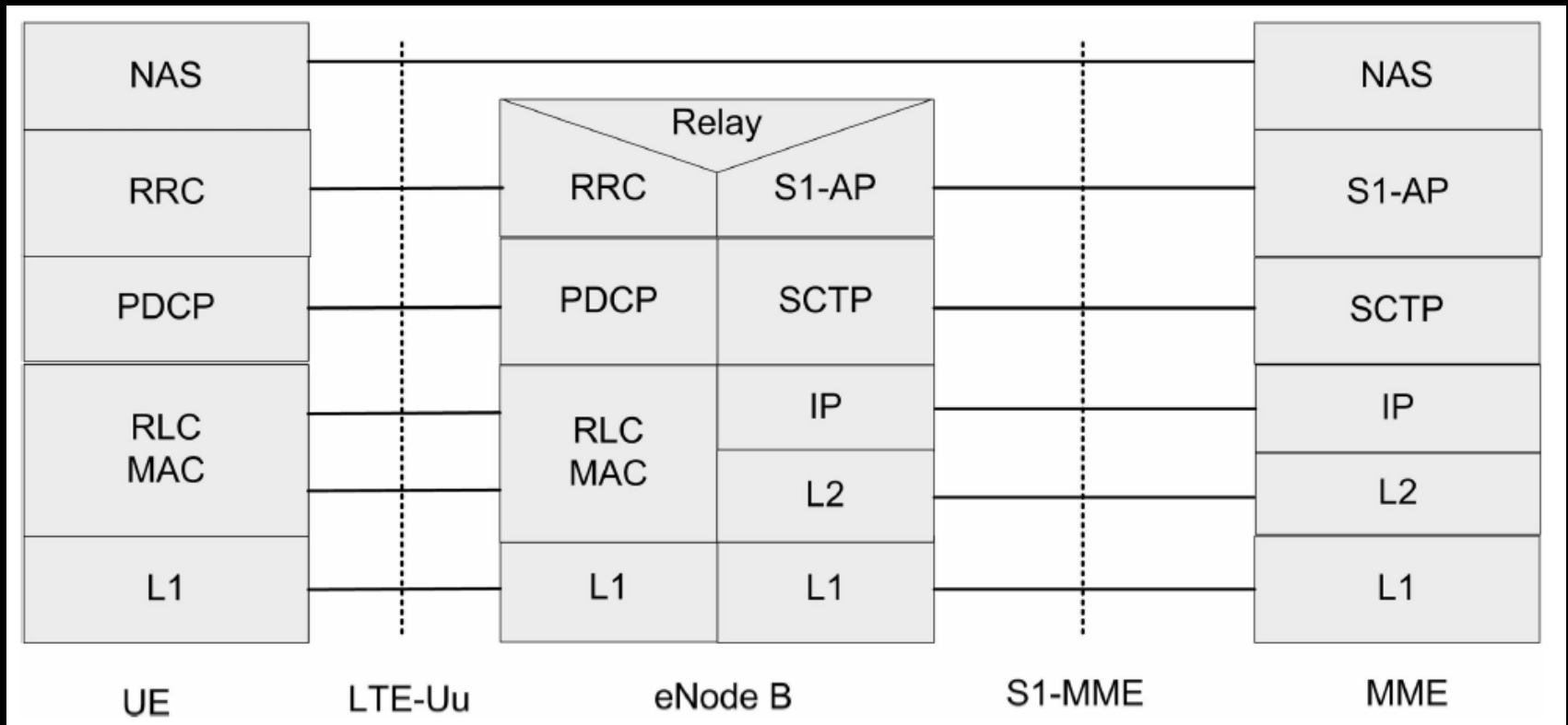
NAS: Non Access Stratum
RRC: Radio Resource Control

Cellular Network(LTE)



