# Quality of Service Support for High Performance Computing on Mobile Devices

Ramneek
Korea University of Science and Technology,
Korea Institute of Science and Technology Information,
Daejeon, Korea
ramneek@kisti.re.kr

Patrick Hosein
The University of the West Indies
St. Augustine, Trinidad and Tobago
patrick.hosein@sta.uwi.edu

Wonjun Choi
Korea University of Science and Technology,
Korea Institute of Science and Technology Information,
Daejeon, Korea
cwj@ust.ac.kr

Woojin Seok
Korea University of Science and Technology,
Korea Institute of Science and Technology Information,
Daejeon, Korea
wjseok@kisti.re.kr

***Abstract*—High Performance Computing (HPC) over mobile devices has emerged as a key area of research interest owing to the need for high computing capabilities even when users are on the move. Cellular device technology has evolved to the point where powerful computing features such as high performance processors and large flash memory are available at relatively low cost. However, to use such devices for performing HPC tasks or as interfaces to access remotely existing HPC facilities, several challenges must be addressed. Some of these include development of light-weight applications for the mobile environment, optimization of power consumption, need for high performance and fault-tolerant networks, etc. Such applications also require stringent Quality of Service (QoS) communication and hence a large amount of network resources. In this paper we propose a framework for provisioning high QoS for supporting HPC services and applications over existing wireless mobile networks. In addition we briefly discuss how proper charging and resource allocation can be used to balance resource consumption and the QoS provided.**

***Keywords-HPC; Mobile Computin; QoS; Pricing; Resource Allocation.***

## I. INTRODUCTION

High Performance mobile computing involves the convergence of emerging mobile wireless communication networks, intelligent and powerful mobile devices and efficient application and services. Recently high performance mobile computing, especially over cellular networks, has emerged as a key area of research due to the need for high computing capabilities even when users are on the move. Significant work is being done to optimize performance and create applications for accessing HPC services on smartphones [1][2]. Researchers at MIT have developed a software application for super-computing over a cell phone capable of performing complex calculations for different classes of problems such as fluid flow around spherical objects, effect of force on cracked pillars,etc. [3]. Since it is likely that mobile devices such as smartphones and tablets will replace desktop computers for accessing HPC infrastructure, there is a need for high speed mobile devices with low power consumption, connected through robust, high speed fault tolerant networks. Many companies such as Intel and NVIDIA have been working on the optimization of mobile platforms for supporting HPC applications [4].

Advanced mobile devices, with high computing capabilities and relatively lower costs, have changed the manner in which mobile communication networks have evolved over time. In order to support the HPC services and applications in a mobile environment, various factors such as optimized application design, powerful devices with high computing capability, low power consumption and efficient network performance, etc., have to be taken into consideration. In the current work, we focus on the communication aspects that would be required to support such high performance computations involving mobile devices.

HPC applications are characterized by high computational power, data intensive workloads, parallel processing, high data storage and access rates, high data transfer speeds, and high reliability and accuracy. Hence, to support such applications, a high throughput, low latency, reliable network with low packet loss rates is needed. A lot of work has been done to enhance the performance and efficiency of dedicated data-center networks and HPC cluster networks to meet the demands of large scale resource intensive HPC applications [5][6][7][8]. However, the need to provision strict QoS for supporting high performance mobile computing over existing wireless mobile networks is still a challenging issue. The existing high speed data networks such as the third (3G) and the fourth (4G) generation networks are capable of providing high throughput to the users on both the forward and the reverse channels. Hence, they can potentially cater to the performance requirements of HPC services. However, mobile data traffic has grown exponentially

978-1-5090-2088-1/16/$31.00 ©2016 IEEE

over the past few years and it is expected to grow up to 30.6 exabytes per month by 2020 [9]. Hence, the efficient support of HPC applications over mobile networks is a critical issue due to limited bandwidth, interference of radio signals, channel fading, etc.

In this paper, we provide a mathematical framework for provisioning high QoS for supporting HPC services and applications over existing wireless mobile networks. In order to provide high QoS, while managing the overall network capacity at the same time, resource allocation and charging must be managed carefully. Hence, we also briefly discuss how proper charging and resource allocation can be used to balance resource consumption and the QoS provided.

