Facing the Signaling Storm: A Method with Stochastic Concept

Hong Zhang, Kaiping Xue*, Peilin Hong
The Department of EEIS, University of Science and Technology of China, Hefei, Anhui 230027 China
*kpxue@ustc.edu.cn

***Abstract*—To keep "Always-on", terminals contact with remote servers by sending the keep-alive messages (or heartbeat messages). Any message in data plan can be sent if and only if the RRC (Radio Resource Control) connection[1] is built. Keep-alive messages are transmitted periodically, which leads to periodic establishment of RRC connections in some cases. High frequency of state transition may induce heavy burden in control channel in both UMTS and LTE network. Besides, it may also induce low efficiency in data channel in UMTS network. This paper presents a novel scheme to deal with these problems. Two new concepts, establishment probability "*p*" and "Rejected" state, are introduced to reduce the frequency of RRC connection establishment. Hence, the load in channels, especially in control channel, can be reduced to a low level. An analysis is developed based on queuing theory for the scheme and the performance is verified with simulations.**

I. Introduction

With the development of mobile network, Internet can be accessed ubiquitously. Via smart phones or other smart terminals, people can enjoy network service anywhere at any time. For example, they can contact with friends with popular instant messagers, such as Skype, MSN, QQ, and they can also do direction query with Google map.

However, mobile network does not exactly meet the requirements of portable applications(APPs), especially SNS (Social Network Softwares). It is mainly because of the keep-alive scheme in APPs which aims at making the APPs "Always-on". According to the keep-alive scheme, APPs work in the background of smart terminals, submit their local information via keep-alive (KA) messages and receive push service messages. Researchers found that, for instant messager, chatting messages are only a small percent of the total traffic while the larger part of the traffic is caused by massive amount of status update messages[1].

It is easy to achieve "Always-on" in the wired network or via WiFi because the channel resource usage is based on competition. However, in mobile network, such as GPRS/EDGE, UMTS and LTE, radio resource usage is depended on dynamically allocation[2–4]. Extra control signaling is exchanged between terminals and the network to sustain data communication.

In UMTS, the RRC states contain IDLE, CELL_FACH, URA_PCH, CELL_PCH and CELL_DCH. The last four are in connected mode, which means the terminal has established an RRC connection (i.e. resource to bear communication task) to RAN(Radio Access Network). In LTE, there are two RRC states, named RRC_IDLE and RRC_CONNECTED. Only in the connected state, the wireless network is ready to transfer data[5, 6]. The establishment and release of RRC connection leads to state transitions.

Keep-alive messages lead to frequent transition of network states. KA messages are sent by terminals frequently and periodically. The intervals of keep-alive messages may be widely dispersed in different APPs[7]. The interval between two successive KA messages in a specific APP depends on the requirements of the APP itself, which varies from several seconds to several minutes[8, 9]. The RRC connection will be released soon if no more data is to be transmitted. Hence, when a KA message is to be sent, RRC connection is rebuilt with high probability, leading to repeated and frequent state transition. Therefore, the RRC connections between terminals and network side are established repeatedly and frequently due to keep-alive scheme.

Considering the characteristic of the KA messages, the keep-alive scheme has some negative impacts on mobile network.

Firstly, transmission of KA messages leads to frequently state transition when multiple APPs work in the background concurrently, which results in **signaling storm** in the control channel[10, 11]. Nowadays, the same amount of messages will consume much more signaling than before. Data growth and signaling growth in a live network in Western Europe is recorded between December 2009 and July 2010. During the period, while data volume grew 65% and the signaling volume grew 177%[12].

Secondly, the packet size of KA message is small[13], so the transfer duration is extremely short. However, in UMTS, after finishing transmission, the terminal keeps RRC connection for a few seconds(inactivity timer) before the RRC connection is released, during which no data is transferred. Therefore, it leads to low resource utilization in data channel[14]. In LTE network, shared data channels are adopted, so inactivity timer's effect can be neglect. However, the overload of control signaling even influences the core network because of the flatted architecture[14].