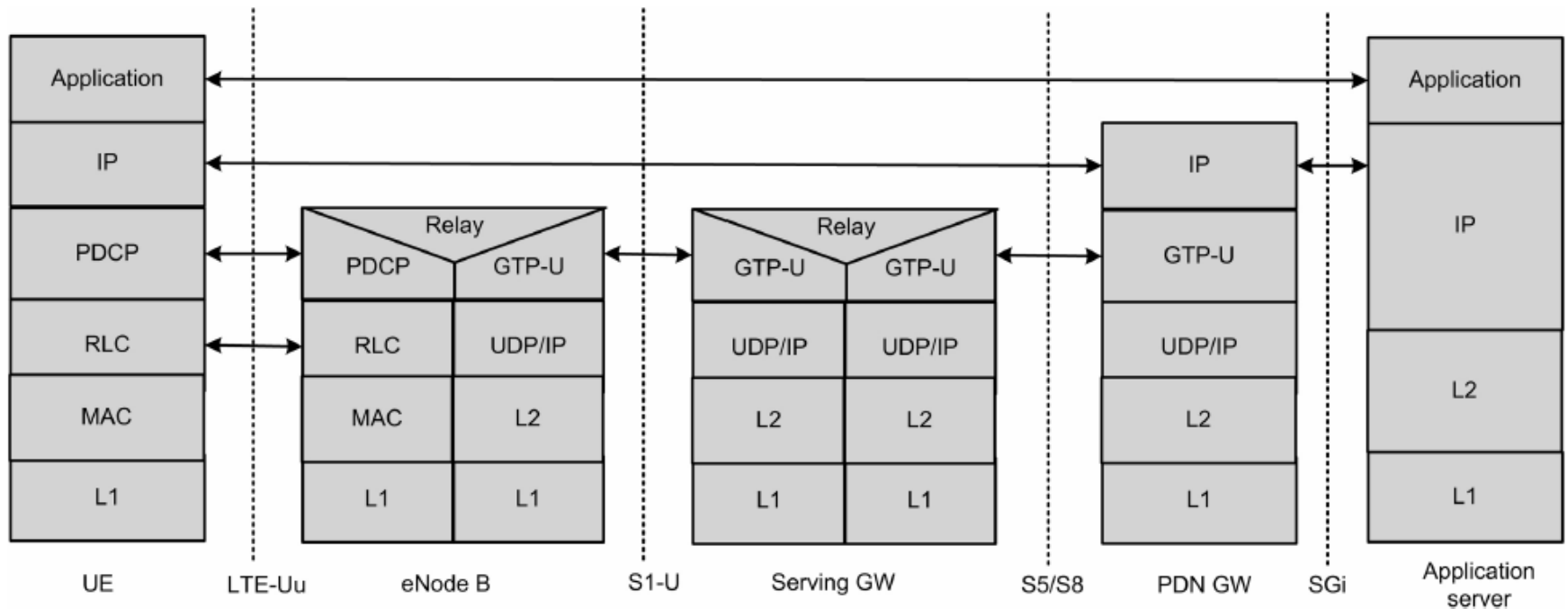
LTE Protocol Stack

Control Plane

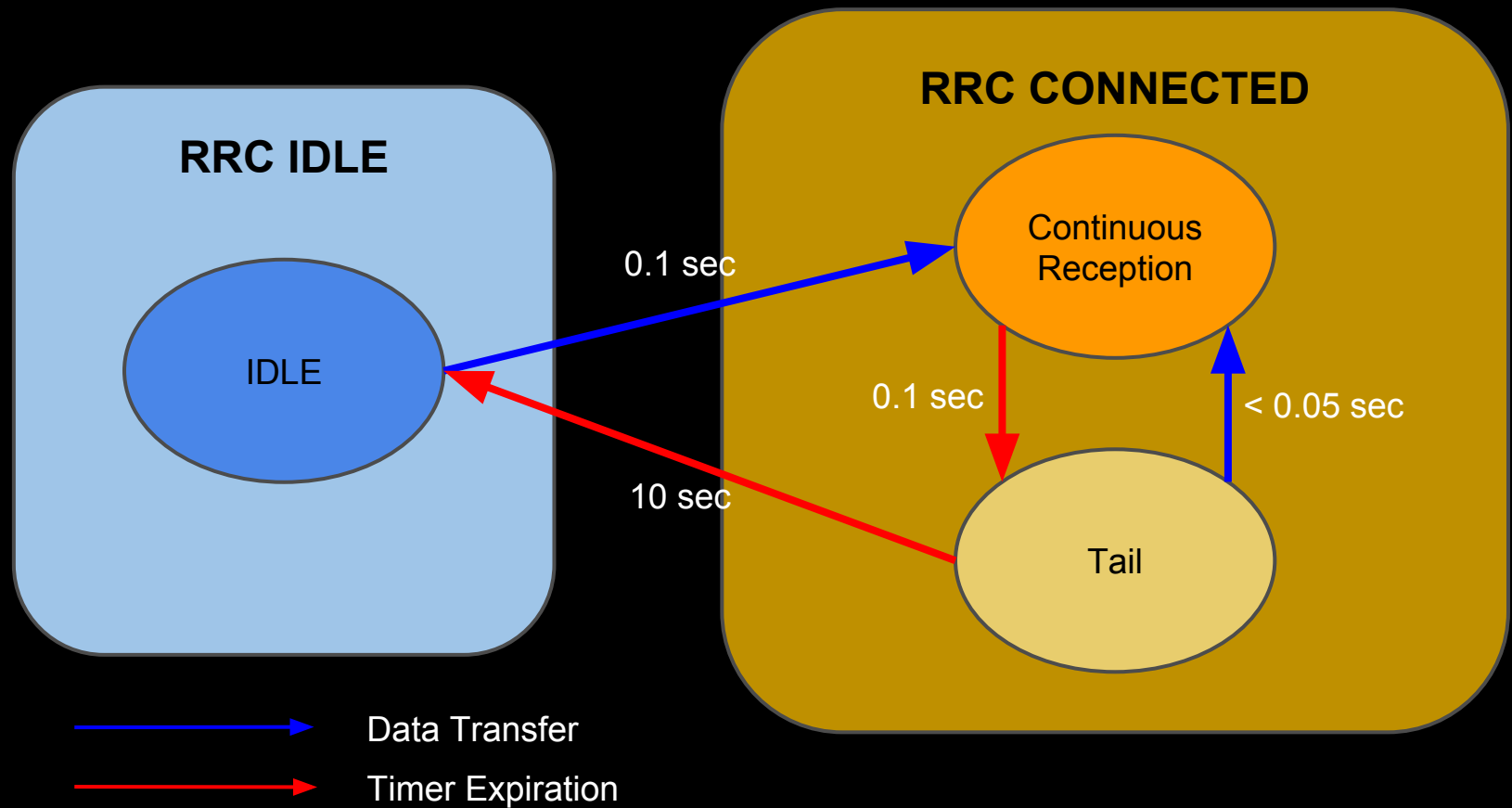


LTE Protocol Stack

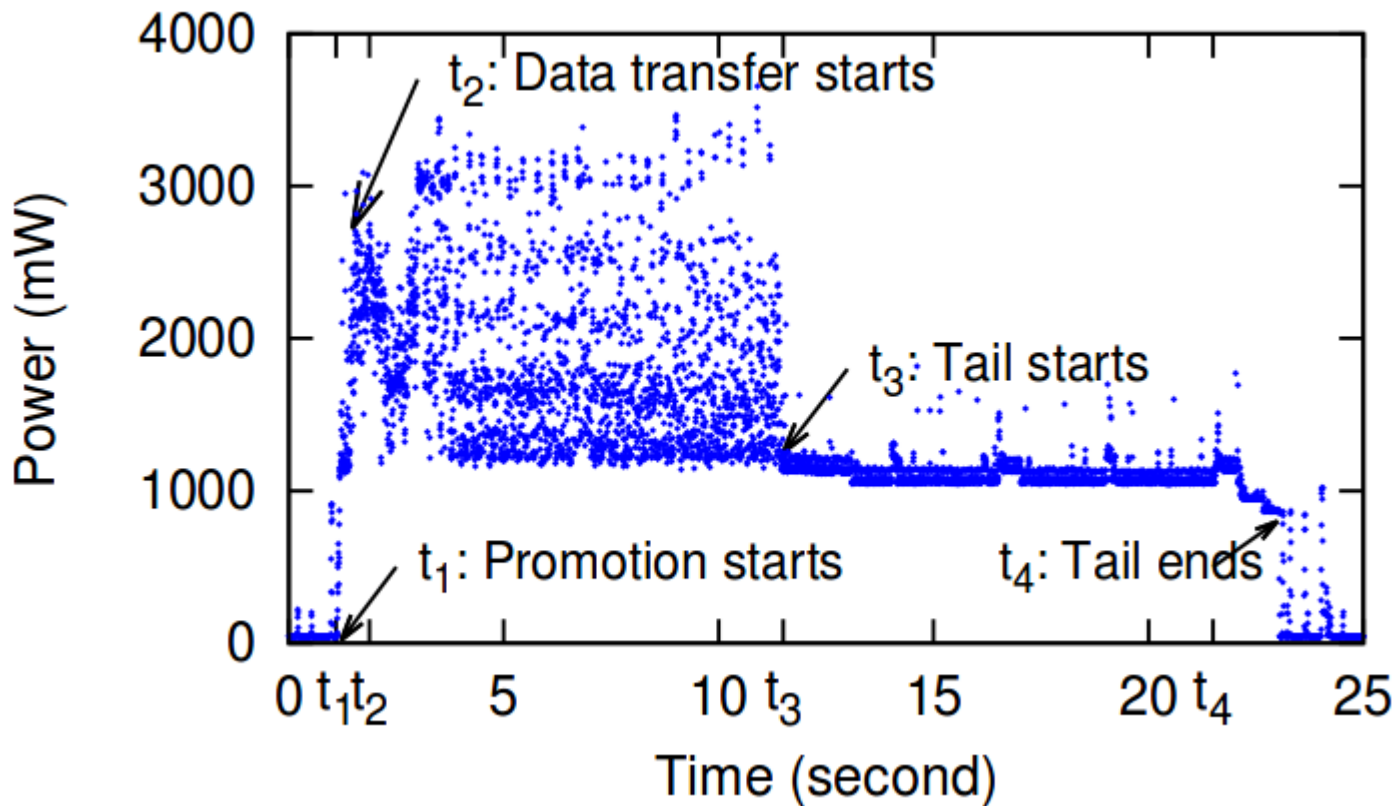
User Plane



LTE RRC State and Time Consumption



Power states of LTE



- Radio is the **second** most expensive component after **screen**
