The Impact of Application Signaling Traffic on Public Land Mobile Networks

Yongmin Choi, Cha-hyun Yoon, and Young-sik Kim, KT Corporation
Seo Weon Heo, Hongik University
John A. Silvester, University of Southern California

ABSTRACT

As mobile Internet applications supporting real-time communications services are pervasively used, traffic patterns in mobile networks have changed significantly. New mobile Internet applications differ from traditional applications such as web browser and FTP in that they need always-on connectivity and generate a large amount of signaling traffic. The traffic patterns of these new mobile applications lead to frequent change of radio resource control states between connected and idle in the user equipment, consuming device battery power and causing excessive signaling overhead. The growth in signaling has resulted in a traffic surge that has attracted the attention of mobile network operators. In this article, we first explore traffic composition in a 3G network and show that chatty applications such as instant messenger and social network services contribute to the frequent change of radio states. We then describe the mechanism behind the surge in signaling traffic and describe the disastrous outcome when an application server failure and recovery occurs. Lastly, we discuss some remedies to reduce the application signaling traffic and network load in 3G and LTE networks. These remedies include development of network-aware smart applications that consider the mobile network characteristics and introduction of push notification services. We also present related future work and our research directions.

INTRODUCTION

The widespread use of mobile devices using third-generation (3G) and Long-Term Evolution (LTE) networks has led to the development of various applications that take advantage of the always-on Internet connectivity provided by these networks. Instant messenger (IM) or social network services (SNSs) like Facebook and Twitter are some examples of this class of new mobile applications. Traditional Internet applications, such as web surfing and file transfer, are characterized by a usage pattern that has distinct active and inactive phases. An active phase is a period in which several bursts of packets are transmitted, while an inactive phase is characterized by no data transmission during a sustained time period. The traffic pattern of recent and emerging applications that rely on always-on connectivity is quite different. Since the emerging mobile applications support real-time communications services, they are often constantly running in background mode to receive status updates or messages from other parties. Thus, the applications continuously generate short signaling messages such as keep-alive and ping requests to maintain the always-on connectivity.

Although the traffic volume of keep-alive messages is not large, frequent short messages can incur a large amount of related signaling traffic in the mobile network. In 3G or LTE networks, the user equipment (UE) and radio access networks keep the radio resource control (RRC) states. The UE stays in RRC Connected mode when it transmits or receives data during active periods and stays in RRC Idle mode during inactive periods. To send even a small data packet, the UE changes the RRC state to the RRC Connected mode prior to transmission. This RRC radio state change generates a lot of signaling messages, resulting in a rapid increase in traffic loading. The amount of signaling traffic leads to two major problems: rapid drainage of the mobile device's battery and a signaling traffic surge in the mobile network. In [1], the authors focused on the issues of the energy impact on the mobile device. In this article, we focus on the signaling impact of these applications on public land mobile networks (PLMNs).

The signaling traffic surge, or so-called signaling storm, due to the rapid growth in use of these applications is having a serious impact on mobile network performance. The frequent RRC state change leads to increased signaling overhead over the air interface and through the core elements of a mobile network. The effect of signaling traffic loading gets more severe for the core network as the number of UE devices connected to the core network elements increases. Several mobile network operators (MNOs) have experienced severe service outage or degraded network performance due to the increase of application signaling traffic [2-4]. Furthermore, the stability of the network can also be impacted by signaling traffic when there is an application server failure or outage. If an application server

 0163-6804/14/$25.00

is unexpectedly out of service, all the clients of the server lose their connections and try to restore them. When the server is reinstated, simultaneous reconnection tries from all the clients occur. This phenomenon is similar, in many ways, to a distributed denial of service (DDoS) attack, resulting in a traffic overload in the core network.

This article explores these signaling traffic issues in mobile networks. We first analyze mobile data traffic characteristics based on the type of application and show how the traffic characteristics of emerging mobile applications are responsible for an increase of RRC state change attempts. We then discuss how the frequent RRC attempts impact the network traffic load due to increased signaling messages. We present solutions and standardization efforts to reduce application signaling. Finally, we make some concluding remarks.

Traffic Composition in a WCDMA Network

In this section, we analyze the traffic composition in the commercial wideband code-division multiple access (WCDMA) network of KT Corporation, one of the major MNOs in Korea. We demonstrate that the emerging mobile applications generate a large proportion of HTTP requests compared to their proportion of traffic volume. We also discuss why frequent application signaling is used in the emerging mobile applications.