In a word, the growing rate of the amount of the signaling is much higher than the amount of data messages. Compared

[1] Actually, in UMTS, RRC connection must be built before starting transmitting data. However, in LTE, S1-connection is also necessary for data transferring, but RRC connection is completed prior to S1-connection establishment. Hence, only RRC connection is emphasized here.

978-1-4799-7339-2/14/$31.00 ©2014 IEEE

with regular data, sending KA messages will consume much more signaling and channel resource, which affects the performance of the mobile network.

In this paper, a new scheme with stochastic concept is proposed, which requires only the terminals to be modified. Two new concepts, named **establishment probability** $\boldsymbol{p}(0 < p \leq 1)$ and **Rejected state**, are introduced. When in idle state, the RRC connections between terminals and network are established with probability $\boldsymbol{p}$ if new KA messages are to be sent. If the terminal fails to trigger to build the RRC connection, the KA messages are stored and wait to be transmitted via subsequent RRC connection. After all the data is transmitted and the RRC connection is released, the terminal turns to rejected state. A terminal mustn't build the RRC connection within its own rejected state. The terminal leaves the rejected period if the rejected timer expires. The terminal can request to build the RRC connection freely after existing the rejected state.

The main feature is that several KA messages stored in the terminal can be transmitted aggregately. The benefits are obvious. Firstly, it decreases the frequency of connection establishment, so the load on the control channel can be effectively reduced. Secondly, it avoids affecting the users' QoE (Quality of Experience) heavily because the average waiting time of KA messages is limited by carefully choosing proper parameters.

The rest of the paper is organized as follows. Section II provides a brief introduction to related work. The new scheme is given in Section III. In Section IV, a guidance of the parameter settings is presented based on theoretic analyses. Simulation results and performance comparisons are given in Section V. Section VI concludes the paper.

II. Related Work

Considering energy consumption of terminals and resource allocation in wireless network, keeping connections alive all the time is impractical. The continuity of communication can be indirectly kept by setting up RRC connections periodically and transmitting KA messages. It is important to trade off among energy consumption, radio resource utilization and QoE.

In [4], focusing on the RRC connection release phase in LTE network, the scheme presented the concept of "activeness" in the system. The activeness was taken into consideration when the connections were to be released. In [15], it was proved that the frequency of RRC connection requests dropped dramatically with the increase of the duration of the inactivity timer. However, longer active duration leads to inefficient energy consumption in terminals and low efficiency in data channels.

Fast Dormancy[2] keeps a low energy consumption in terminals, but it causes frequent establishment of RRC connections which increases the network signaling load[5, 16]. To address this problem, in 3GPP Release 8, terminal adds releasing reason to SCRI and set it as "UE Requested PS Data Session End". Hence, whether the connection is to be released or not is depended on both the terminal side and the network side.

The above focus on modifying the terminals, while another method is to modify the network side. The RRC connection establishment can also be triggered by the network side, such as push services. Centralized model for push service can prevent signaling storm. It is adopted by Apple Push Notification Service (APNS) for IOS[17]. In this model, messages of push service (e.g. e-mail and news) from remote servers are firstly aggregated at proxy server, and then delivered to the terminals. Hence, the connection only need to be built between the proxy server and the terminals. However, long delay in centralized model may affect QoE.

III. The New Scheme with Stochastic Concept

To decrease the frequency of state transition, it should either prolong the intervals of the KA messages or send more KA messages every time the RRC connection is built(i.e. send more KA messages per RRC connection). However, in the first way, prolonging the intervals leads to updating information less timely, which could deteriorate QoE. Moreover, the efficiency is still low because almost every transmission of KA messages will consume much control signaling to build RRC connection, which has no performance enhancement compared with the original scheme. In the second way, it is difficult to coordinate the transmitting timing of KA messages among different APPs.