- Radio use is expensive **regardless of transfer size**

Download a HTML page in your App

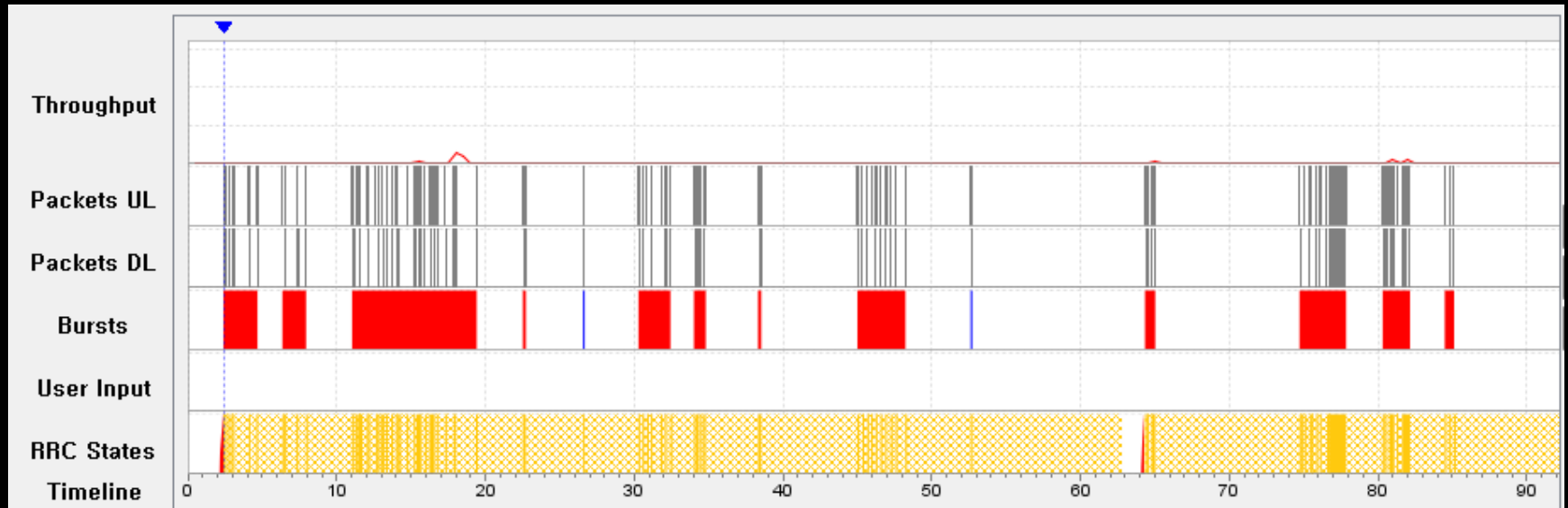
1. Launch the Application
2. RRC IDLE -> RRC CONNECTED
 - a. It takes 2 seconds at 3G network
3. DNS Query to DNS Server and Response
4. TCP SYN, SYN/ACK, ACK
5. HTTP GET Request/200 OK