The traffic data of KT's WCDMA network was collected at the interface connecting the mobile data backbone network and the Internet (G_i interface in Third Generation Partnership Project [3GPP] specification) in August 2011. We used the proprietary network monitoring system, WCDMA Network Traffic Analysis System (WNTAS), for collecting and analyzing mobile data traffic. It collects traffic by mirroring all the links at the G_i interface and then collects traffic log data based on the source/destination IP addresses, port numbers, and transport layer protocols (TCP and UDP). It further analyzes the TCP sessions having port numbers 80 and 8080 as HTTP traffic and records HTTP transactions for each TCP session based on HTTP header information.

At the time of the study, the number of subscribers for WCDMA service was 16 million. The number of smartphone users was 6 million, corresponding to 38 percent of the total subscribers. The monthly traffic volume was 5175 Tbytes. The proportion of HTTP traffic was approximately 80 percent of the total data traffic. Since many smartphone applications are web-based, HTTP traffic constitutes most of the total data traffic. The HTTP traffic is classified according to the type of application identified with the URL field in the HTTP header. To accurately classify the traffic with URL, we keep a database of website information, which is frequently updated and verified. In this article, we classify mobile applications in five categories: media, application download, web surfing, IM, and "other."

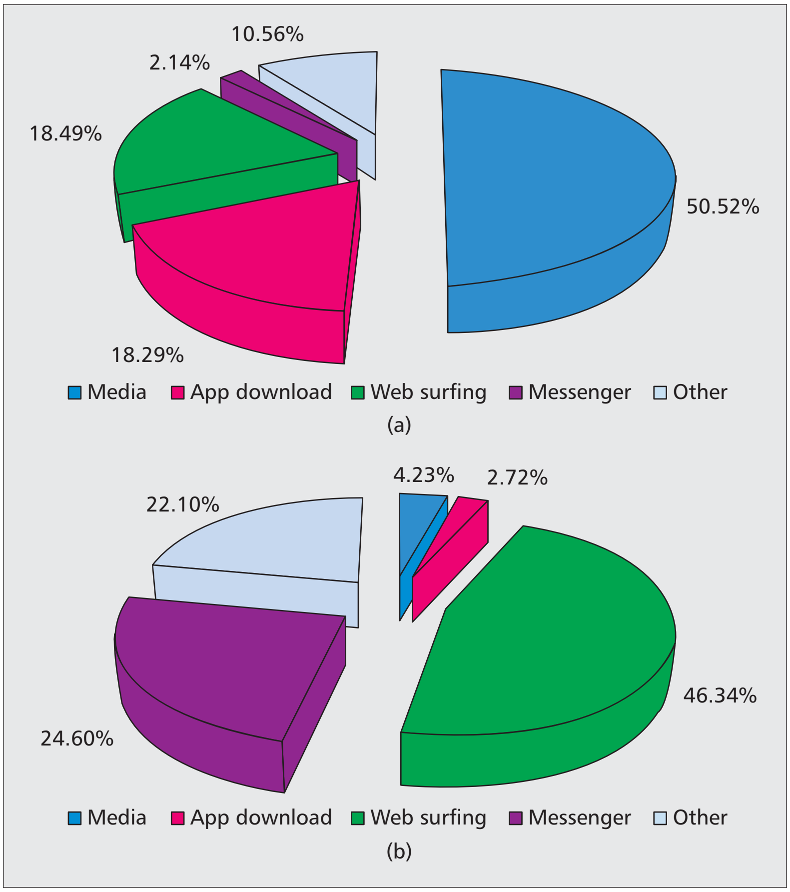

Figure 1. *Traffic composition based on the type of application: a) traffic volume; b) HTTP requests.*

Figure 1 shows the composition of HTTP traffic based on the application type. Media traffic is generated by music and video download/streaming services, which account for a large fraction of the explosive increase in mobile network traffic volume. The application download category refers to the traffic used to download applications from Apple's Appstore, Google Play Store (formerly Android Market), and similar application download sites. This category accounts for the second largest volume of traffic. The web surfing category accounts for the traffic used by browsers such as Apple's Safari, Google's Android browser, and other web browsers. In the IM category, only the traffic related to the most widely used mobile messenger in Korea (KakaoTalk) is included. The remaining traffic is classified as other.