To prevent the signaling storm as well as ensure high QoE, our scheme focuses on transmitting the KA messages aggregately. Firstly, several successively generated KA messages can be transmitted via a single RRC connection. It reduces the frequency of state transition. The KA messages can be generated at the normal intervals as specified by the application providers. Secondly, the efficiency is improved. The signaling and channel resource, which were only used by a single KA message in the original scheme, are consumed by several KA messages here. Finally, we take advantage of stochastic concept to improve the performance on average, which does not require any coordination among different APPs.

Establishment probability $\boldsymbol{p}$ is introduced. When a terminal is in idle state, it will request to establish an RRC connection between itself and the network with establishment probability $\boldsymbol{p}(0 < p \leq 1)$ if a new KA message is generated. If what need to be transmitted is not KA messages, then $\boldsymbol{p}$ can be set as 1. Whether the terminal requests to establish a connection or not, and how many KA messages can be transmitted via the connection is depended on $\boldsymbol{p}$. The terminal judge the type of messages to be sent and set the value of $\boldsymbol{p}$ accordingly. Every time when a KA message arrives, a random number $Nu(Nu \in [0, 1])$ is generated. If Nu is less than or equal to

[2] If there is no data in a few seconds, terminal will tear down the connection between itself and network. Terminals send RRC SCRI (RRC Signaling Connection Release Indication) to notify the network side that connection is torn down.

p, the terminal will request to establish an RRC connection to bear KA messages; otherwise, the message is stored.

The other new concept is “Rejected” state. The RRC state turns to “Rejected” state immediately after RRC connection is released. The terminal mustn’t set up RRC connection in its own rejected period. In our proposed scheme, there are several problems to be further addressed:

- How to differentiate the regular data and KA messages
- What is the modified RRC state machine
- How to dynamically adjust ***p*** and the Rejected duration
- What impact will the TCP timer has on the scheme

According to statistical analysis, the KA messages are approximately periodical. Therefore, the message interval and packet size are closely related[14]. The type of content can be inferred from the intervals of messages without checking detailed data in packets. If a KA message arrives after a relatively fixed interval, then it can be inferred as the KA message. If messages of a specific APP arrive successively, they are generated by user behavior and all of them must be transmitted immediately. The KA messages stored in the terminal should keep the latest information. If a newer version is generated before the old ones of the same APP are sent out, then the old ones are abandoned.

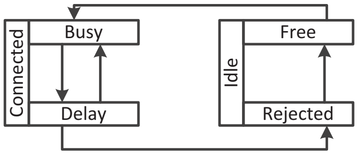


Fig. 1. RRC state machine between the terminal and the network

The RRC state machine is used to describe the connection state between the terminal and the network. In Fig.1, four states(named “Busy” state, “Delay” state, “Rejected” state and “Free” state) are supported in our scheme. “Busy” state and “Delay” state are the same as the connected mode in UMTS or LTE, in which the RRC connection is successfully built. In the other two states, the connection is released.

- “Busy” state: Data is transmitted. The duration is depended on the volume of data.
- “Delay” state: The inactivity timer is started. No data is transferring and RRC connection is waiting to be released. Any newly generated data of the same terminal can turn the state to “Busy” state again.
- “Rejected” state: The terminal cannot build RRC connection in its own Rejected period. This state has no effect on messages generated by users.
- “Free” state: RRC connection can be built freely with probability ***p***. The duration is depended on the probability ***p*** and the arrival rate of messages.

The duration of “Delay” state and “Rejected” state is fixed, and they are all implemented by timers. If the terminal is in “Free” state and KA messages are to be sent, it firstly request to build the RRC connection with probability ***p***. If the connection is successfully built, the state turns to “Busy” state. When in the “Busy” state, if there are no more data to be transmitted, the inactivity timer is started and the state turns to “Delay” state. After the timer expires, RRC connection is released and the terminal turns into “Rejected” state. A new timer, named “Rejected timer”, is introduced to the terminal. Before the timer expires, the terminal is in its “Rejected” state, and it mustn’t build the RRC connection. The “Rejected” state is to prevent the same terminal from rebuilding RRC connections within a short time.