AT&T ARO



- Application Resource Optimizer
- Latest Version: 2.4
- Open Source(<https://github.com/attdevsupport/ARO>), Apache License 2.0
- Data Collector
 - must be Rooted(su command is required) in Android
 - tcpdump is used for packet capture in Android
 - NetMon in Windows
- Data Analyzer
 - Windows & Mac OSX(Java based)
- Filter per an application/IP address
- Flurry is used but can be removed from source code and rebuild it
- Limitation
 - HTTP Header in HTTPS can't be analyzed

AT&T ARO Data Analyzer



TCP/UDP Flows

Time	Application	Domain Name	Local Port	Remote Port	Packet Count	TCP/UDP
2,387	com.kakao.talk	110,76,141,20	local:58848	110,76,141,20:5228	10	TCP
2,802	com.kakao.talk	110,76,141,168	local:42525	110,76,141,168:5228	17	TCP
4,085	...ndroid,vending, com.google.proces...	173,194,38,102	local:55929	173,194,38,102:443	140	TCP
11,145	com.google.process.gapps	mtalk.google.com	local:56098	173,194,72,188:5228	48	TCP
13,667	com.sec.android.app.samsungapps	14,0,70,99	local:53998	14,0,70,99:443	4	TCP
15,177	com.google.android.apps.maps	clients4.google.com	local:55999	173,194,38,96:443	43	TCP
22,513	com.google.android.youtube	www.googleadservices...	local:43198	173,194,38,121:80	10	TCP
30,325	com.twitter.android	settings.crashlytics.com	local:59703	54,235,164,80:443	22	TCP
34,157	com.skt.rcs:nable_lisp	113,217,234,33	local:37673	113,217,234,33:8082	19	TCP

ARO Best Practice

- Text File Compression
- Duplicate Content
- Cache Control
- Cach Expiration
- Content Prefetch
- Combine JavaScript and CSS requests
- Resize Images for Mobile
- Minify CSS, JS, and HTML
- Managing Imags with CSS Sprites
- Opening Connections
- Multiple Simultaneous TCP Connections
- Periodic Transfer
- Screen Rotations
- Closing Connections
- Offloading to Wi-Fi
- Third-Party Scripts
- HTTP 400 and 500 Error Codes
- HTTP 300 Status Codes
- Asynchronous Load of JavaScript in HTML
- HTTP 1.0 Usage
- File Order of External Style Sheets and Scripts
- Empty Source and Link Attributes
- Flash Files
- Displaying None in CSS
- Accessing Peripherals
- Comparing LTE and 3G Energy Consumption
- Content Optimazation for Mobile Devices

Text File Compression



Test: Text File Compression

About: Sending compressed files over the network will speed delivery, and unzipping files on a device is a very low overhead operation. Ensure that all your text files are compressed while being sent over the network. [Learn more...](#)

Results: AT&T ARO detected 53% of text files (93kB) were sent without compression. Adding compression will speed the delivery of your content to your customers.

Time	Host Name	File Size	File Name
31,517	appdown.naver.com	742	/naver/android/nni/v1/service_list.xml
32,150	appdown.naver.com	219	/naver/android/nni/v1/global/service.xml
85,888	tig.nate.com	246	/TRS?SERVICE_ID=61001610016000A
87,657	appdown.naver.com	742	/naver/android/nni/v1/service_list.xml
88,578	appdown.naver.com	219	/naver/android/nni/v1/global/service.xml