We observe that the large majority of the traffic in mobile networks is related to media streaming and application download services. While the proportions of traffic volume for web surfing and IM are small, the proportions of associated HTTP requests for them are much larger. This phenomenon is consistent with the usage pattern of smartphones: users frequently send short messages, and messenger clients constantly send keep-alive messages to maintain their connections to the application server. While the share of IM traffic volume is only 2 percent, the share in the associated number of HTTP requests is more than 24 percent.

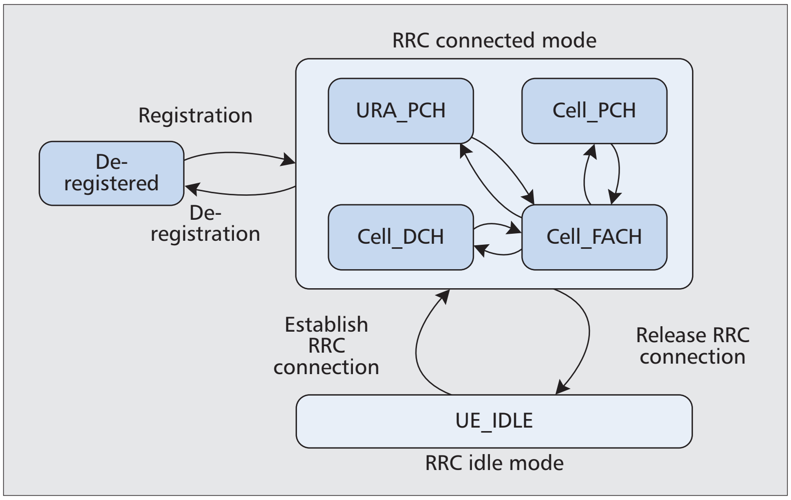

Figure 2. *RRC state diagram in UE or network.*

Although the large increase in mobile data traffic volume has been widely recognized, the traffic load from short (but frequent) signaling messages has lately become a major concern for MNOs, as explained in the following section.

Impact of Application Signaling

In this section, we briefly explain the basic keep-alive mechanisms and review the operation of RRC, which is responsible for assigning radio resources between the UE and the network. We then investigate the message sequences exchanged between the UE and the network to send a short keep-alive message (when in the idle state), and the resulting signaling traffic load in WCDMA or LTE networks.

In many networks, including Internet access networks, Network Address Translators (NATs) or other forms of middle-boxes are deployed. In a system using a NAT, a local network uses IP addresses within a private IP address range, and the local network is connected to the outside Internet with a publicly routable IP address. As the UE sends data traffic from the local network, the NAT translates the local IP address to the publicly routable address by modifying the IP address and TCP/UDP port numbers. The NAT needs to keep the state of the active connections so that inbound packets can be routed to the correct local addresses and applications. The NAT uses expiration timers to remove unneeded connection state table entries, which are recreated whenever there is traffic. If a "chatty" application needs to be accessed from outside the local network, the expiration of connection state during an idle period could be a problem. In order to reset the expiration timers in the NAT, many protocols use a keep-alive mechanism, such as having the UE send a dummy packet on a regular basis when there is no traffic to send.

The RRC states of the UE for packet data transmission are illustrated in Fig. 2. When the UE powers on, it indicates its existence to the network by performing attachment or registration procedure. Roughly speaking, UE can be in either of two different operating modes: RRC connected and RRC idle mode. In connected mode, the radio link is kept active, allowing the UE to transmit or receive traffic. In idle mode, the radio link is released to reduce power consumption. Thus, prior to data transmission the UE must first change state to the connected mode.

To be more specific, the UE can be in one of four RRC states: Idle, PCH (Cell_PCH, URA_PCH), Cell_FACH, and Cell_DCH. In the Idle state the UE does not maintain the RRC connection with the network, so connection must be re-established before any data transmission. In PCH state the UE is RRC connected with the network. No user data can be sent, and the UE only checks the paging information. In Cell_FACH state the UE is in RRC connected mode and can communicate with the network using the common or shared radio channels. This state is ideal when the UE transmits or receives short data packets. In Cell_DCH state the UE is connected to the network with a dedicated channel or a high-speed downlink shared channel (HS-DSCH). This state is ideal for the exchange of large data packets; however, a lot of signaling traffic is necessary to change RRC state from Idle to Cell_DCH as a dedicated channel must be established prior to data transmission.