To keep a high QoE, the value of ***p*** and the duration of “Rejected” state can be adjusted according to the status of APPs in the background. For example, if there are few applications generating KA messages or the intervals of KA messages are large, the value of ***p*** shall be increased to avoid long waiting time of KA messages stored at the terminal. More detailed considerations are in Section IV and Section V.

The last problem is what impact will the TCP timer has on the scheme. Some of the KA messages are TCP based. If the connection is not successfully built, KA messages are stored in the terminal. The question is that whether the scheme will leads to unnecessary timeout in TCP layer. Fortunately, timeout will not happen with high probability even though the KA messages are not transmitted immediately. It is because that timeout interval is based on measured round trip time (RTT), and storing the KA messages in the terminals leads to longer RTT. The sender sets a timer with longer timeout interval when sending a KA message. Therefore, timeout scheme of TCP protocol will not be effected.

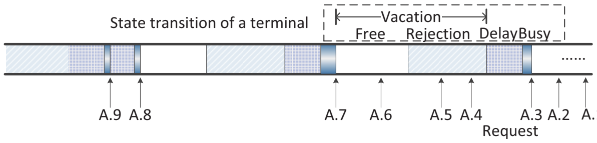


Fig. 2. An example of state transition triggered by keep-alive messages

An example of state transition is shown in Fig.2. Let A.x denote the transmission requests of KA messages generated by terminal “A”. When A.x arrives, $Nu(Nu \in [0, 1])$ is generated correspondingly if new RRC connection is required to be built.

It is assumed that the KA messages here are the latest ones. A.1 and A.2 are not permitted to request to build RRC connection because the corresponding values of Nu are both bigger than ***p***. On the contrary, when A.3 arrives, RRC connection is established successfully(because $Nu \leq p$), and all the 3 KA messages are transmitted in the “Busy” duration if they belong to different APPs. Then after inactivity timer is timeout, the state of terminal “A” turns to “Rejected” state. The A.4 and A.5 arrive at the “Rejected” state, so the RRC connection mustn’t be built. Similar to the A.1 and A.2, A.6 arrives in the “Free” state but fails(because $Nu > p$) to trigger to build RRC connection. The KA messages can be transmitted via the connection requested by A.7(the Nu for A.7 satisfies $Nu \leq p$). The KA message with A.9 can be transmitted immediately because it arrives at the “Delay” state of “A” and the connection of the terminal is still active.

TABLE I

VARIABLE DESCRIPTION

name	description
Q_v	the extra queue length(i.e. the KA messages stored in the terminal) caused by vacations
W_v	the extra waiting time caused by vacations
T_i	the interval between $(i-1)^{th}$ and i^{th} KA messages
λ	the average arrival rate of the KA messages generated by all the APPs in the terminal
T_d	the transmission delay, which is the duration that the message is transmitted through the interface
T_D	the duration of "Delay" state
V_N	the duration of vacation, during which the number of newly generated KA messages is N
K	the number KA messages sent in "Busy" state
T_r	the duration in "Rejected" state
$E(Q_v)$	the average number of stored KA messages induced by vacation
$E(W_v)$	the average waiting time induced by vacation
$E(V)$	the average vacation time
$E(V^2)$	the mean square of vacation time
N	the number of KA messages generated in vacation

IV. Theoretical Analysis

The majority of RRC connection establishment is caused by KA messages rather than user behavior[11, 14]. Hence, focusing on the KA messages, we analyze the performance of our scheme using queuing system with vacations. Vacations here mean the "Rejected" state and the "Free" state. The notations used in this paper are given in Table I.