## II. QoS Based Scheduling Framework

In order to provide high QoS for HPC mobile applications and manage the limited capacity at the same time, resource allocation must be done efficiently. Hence, there is a need for a QoS based scheduler that takes the throughput and delay constraints into consideration. In addition, the Packet Loss Rate (PLR) guarantees can be provided by using the Hybrid-Automatic Repeat Request (H-ARQ) feature, which retransmits the physical layer frames until an ACK is received or the maximum number of transmissions is reached.

### A. Mathematical Formulation of QoS Based Scheduling

In this section, we provide a mathematical formulation for the scheduling problem and show how various QoS constraints can be imposed to provide an acceptable performance for supporting HPC over wireless mobile networks. In [10], Kelly provides a mathematical formulation of the shared resource allocation problem with the utility function defined as a function of throughput. A similar formulation is also possible in terms of other QoS parameters such as delay, PLR, jitter, etc. For instance, a similar formulation has been defined in terms of delay [11], for the scheduling of VoIP traffic over a time-shared wireless data network. However, in the case of HPC mobile applications, the QoS constraints are stringent and there is a need for high throughput, low latency and reliability for providing an acceptable performance. Therefore, all QoS constraints, including throughput, delay and packet loss rate, have to be taken into consideration. In this case, the QoS based scheduling problem can be formulated as follows:

$$maximize \quad F(\vec{r},\vec{d}) \equiv \sum_{i=1}^{k}(UT_i(r_i)+UD_i(d_i)) \qquad (1)$$

$$subject\ to \quad \sum_{i=1}^{k} r_i < C \qquad (2)$$

$$and \quad r_i \geq rmin_i \qquad (3)$$

$$and \quad d_i \leq dmax_i \qquad (4)$$

$$over \quad r_i \geq 0, \quad d_i \geq 0, \ 1 \leq i \leq k. \qquad (5)$$

where,

$k$ = the number of active users competing for the channel,

$UT_i(r_i)$ = Throughput based utility function of user $i$ experiencing an average throughput $r_i$,

$UD_i(d_i)$ = Delay based utility function of user $i$ experiencing an average delay $d_i$,

$C$ = the total channel capacity,

$rmin_i$ = the minimum throughput for user $i$,

$dmax_i$ = the maximum delay allowance for user $i$

For each user, if the utility function is assumed to be strictly concave and differentiable, then it will also hold true for the objective function $F$. Considering the feasible region to be compact, the unique solution that exists can be found using Lagrangian methods. However, due to the dynamic nature of the network (due to variations in the channel conditions and number of active users over time), the optimal solution changes continuously. Hence, we aim at approaching the optimal solution by making appropriate decisions at each scheduling instance. In this case, the gradient ascent method can be used by serving the user for which the objective function gradient is maximum. If the above formulation has no feasible solution, which may occur due to the scarcity of network resources, then congestion is expected to increase continuously. In that case, congestion can be detected by monitoring the gradient of the objective function, i.e. the average priority over all users, which will increase with the increase in network load [12]. Hence if congestion is detected, the number of users can be reduced or the QoS constraints can be lowered until a feasible solution can be achieved.

### B. Enforcing QoS Guarantees for each Attribute

The QoS demands of HPC mobile applications will impose throughput, delay and PLR constraints on the underlying wireless networks. Given the framework for QoS based scheduling, we will now address the issue of providing QoS guarantees for each QoS parameter. In the case of throughput and delay, the concept of a barrier function [13] can be used to enforce various constraints. In addition to the delay and the throughput constraints, the PLR can be maintained by using the H-ARQ retransmission mechanism which was introduced as a one of the enhancements for packet data transmissions in 3G networks and beyond.