Application signaling traffic, such as keep-alive messages, causes frequent switching between Idle/PCH state and Cell_FACH/Cell_DCH state. The result is that a significant amount of radio resource is consumed, and there is an increased chance of packet collision on the common channels if several UE devices try to access the server simultaneously. To make matters worse, this frequent switching of RRC states generates increased control signal processing loads in network entities such as the radio network controller (RNC) and serving gateway support node (SGSN).

As discussed in the previous section, the traffic composition in a WCDMA network resulting from signaling traffic from real-time communication services generates a substantial portion of the HTTP request traffic. The increase in signaling traffic is incurred, in part, by the application signaling messages such as keep-alive or ping requests. Based on our measurement with commercial devices, the most popular mobile messenger in Korea (KakaoTalk) sent keep-alive messages every 10 minutes in 2011. There is another mobile application (Stock Radar) that sends more than 120 keep-alive messages per hour. The average number of RRC attempts per smartphone user was 237 times a day in July 2011, while that of a feature phone user was only two times a day.

In summary, pervasive dissemination of smartphones and widespread use of real-time communication applications contribute to the rapid increase of RRC attempts for data calls, as shown in Fig. 3. As opposed to iOS, in which the Apple Push Notification service (APNs) is mandatory for all applications, this phenomenon is more prominent in Google Android smartphones in which each application may maintain its connection individually; so each application installed in UE may send separate keep-alive messages. The total volume of keep-alive messages grows rapidly in proportion as more applications are installed in the UE.

In addition to the amount of signaling traffic, the traffic pattern is also an important factor for the network load. Figure 4 shows the signaling traffic caused by a server failure. In this figure, each curve corresponds to the number of RRC attempts in a SGSN. The sum of all the curves is the number of RRC attempts in KT's WCDMA data network (i.e., packet service [PS] RRC attempts). If an application server is unexpectedly out of service or malfunctions, all the clients of the server lose their connections and try to restore them. When the server is reinstated, simultaneous reconnection tries from all the clients happen. This phenomenon is similar to a DDoS attack, and overload in the core network happens. (We see an example of such a surge of PS RRC attempts between 5 a.m. and 6 a.m. in April 2010 in Fig. 4). Several MNOs have reported network failures or degraded performance arising from application server problems [2–4].

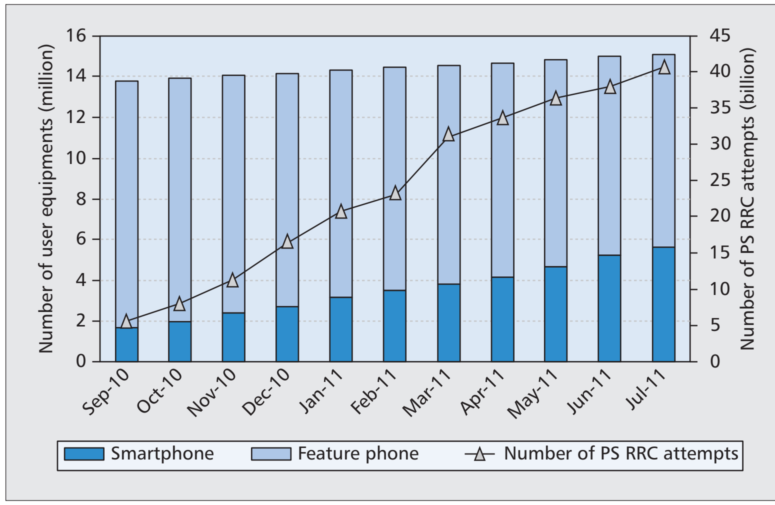

Figure 3. *Relationship between the number of smartphones/feature phones and the number of packet service RRC attempts.*

Efforts on Controlling the Application Signaling Traffic

Compared to fixed broadband wireline networks, wireless networks are characterized by limited bandwidth, higher latency, and non-permanent communication channels. As we have seen, it makes sense to limit or control application signaling traffic to alleviate the radio resource and processing loads in wireless networks. The mobile environment also puts constraints on the resources available to mobile device applications. Ideally, smartphone application developers would consider the characteristics and constraints of mobile environments, but developers are not always familiar with the details of the mobile environment and wireless network issues. This has led to the concept of a *network-friendly* application architecture.