A. Network Model and Problem Formulation

A queuing system M/G/1 with vacations[18] is adopted here. According to our scheme, KA messages will be stored in the terminal if RRC connection is failed to be built. Hence, the queue length is used to model the number of KA messages stored in the specific terminal. The average delay means the average duration between the time point that KA messages are generated and the time point that KA messages are sent out. Several APPs coexist in one terminal and each of them sends KA messages independently with different intervals.

There are several assumptions. Firstly, only the extra queue length(Q_v) and extra delay(W_v) induced by "Rejected" state and "Free" state are analysed. The queue is infinite. Secondly, the arrival rate is total arrival rate of different APPs in a terminal. Referring to[19], the arrival interval obeys the exponential distribution with average interval of $1/\lambda$.

The vacation time is $V_N = \sum_{i=1}^{N} T_i - K \cdot T_d - T_D$. The value of T_d is much smaller than T_i and T_D. The feasible value of K is also limited to avoid long waiting time of KA messages. Hence, the effect of $K \cdot T_d$ can be neglect, the vacation time is $V_N = \sum_{i=1}^{N} T_i - T_D$ for a specific N.

B. Model without Rejected Time

Here, we first discuss the situation where the rejected time T_r is zero(No "Rejected" state). The "Free" state is the "Idle" state. When a new KA message arrives at the "Free" state, the terminal either requests to establish a connection with probability p or buffers the KA message with probability $(1-p)$. Hence, the probability to turn to "Busy" state obeys the geometric distribution, shown in equation(1). The average value and the mean square value of the vacation is obtained in equation (2) and (3).

$$P(N=n) = p \cdot (1-p)^{n-1} \tag{1}$$

$$\begin{aligned} E(V) &= E\{E(V_N|N)\} = \sum_{n=1}^{\infty} E(V_n|N=n)P(N=n) \\ &= \sum_{n=1}^{\infty} (\frac{n}{\lambda} - T_D) \cdot p \cdot (1-p)^{n-1} \\ &= \frac{1}{\lambda p} - T_D \end{aligned} \tag{2}$$

$$\begin{aligned} E(V^2) &= E\{E(V_N^2|N)\} = \sum_{n=1}^{\infty} E(V_n^2|N=n)P(N=n) \\ &= \sum_{n=1}^{\infty} (\frac{n+n^2}{\lambda^2} + T_D^2 - \frac{2nT_D}{\lambda}) \cdot p \cdot (1-p)^{n-1} \\ &= \frac{2}{\lambda^2 p^2} + T_D^2 - \frac{2T_D}{\lambda p} \end{aligned} \tag{3}$$

Finally, the average number of extra KA messages $E(Q_v)$ is obtained in equation (4) according to [18]. Extra waiting time is $E(W_v) = E(Q_v)/\lambda$. $E(Q_v)$ denotes the average number of KA messages transmitted when RRC connection is built. Given that, the average interval of two successive KA messages is about several hundred seconds. The value of λ is small and the duration of "Delay" state is not too long. Hence,the approximation of $E(Q_v)$ is in equation (4) based on above conditions. What we can learned is that, reducing the frequency of transmission of KA messages without causing long delay can be achieved by choosing a proper value of p.

$$E(Q_v) = \frac{\lambda \cdot E(V^2)}{2E(V)} =\approx \frac{1}{p} - \frac{\lambda T_D}{2} \tag{4}$$

C. Model with Rejected Time

Rejected time is introduced to avoid the channel being occupied by the same terminal frequently within a short time. After the RRC connection is released, the terminal cannot apply for the resource again in its "Rejected" state. It is assumed that, after the RRC connection is released, the total number of KA messages arriving in "Idle" state is N; the number of KA messages arriving in "Rejected" state and "Free" state is $N(T_r)$ and $N - N(T_r)$ respectively. Only the KA messages arriving in "Free" state can turn the state to "Busy", which is the primary difference compared with the model in IV.B. The probability to turn to "Busy" state obeys the geometric distribution as shown in equation(5).