Request/Response View Packet View Content View

```
GET /naver/android/nni/v1/service_list.xml HTTP/1.1
Host: appdown.naver.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

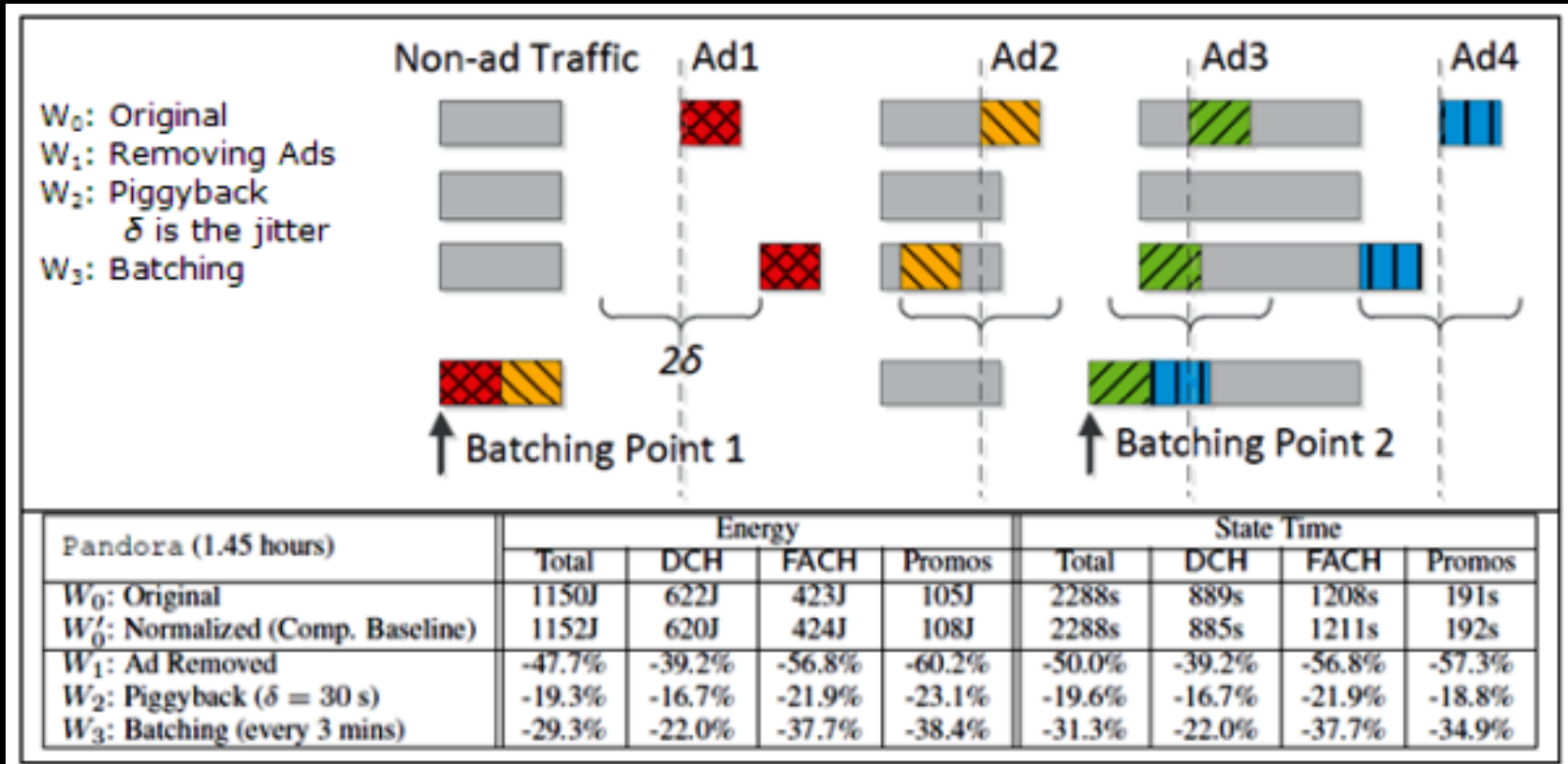
HTTP/1.1 200 OK
Date: Tue, 01 Oct 2013 10:42:54 GMT
Server: GSCDN/G-Platform/1.1.7 (Unix) DAV/2
Last-Modified: Tue, 06 Nov 2012 10:04:37 GMT
ETag: "2e6-4cdd0bac1d740"
Accept-Ranges: bytes
Content-Length: 742
Content-Type: application/xml
Age: 87
```