The signaling load in mobile networks is associated with several operational characteristics of the emerging mobile applications [6]. The client of emerging applications receives data or updates from the application server and other parties by either client-driven polling or connection-oriented push methods (e.g., persistent connections with enabling server-initiated data delivery). With the polling approach, the client occasionally connects with its server and downloads data. If the polling interval is too short, unnecessary traffic is generated, whereas too long of a polling interval causes delayed response time.

In the connection-oriented push method, the server sends a push notification to inform the client of receiving data or status updates. The client establishes a persistent connection (e.g., via HTTP, a TCP-based protocol such as XMPP, or other proprietary TCP- based protocols), and either the client or the server regularly sends signaling messages (e.g., keep-alive or ping requests). Sending signaling messages regularly maintains always-on connectivity and prevents expiration of the IP session allocated to the mobile device. Maintaining always-on connectivity facilitates fast receipt of data or updates from the server since the call setup delay to (re-)establish the PS session is not required. However, frequent transmission of signaling messages not only occupies the radio resources but increases the processing load in the WCDMA network.

This section describes related activities to reduce the application signaling traffic. A Korean standards organization and the Global System for Mobile Communications Association (GSMA) have developed some guidelines for application development [7, 8]. In this section, we briefly review the recommendations in these guidelines, discuss the use of a push notification server (or push proxy gateway) to manage connections between UE and application server, and present the effect of an MNO's push notification server in Korea.

Application-Layer Solutions

We first discuss push and polling methods recommended by the Telecommunications Technology Association (TTA), a national standardization body in Korea. MNOs in Korea participated in TTA to establish guidelines for application signaling procedures in 2011 [7].

In the push method, the guideline recommends adjusting the intervals of sending signaling messages appropriately. To keep always-on connectivity to the application server, the client of an application on UE should send keep-alive messages before the IP session of the UE expires. The holding time of an IP session of UE is determined by MNOs. The typical value of session holding time is on the order of an hour. Thus, application signaling traffic can be reduced by adjusting the time interval of sending keep-alive messages to the session holding time provisioned by the mobile network. However, duplicated signaling by multiple applications in a mobile device should also be controlled appropriately.

To avoid network load from simultaneous reconnect attempts from UE devices, the guidelines suggest regulation of the connection restoration method. When the connections between application clients and servers are lost, the clients should try to reconnect to the servers

While Apple mandates the use of APNs for all applications registered at Appstore, Google does not enforce the use of GCM. As might be expected, for a similar application environment, Android based smartphones tend to generate more signaling traffic than iPhones.

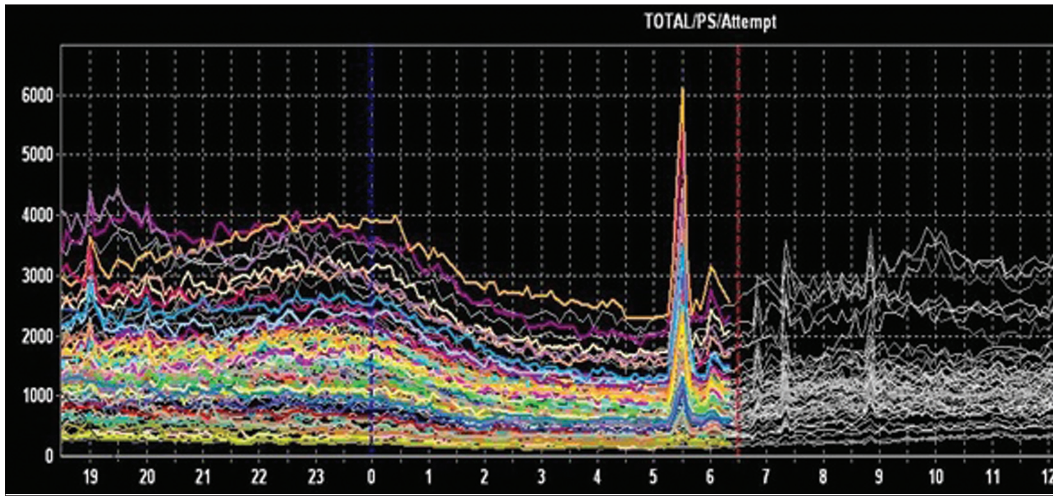

Figure 4. *Snapshot image of the traffic monitoring system showing PS RRC attempt surge.*

at random times. If users make reconnection attempts manually, the guideline suggests not making attempts too frequently.