$$\begin{aligned} P(N=n) &= \sum_{m=0}^{n-1} P(n-m|N(T_r)=m) \cdot P(N(T_r)=m) \\ &= \sum_{m=0}^{n-1} p(1-p)^{n-m-1} \cdot P(N(T_r)=m) \end{aligned} \tag{5}$$

D. Analysis

Fig.3 presents a numerical result of $E(Q_v)$. The rejected time in two scenarios is set to 0s and 150s respectively. The duration of “Delay” state is 5s. In Fig.3, it presents an intuitive feeling of the variation trend of $E(Q_v)$ when the value of $\boldsymbol{p}$ varies. Obviously, there is no optimal value for $\boldsymbol{p}$. To achieve that the channel is shared by more APPs, it is necessary to buffer KA messages as many as possible in the terminal, which means the queue need to be as long as possible. The lower value of $\boldsymbol{p}$ leads to more KA messages being transmitted in one RRC connection, and lower frequency of RRC connection establishment. However, if the $\boldsymbol{p}$ is too low, the QoE will be seriously damaged due to the longer waiting time. By limiting the maximum number that shall be transmitted through an RRC connection, or limiting the average waiting time, the value of $\boldsymbol{p}$ can be decided.

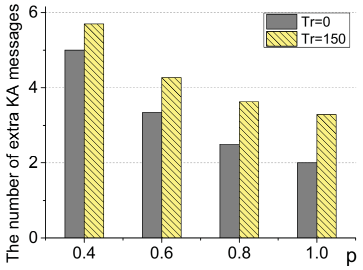


Fig. 3. Result of the analysis of $E(Q_v)$

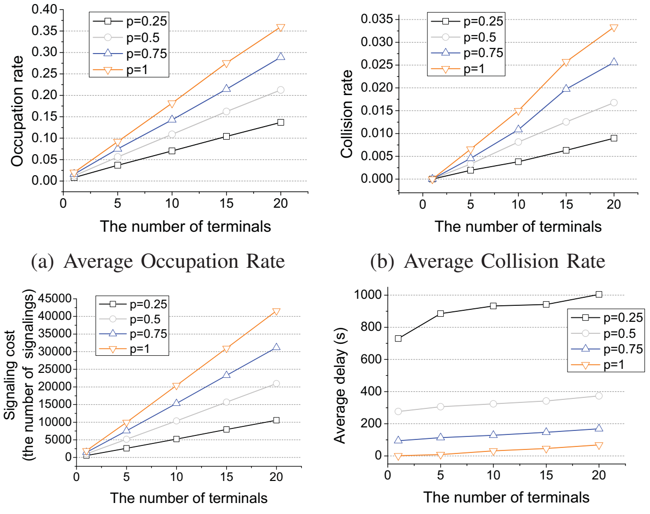
V. Performance Evaluation

In this section, the new scheme is verified in Matlab. The radio resource here is represented by a logical resource block which is requested by several terminals. Because the majority of the messages are generated by the keep-alive scheme, only KA messages are considered here. For a scenario with specific parameters, the simulation is run several times and the results are averaged. Detailed metrics are shown below.

- Data load: Resource utilization in data channel, including the occupation rate and the collision rate of the channel.
- Control load: Resources utilized in control channel, including the number of generated signaling.
- Average delay: The duration measured from the time point when the KA message is generated to the time point when the message arrives is sent out. The delay has an effect on the QoE.

Generally, the duration in “Delay” state is set as 5s. Each terminal generates 160 keep-alive messages. The messages are generated according to the Poisson process. The average interval of the keep-alive messages, the rejected duration and the establishment probability $\boldsymbol{p}$ are variables. The influence of the parameters is analysed later.

Firstly, the overall performance of the new scheme is verified, including data load, control load and average delay. The effect of the value $\boldsymbol{p}$ is emphasized. In Fig.4, it is proved that the new scheme can significantly reduce the load in data channel and control channel. As shown in Fig.4(a), Fig.4(b) and Fig.4(c), it is obvious that lower value of p leads to lower collision rate, lower occupation rate, lower control signaling, higher number of KA messages sent per RRC connection and lower frequency of state transition. In a word, the signaling storm can be relieved by adopting our scheme.