- **No encoding** method is specified in HTTP header

Text File Compression

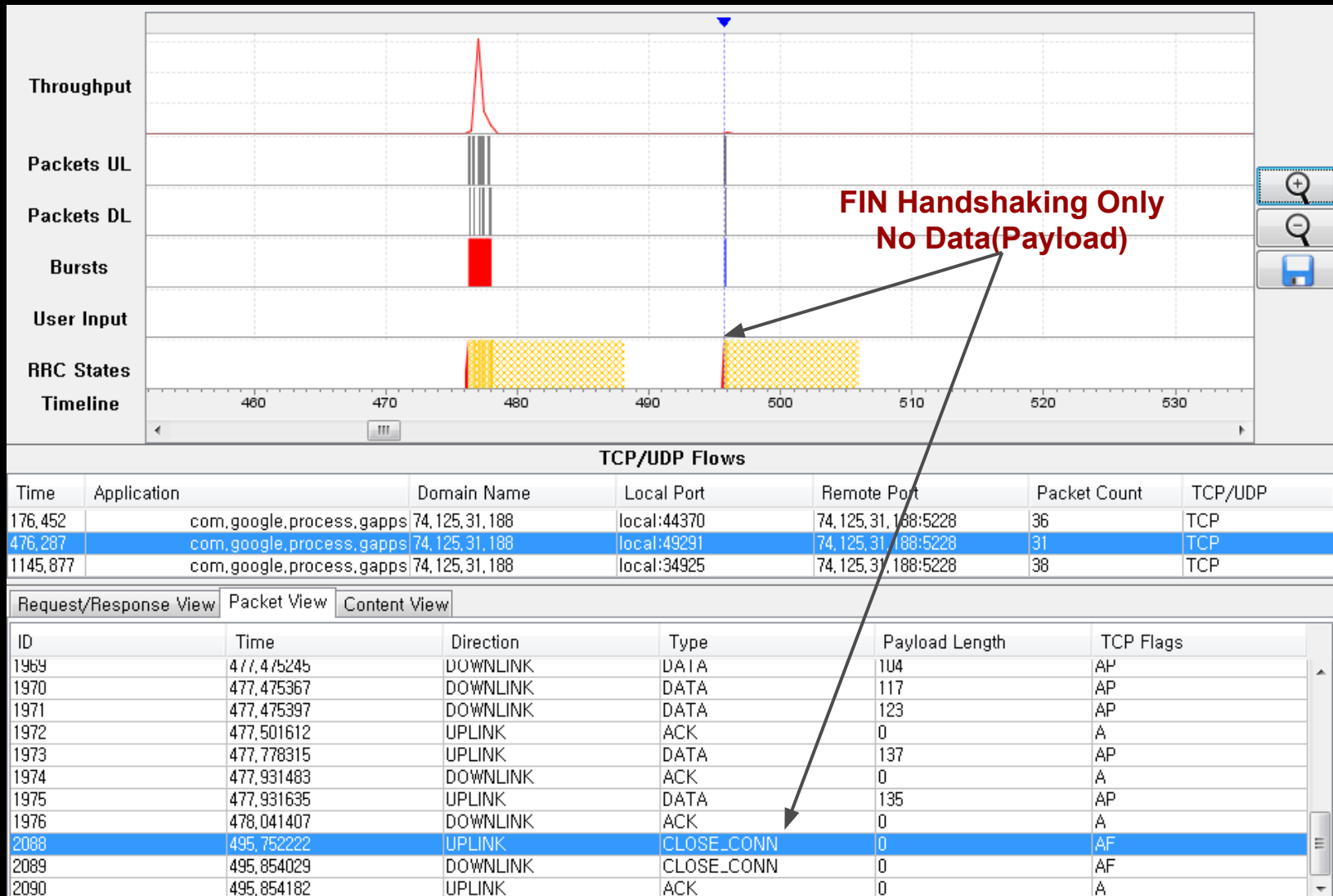
- HTML, CSS, JavaScript, Text is used
- HTTP 1.1 can support gzip, DEFLATE
- Limitation
 - image format and media format
 - very small files
- How to
 - App send *Accept-Encoding: gzip, deflate* in HTTP request header
 - Server send *Content-Encoding: gzip or deflate* in HTTP response header