In the polling method, an application client occasionally connects with its server and downloads data. Since too short polling intervals result in signaling traffic overload, the guideline recommends the following polling methods. First, the default time interval for polling should be at least one hour. Second, to avoid simultaneous polling of many clients, the polling time of each client should be different. Since the email service is considered to require more real-time communication in the guideline, it can use a polling time of less than one hour.

GSMA also provides guidelines for smartphone application developers [8]. It suggests key principles to bear in mind when developing applications for mobile devices. In addition to the application signaling issues we have discussed, it provides detailed tips for developing mobile applications to make them smarter. The guideline describes seven requirements of an ideal mobile application and provides detailed specific tips for Android, iOS, and Windows mobile phones.

Network-Layer Solutions

Since the push method makes a connection only when there are data or status updates, it is more efficient than polling in which periodic connection attempts are made regardless of data or updates from servers. However, to keep always-on connectivity between application clients and servers, it is necessary to send signaling messages on a regular basis. If each application maintains its connection independent of other applications, the signaling messages increase in proportion to the number of applications that are in use in the UE devices. Application connection management can be simplified by introducing a push notification server (or push proxy gateway).

The push notification server maintains a single connection to a UE device and notifies it when there are data or updates to send. Since each client application no longer has to maintain its own session, the signaling traffic is significantly reduced. Figure 5 compares these two push methods: notification by the application server itself (Fig. 5a) and use of a push notification server provided by the network operator (Fig. 5b).

The architecture of push notification by an integrated push server is shown in Fig. 5b. A single connection is maintained between a UE device and the push server regardless of the number of mobile applications installed. Apple's Push Notification service (APNs) or Google's Cloud Messaging (GCM, formerly Cloud to Device Messaging, C2DM) are examples of push servers. While Apple mandates the use of APNs for all applications registered at Appstore, Google does not enforce the use of GCM. As might be expected, for a similar application environment, Android-based smartphones tend to generate more signaling traffic than iPhones.

Since C2DM had limitations on the number of push messages to send for each application and limitations on the supported Android OS versions (2.2 or later), MNOs in Korea deployed their own push servers for Android smartphones starting in 2011. Furthermore, any MNO can attempt to control application signaling by introducing their own push server. To demonstrate the effectiveness of an MNO-based push server, we exemplify SK Telecom's smart push system [9]. SK Telecom commercialized the first MNO-based push server in May 2011. It accommodated up to 7.5 million Android smartphones in December 2011. The total number of Android smartphone users was 8.8 million, and the accommodation ratio was 85.2 percent.

The smart push system provided push notification service to many popular applications in Korea including four IM applications such as KakaoTalk. In order to use SK Telecom's push notification service, the mobile applications were modified to utilize a push user agent provided by SK Telecom. The smart push user agent is programmed to send a keep-alive message every hour. It is reported that the smart push system reduced 75.7 percent of RRC attempts for the four messaging services [9].

Future Directions of Push Notification Service

While the introduction of push servers in MNO helps to alleviate the signaling traffic problem, the problem of duplicate push notification still occurs with Android smartphones, since push notifications can be by an application server, by an MNO push server, or by a GCM server, and it is necessary to maintain connections to several or all of them. The signaling traffic increases as multiple push notification functions are used. Moreover, it becomes complicated to develop smartphone applications if MNOs' push servers have different application programming interfaces (APIs). Application providers therefore prefer to use GCM or their own servers over MNOs' push servers. On the other hand, MNOs prefer to use their own push servers since this gives them greater control over the UE, and, in addition, they can develop value-added services to go along with the push servers.

If MNOs mandate the use of their own push services, application providers need to modify their applications according to the MNO's push APIs. To reduce the burden of application developers, it is desirable to standardize MNOs' (proprietary) push server solutions, including APIs, message formats, functions, and procedures of an MNO's push service. In 2011, KT proposed standardization of the interfaces between application server and push server, push server and push client, and push client and application. The recommendation for standardized interfaces was approved in 2013.