(a) Average Occupation Rate (b) Average Collision Rate

(c) Total Signaling Cost (d) Average Delay

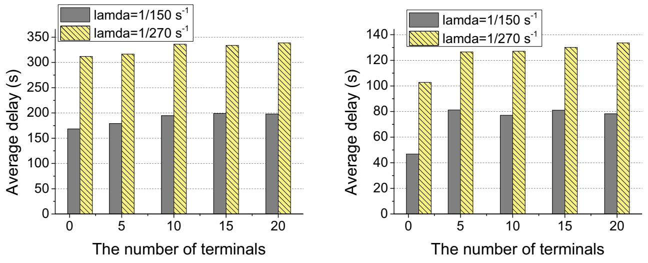
Fig. 4. The performance of the scheme with $\lambda = 270s, Tr = 90s$

However, the value of $\boldsymbol{p}$ cannot be too small. In Fig.4(d) , if the value of $\boldsymbol{p}$ is too small, e.g $\boldsymbol{p}= 0.25$, the average delay becomes much longer, which has a negative effect on QoE. Hence, the value of $\boldsymbol{p}$ need to be carefully chosen. The value of $\boldsymbol{p}$ is suggested to be medium. Detailed consideration on the limitation of the average delay is discussed below.

Secondly, to keep a high QoE, the impact of the parameters on the average delay of the KA messages are discussed, including the probability $\boldsymbol{p}$(in Fig.4(d)), the arrival rate of all the KA messages in the terminal(in Fig.5), and the duration of “Rejected” state(in Fig.6).

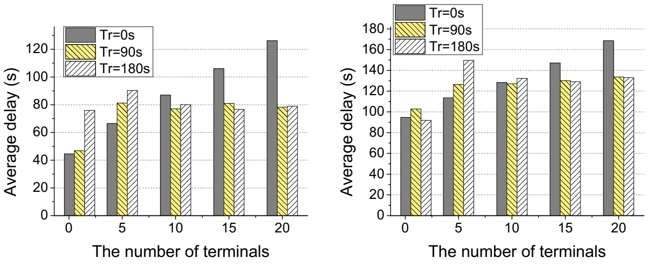
In Fig.5, it can be obtained that the average delay is influenced by the average arrival intervals of messages and the value of $\boldsymbol{p}$. The value of $\boldsymbol{p}$ affects the number of KA messages transmitted in an RRC connection. If $\boldsymbol{p}$ is fixed, then larger average interval leads to larger delay of messages, as shown in Fig.5(a) or Fig.5(b). The reason is that, according to the models in Section IV, the average delay is positively related to the average arrival intervals $1/\lambda$. Therefore, the arrival intervals of messages in different APPs, as well as the number of APPs running in the background, need to be considered when choosing the proper parameters. For example, if there is only one APP communicating with the remote sever, then the value of $\boldsymbol{p}$ need tend to one to keep delay in a low level.

The average delay is also influenced by the duration of “Rejected” state, shown in Fig.6. It is because that, longer rejected duration leads to low frequency of RRC connection establishment; KA messages have to wait longer before being sent out. What is also to be noted is that, with the increase



(a) Average delay with $p = 0.5$ (b) Average delay with $p = 0.75$

Fig. 5. Average delay with $Tr = 90s$



(a) Average delay with $\lambda = 150$ (b) Average delay with $\lambda = 270$

Fig. 6. Average delay with $p = 0.5$

of the number of terminals, the average delay does't change much. It is because that the higher number of terminals leads to less probability of successive requests from the same terminal in a short time, which weakens the effect of rejected duration.

Therefore, it is suggested that the value of $\boldsymbol{p}$ and rejected duration "T_r" need to be adjusted dynamically according to the status both at user side or network side. For example, larger arrival interval requires larger $\boldsymbol{p}$ and larger rejected duration. Higher amount of terminals requires the rejected duration to be shorter to admit free competition.