Periodic Data Transfer

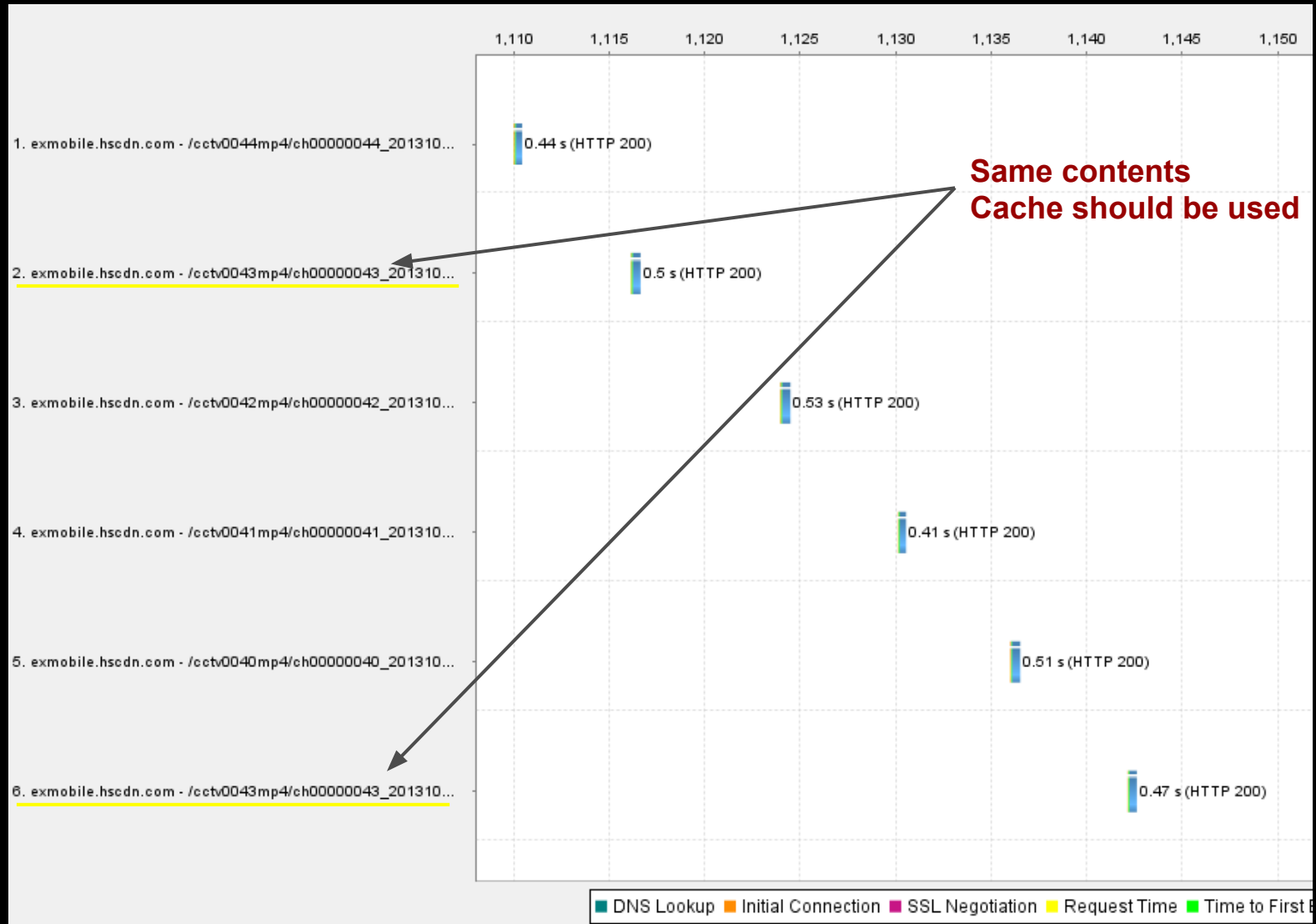


- Minimize the periodic data transfers
- Avoid polling, leverage server push such as GCM(Google Cloud Message)
- coalesce, defer requests

Closing Connections



Cache Control/Expiration



Optimizing Battery Life

- Monitoring the Battery Level and Charging State
- Determining and Monitoring the Docking State and Type
- Determining and Monitoring the Connectivity Status
- Manipulation Broadcast Receivers On Demand

Monitoring the Battery Level and Charging State

- Determine the Current Charging State

```
IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);  
Intent batteryStatus = context.registerReceiver(null, ifilter);
```

```
// Are we charging / charged?
```

```
int status = batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);  
boolean isCharging = status == BatteryManager.BATTERY_STATUS_CHARGING ||  
    status == BatteryManager.BATTERY_STATUS_FULL;
```

```
// How are we charging?
```

```
int chargePlug = batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);  
boolean usbCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_USB;  
boolean acCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_AC;
```

Monitoring the Battery Level and Charging State

```
<receiver android:name=".PowerConnectionReceiver">
    <intent-filter>
        <action android:name="android.intent.action.ACTION_POWER_CONNECTED"/>
        <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED"/>
    </intent-filter>
</receiver>
```

```
public class PowerConnectionReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        int status = intent.getIntExtra(BatteryManager.EXTRA_STATUS, -1);
        boolean isCharging = status == BatteryManager.BATTERY_STATUS_CHARGING ||
            status == BatteryManager.BATTERY_STATUS_FULL;

        int chargePlug = intent.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);
        boolean usbCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_USB;
        boolean acCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_AC;
    }
}
```

Monitoring the Battery Level and Charging State

- **Determine the Current Battery Level**

```
int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);  
int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);  
  
float batteryPct = level / (float)scale;
```

- **Monitor Significant Changes in Battery Level**

```
<receiver android:name=".BatteryLevelReceiver">  
  <intent-filter>  
    <action android:name="android.intent.action.ACTION_BATTERY_LOW"/>  
    <action android:name="android.intent.action.ACTION_BATTERY_OKAY"/>  
  </intent-filter>  
</receiver>
```

Determining and Monitoring the Connectivity Status

- Check an Internet Connection

```
ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.  
CONNECTIVITY_SERVICE);
```

```
NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
```

```
boolean isConnected = activeNetwork != null && activeNetwork.isConnectedOrConnecting();
```

- Check the Type of Internet Connection

```
boolean isWiFi = activeNetwork.getType() == ConnectivityManager.TYPE_WIFI;
```

```
boolean isMobile = activeNetwork.getType() == ConnectivityManager.TYPE_MOBILE;
```

- Monitor for Changes in Connectivity

```
<action android:name="android.net.conn.CONNECTIVITY_CHANGE"/>
```

Manipulating Broadcast Receivers On Demand

- wake the device each time any of these receivers is triggered
- enable or disable the broadcast receivers at runtime

```
ComponentName receiver = new ComponentName(context, myReceiver.class);  
PackageManager pm = context.getPackageManager();  
pm.setComponentEnabledSetting(receiver,  
    PackageManager.COMPONENT_ENABLED_STATE_ENABLED,  
    PackageManager.DONT_KILL_APP);
```