Additionally, it is preferable to establish a unified push notification service specification independent of the mobile operating system to simplify the application development environment. Web application-based push service is also being considered by the World Wide Web Consortium (W3C). Since web applications run on a user's browser, a push notification service can be implemented regardless of the mobile operating system.

Techniques to reduce application signaling at the network layer are also considered by the 3GPP. The Technical Specification Group-Service and System Aspects (TSG-SA) is defining work items on small data and device triggering enhancements (SDDTE) to be included in the Release 12 specification [10]. Small data transfers like signaling messages or frequent device triggering are to be transmitted efficiently with minimal network impact (e.g., signaling overhead, network resources, and delay for reallocation). The proposed solutions are based on the optimized handling of control plane or RRC connections in order to avoid user plane call setup or alleviate non-access stratum (NAS) signaling. These efforts can contribute to reduction of signaling procedures in the core network and radio interface, but modifications to the existing interfaces and nodes will be required [10].

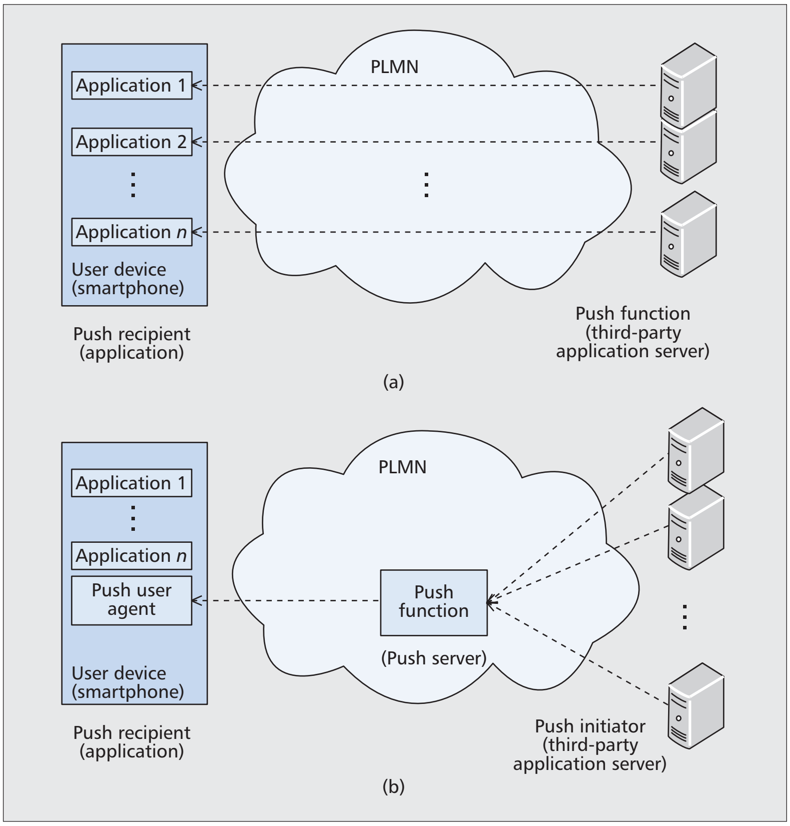

Figure 5. *Comparison of push notification methods: a) push by application server; b) push notification server.*

Concluding Remarks

In addition to the explosive increase in data traffic, mobile network signaling traffic is a major concern for network operators because of the network investment required for processing the increased signaling traffic. As mobile devices such as smartphones and tablets are used more and more, application signaling traffic will become increasingly important in future mobile networks such as LTE. In this article, we have quantitatively explored the traffic composition according to types of mobile applications and showed that signaling traffic mostly arises from applications supporting real-time communication. One example is the increased signaling traffic and associated increased network load to process RRC attempts.

To help resolve the signaling traffic issue, application signaling should be carefully controlled. In this regard, TTA in Korea has developed a recommended specification for the application signaling mechanisms. GSMA has also established guidelines for the development of user- and network-friendly smartphone applications. The signaling traffic load can be significantly remediated by introducing push servers that manage connections between mobile devices and application servers. In addition to the UE operating system vendors' push servers, MNOs have also deployed push servers, resulting in multiple push functions in Android smartphones (with associated increased signaling traffic.) The conclusion is that development of standards for push function APIs and service specifications is desirable. An ideal mix of mobile-operating-system-independent application server and operator based push function support is a work in progress.