VI. Conclusion and Future Work

In this paper, focusing on preventing heavy signaling load in control channel and low efficiency in data channel, a new scheme is proposed that RRC connection is established with probability $\boldsymbol{p}$ to make more keep-alive messages be transmitted via one RRC connection. Feasible parameters are analyzed with queuing theory and simulation. It is obtained that signaling load can be significantly reduced. Although the average delay of KA messages in our scheme is a little longer than the original scheme, it can be limited to the acceptable degree by choosing proper parameters. In the future, the parameters, such as probability $\boldsymbol{p}$ and rejected duration T_r, need to be carefully designed, which may be depended on the number of terminals, the load in the network, QoE and so on.

Acknowledgements

This work is partly supported by the National Natural Science Foundation of China under Grant No. 61170231, No. 61379129, and Intel ICRI MNC.

References

[1] Z. Xiao, L. Guo, and J. Tracey, "Understanding instant messaging traffic characteristics," in *Proceedings of the 27th International Conference on Distributed Computing Systems*. IEEE, 2007, pp. 51–51.

[2] L. Ma, "Analysis of network behavior of internet application in the gprs network and simulation," Master's thesis, Beijing University of Posts and Telecommunications, 2013.

[3] H. Haverinen, J. Siren, and P. Eronen, "Energy consumption of always-on applications in wcdma networks," in *Proceedings of the 65th Conference on Vehicular Technology Conference*. IEEE, 2007, pp. 964–968.

[4] H. Zhou, "The research of signaling storm in lte system," Master's thesis, Beijing University of Posts and Telecommunications, 2012.

[5] *New Wireless Broadband Applications and devices: Understanding the Impact on Networks*, 4G Americas report., 2012.

[6] *Taming Signaling: Addressing the Signaling Storm*, Openet Labs Technical White Paper, 2012.

[7] *LTE RAN Enhancements for Diverse Data Applications*, 11th ed., 3GPP TR 36.822 Technical Specification Group, 2012.

[8] S. K. Baghel, K. Keshav, and V. R. Manepalli, "An investigation into traffic analysis for diverse data applications on smartphones," in *Proceedings of 2012 National Conference on Communications(NCC)*. IEEE, 2012, pp. 1–5.

[9] *Behavior analysis of smartphone*, Huawei Smart Lab, 2012.

[10] X. He, P. P. Lee, L. Pan, C. He, and J. C. Lui, "A panoramic view of 3g data/control-plane traffic: mobile device perspective," in *NETWORKING 2012*. Springer, 2012, pp. 318–330.

[11] L. Qian, E. W. Chan, P. P. Lee, and C. He, "Characterization of 3g control-plane signaling overhead from a data-plane perspective," in *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. ACM, 2012, pp. 325–332.

[12] *Understanding Smartphone Behavior in the Network*, Nokia Siemens Networks Smart Labs, 2011.

[13] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 281–287.

[14] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and H. Zhang, "Understanding the nature of social mobile instant messaging in cellular networks," *Communications Letters*, pp. 1–4, 2013.

[15] Z. Zhang, Z. Zhao, H. Guan, D. Miao, and Z. Tan, "Study of signaling overhead caused by keep-alive messages in lte network," in *Proceedings of the 78th Conference on Vehicular Technology Conference (VTC Fall)*. IEEE, 2013, pp. 1–5.

[16] *Fast Dormancy Best Practices*, GSM association network efficiency task force, 2010.

[17] "Local and push notification programming guide," http:https://developer.apple.com/library/IOS/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html.

[18] B. Doshi, "Queueing systems with vacationsła survey," *Queueing systems*, vol. 1, no. 1, pp. 29–66, 1986.

[19] *GERAN study on mobile data applications*, 0th ed., 3GPP TR 43.802 Technical Specification Group, 2013.