Case Study

Bank Application

No.	Time	delta time	Source	Destination	Protocol	Length	Sequence number	Info
17	2013-10-20 12:35:58.611	0.000	10.179.1.197	211.32.160.195	TCP	76	3027085080	50019 > xmpp-client [SYN] Seq=3027085080 win=14000
18	2013-10-20 12:35:58.773	0.161	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50019 [RST, ACK] Seq=0 Ack=302708508
1573	2013-10-20 12:40:58.825	300.051	10.179.1.197	211.32.160.195	TCP	76	2096008855	50055 > xmpp-client [SYN] Seq=2096008855 win=14000
1574	2013-10-20 12:40:59.034	0.208	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50055 [RST, ACK] Seq=0 Ack=209600885
3562	2013-10-20 12:45:58.710	299.676	10.179.1.197	211.32.160.195	TCP	76	3058857454	50114 > xmpp-client [SYN] Seq=3058857454 win=14000
3563	2013-10-20 12:45:58.766	0.055	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50114 [RST, ACK] Seq=0 Ack=305885745
4858	2013-10-20 12:50:58.757	299.991	10.179.1.197	211.32.160.195	TCP	76	1722711899	50139 > xmpp-client [SYN] Seq=1722711899 win=14000
4859	2013-10-20 12:50:58.922	0.164	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50139 [RST, ACK] Seq=0 Ack=172271190
5034	2013-10-20 12:55:58.661	299.739	10.179.1.197	211.32.160.195	TCP	76	3919989704	50144 > xmpp-client [SYN] Seq=3919989704 win=14000
5035	2013-10-20 12:55:58.824	0.163	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50144 [RST, ACK] Seq=0 Ack=391998970
5044	2013-10-20 13:00:58.619	299.794	10.179.1.197	211.32.160.195	TCP	76	3391305610	50145 > xmpp-client [SYN] Seq=3391305610 win=14000
5045	2013-10-20 13:00:58.786	0.167	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50145 [RST, ACK] Seq=0 Ack=339130561
5049	2013-10-20 13:05:58.616	299.829	10.179.1.197	211.32.160.195	TCP	76	600604132	50146 > xmpp-client [SYN] Seq=600604132 win=14000
5050	2013-10-20 13:05:58.801	0.185	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50146 [RST, ACK] Seq=0 Ack=600604133
5051	2013-10-20 13:10:58.616	299.814	10.179.1.197	211.32.160.195	TCP	76	1913846200	50147 > xmpp-client [SYN] Seq=1913846200 win=14000
5052	2013-10-20 13:10:58.797	0.181	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50147 [RST, ACK] Seq=0 Ack=191384620
7797	2013-10-20 13:15:58.875	300.078	10.179.1.197	211.32.160.195	TCP	76	1122430561	50168 > xmpp-client [SYN] Seq=1122430561 win=14000
7798	2013-10-20 13:15:58.923	0.047	211.32.160.195	10.179.1.197	TCP	56		0 xmpp-client > 50168 [RST, ACK] Seq=0 Ack=112243056

ELAPSED_WAKEUP #10: Alarm{42885370 type 2 com.sc.danb.scbankapp}

type=2 when=+4m39s363ms repeatInterval=300000 count=1

operation=PendingIntent{431152f8: PendingIntentRecord{43bd0570 com.sc.danb.scbankapp broadcastIntent}}

- The trial of useless connection should be removed in your app.

netstat

```
shell@mako:/ $ netstat
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp6	0	1	::ffff:172.22.21.155:46363	::ffff:74.125.235.107:80	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:59386	::ffff:74.125.235.140:80	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:55689	::ffff:74.125.235.131:80	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:42277	::ffff:74.125.235.136:443	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:50344	::ffff:74.125.235.110:80	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:46585	::ffff:173.194.38.108:443	CLOSE_WAIT
tcp6	0	0	::ffff:172.22.21.155:53935	::ffff:74.125.31.95:443	ESTABLISHED
tcp6	0	0	::ffff:172.22.21.155:52280	::ffff:173.194.72.188:5228	ESTABLISHED
tcp6	0	1	::ffff:172.22.21.155:45740	::ffff:74.125.235.140:443	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:44794	::ffff:74.125.235.140:80	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:34200	::ffff:74.125.235.132:80	CLOSE_WAIT
tcp6	0	1	::ffff:172.22.21.155:47008	::ffff:74.125.235.136:443	CLOSE_WAIT

- /proc/net/tcp, udp, tcp6, udp6 is used in netstat command
- After half closed server triggers, later Android's FIN is not acknowledged from server
- The FIN is retried to the app's server about 8 times
- **The server's last ACK for FIN should be sent to Android handset**

No redirection to mobile pages

Request URL: http://www.zdnet.co.kr/news/news_view.asp?article_id=20131015192618

Request Method: GET

Status Code: 200 OK

▼ Request Headers view parsed

GET /news/news_view.asp?article_id=20131015192618 HTTP/1.1

Host: www.zdnet.co.kr

Connection: keep-alive

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;*

User-Agent: Mozilla/5.0 (Linux; Android 4.1.2; SHV-E160S Build/JZO54K) AppleWebKit, like Gecko) Chrome/30.0.1599.82 Mobile Safari/537.36

Accept-Encoding: gzip,deflate,sdch

Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4

▼ Query String Parameters view parsed

article_id=20131015192618

▼ Response Headers view parsed

HTTP/1.1 200 OK

Date: Sun, 20 Oct 2013 10:52:22 GMT

Content-Length: 70703

Content-Type: text/html; charset=UTF-8; Charset=UTF-8

Set-Cookie: ASPSESSIONIDCCTCTCD=CILAKEMDBDNBOBIGOJCHFBPF; path=

Cache-control: private

- HTTP Request should be redirected to mobile page
 - Loading time too long
- But \(\root\)page can be redirected to mobile page



Additional Readings

- 2013 AT&T Developer Summit - Turbocharge Your Mobile App (<http://soc.att.com/W0Kv6h>), Doug Sillars
- Google IO 2013 - Mobile Performance from Radio Up (<http://bit.ly/11fZTNA>), Ilya Grigorik
- High Performance Browser Networking(<http://bit.ly/12kia3y>), Ilya Grigorik
- DevBytes: Efficient Data Transfers(<http://bit.ly/1guF8eR>), Reto Meier