The conclusion is that development of standards for push function API's and service specifications is desirable. An ideal mix of mobile operating system independent application server and operator based push function support is a work in process.

Acknowledgments

The authors thank the reviewers and editors for their thorough feedback and suggestions, which were crucial to improving the contents and presentation of the article. The authors also appreciate the participation and contribution of other mobile network operators in Korea, SK Telecom and LGU+, to the standardization of push notification service.

References

[1] M. Gupta *et al.*, "Energy Impact of Emerging Mobile Internet Applications on LTE Networks: Issues and Solutions," *IEEE Commun. Mag.*, vol. 51, no. 2, Feb. 2013, pp. 90–96.
[2] C. Gabriel, "O2 and France Telecom Suffer Severe Outages," *Rethink Wireless*, July 2012.
[3] D. Sahota, "O2 Outage Causes Concern Ahead of Olympics," *Telecoms.com*, July 2012.
[4] C. Gabriel, "VoIP Signaling Crashed NTT DoCoMo; Asks Google to Help," *Rethink Wireless*, Jan. 2012.
[5] "Application of the Radio Access Network Application Part (RANAP) on the E-interface," 3GPP TS 29.108, Rel. 11, v. 11.0.0, Sept. 2012.
[6] "Study on Non-MTC Mobile Data Applications Impacts (Release 12)," 3GPP TS 22.801, Rel. 12, V12.0.0, Dec. 2011.
[7] "Push Notification Service for Public Land Mobile Networks," Telecommun. Technology Association, Dec. 2011.
[8] "Smarter Apps for Smarter Phones," GSM Association Smarter Applications Project, Feb. 2012.
[9] H. Kim, "Smartphone Applications and Network Efficiency," GSMA Executive Management Committee(EMC) meeting #100, Barcelona, Spain, Feb. 2012.
[10] "Machine-Type and other Mobile Data Applications Communications Enhancements (Release 12)," 3GPP TS 23.887, Rel. 12, V0.9.0. Apr. 2013.

Biographies

YONGMIN CHOI [M] (yongmin.choi@kt.com) is a director of the Access Network Construction Department for Network Design Business Unit at Network Group, KT Corporation, where he is responsible for the planning and deploying of wireless access networks including LTE femtocell and WiFi. His research interests include practical issues of mobile network deployment and statistical methods of traffic anomaly detection and control incorporating big data analysis techniques. He is a member of Eta Kappa Nu (the Electrical Engineering Honor Society in the United States).

SEO WEON HEO [S'98, M'06] (seoweon.heo@hongik.ac.kr) received B.S. and M.S. degrees in electronics engineering from Seoul National University, Korea, in 1990 and 1992, respectively, and a Ph.D. in electrical engineering from Purdue University, West Lafayette, Indiana, in 2001. From 1992 to1998, he was with the Digital Media Research Laboratory, LG Electronics, Korea. From 2001 to 2006, he worked at the Telecommunication R&D Center, Samsung Electronics, Korea. Since 2006, he has been an associate professor with the School of Electronic & Electrical Engineering, Hongik University, Seoul, Korea. His current research interests are in the area of signal processing, wireless communication, and embedded system design.

JOHN SILVESTER [SM] (silvester@usc.edu) is a professor of electrical engineering at the University of Southern California (USC). His research and teaching interests include computer communication networks, broadband networks, optical networks, and computer system performance evaluation. He served as Vice Provost for Scholarly Technology at USC from 1994 to 2005 and is currently President-Elect of the USC Academic Senate. He was the chair of the Board of the Corporation for Educational Network Initiatives in California (CENIC) from 2000 to 2006, and has served on the National LambdaRail (NLR) Board and the Internet2 National Planning and Policy Advisory Committee. He is also a member of the ACM.

CHA-HYUN YOON (chahyun.yoon@kt.com) is a senior vice pesident and the head of Network Design Business Unit at Network Group, KT Corporation. He received his M.B.A. degree from Korea Advanced Institute of Science and Technology (KAIST), Seoul, Korea. He holds two professional engineer licenses in electrical and information communications.

YOUNG-SIK KIM (kyoungsik.kim@kt.com) is a senior vice president and head of the Network Operations Business Unit in the Network Group, KT Corporation. He received his M.S. degree in computer engineering from Hannam University, Daejeon, Korea, and his M.B.A. degree from Helsinki School of Economics and Business Administration, Finland.