#### *Throughput*

Let $rmin_i$ denote the minimum throughput that must be provided to a given user $i$. If we ignore QoS and assume that the utility is a linear function of the average throughput of a paid user, then $U(r) = \alpha r$ for some constant $\alpha$. In this case, if $\mu_i(n)$ is the bit rate of user $i$ if served during the $n$ th

period, then the resulting scheduler will choose the user for which:

$$j^* = \arg\max_j \{\mu_j(n)\} \quad (6)$$

i.e the user for which the achievable bit rate is the highest. In order to guarantee the minimum throughput, the notion of barrier functions, as proposed in [13] can be used. In this case, the utility function can be defined as:

$$UT(r) = r + (1 - e^{-\beta(r - r_{min})}) \quad (7)$$

Hence, if the throughput $r$ drops below $rmin_i$ for user $i$, it will result in a rapid decrease in the utility function. The variable $\beta$ determines the rate of increase of the penalty for violating the constraint. Hence, instead of imposing hard constraints and forcing the user to be served immediately to fall back into the feasible region, this approach is more efficient as it aims at reducing the movement outside the feasible region while making the scheduling decision.

*Delay*

Let $dmax_i$ denote the maximum delay permitted for user $i$. Similar to the case of throughput, barrier functions can be used to impose soft delay constraints on the system allowing it to more easily handle users who are approaching their maximum delay limit *dmax* [11]. In order to ensure a high penalty for users experiencing a high delay, the barrier function can be defined as follows:

$$UD(d) = \frac{(d_{max} - d)^{(1-\alpha)}}{1-\alpha}, \alpha > 0, \alpha \neq 1 \quad (8)$$

where $\alpha$ is a variable whose value can be optimized to achieve the desired packet loss rate since packets are dropped when their delay exceeds the maximum value. The utility function can be defined as $UD(d) = log(d_{max} - d)$ when $\alpha = 1$. In this case, the user selected at each scheduling decision is given by:

$$j^* = \arg\max_j \{\frac{r_j(n)}{\mu_j(n)}(d_{\max} - d_j(n))^{-\alpha}\} \quad (9)$$

In the case of delay sensitive applications such as VoIP, the above utility function can be used to achieve different delay budgets based on the QoS class and the application requirements. If we assume that the long term throughput is maintained at a certain level using the throughput based utility function, as described in equation (7), then the throughput component in the equation (9) can be ignored. In that case, the delay can be estimated in terms of the queue length. In this case, the scheduling decision will be based on the queue length and the user selected at each scheduling decision can be given by:

$$j^* = \arg\max_j \{\frac{r_j(n)}{(q_{\max} - q_j(n))^{\alpha}}\} \quad (10)$$

Therefore, when the system is lightly loaded, the user with the highest required rate will be served. However, as the queue size for a given user increases, its priority level also increases so that it is served before the delay threshold is reached. In this case, the queue dependent utility function can be described as follows:

$$UD(q) = \frac{(q_{max} - d)^{(1-\alpha)}}{1-\alpha}, \alpha > 0, \alpha \neq 1 \quad (11)$$

The utility function can be defined as $UD(q) = log(q_{max} - q)$ when $\alpha = 1$.

*Packet Loss Rate*

For any given flow, the PLR will be dependent on the loss due to failed transmissions. As discussed above, the rate of dropped packets due to excessive delay or buffer overflow is handled by the delay based utility function. The physical layer transmission frame loss rate can be decreased by increasing the number of allowed H-ARQ re-transmissions. In this approach, the physical layer frames are re-transmitted until an ACK is received or the maximum number of transmissions is reached. In addition, a NACK can be used to report an incorrectly received frame, causing the sender to re-transmit the frame. Since H-ARQ is implemented at the physical layer, the overall delay incurred is relatively low (on the order of milliseconds). In order to further enhance the performance, soft combining can be used whereby the incorrectly received data blocks are not discarded, and combined with the retransmitted block when it is received using either Chase Combining or Incremental redundancy technique. Hence, H-ARQ can be used to increase the reliability and efficiency by allowing fast retransmission of incorrectly received frames. A simple feedback approach can be used to achieve the desired transmission loss rate. As this rate increases, the maximum number of H-ARQ attempts is increased and vice-versa.

## III. PRICING FOR QOS BASED DATA SERVICES

In Section II we proposed a QoS framework and discussed how the constraints on different performance attributes can be satisfied for supporting high performance mobile computing over existing mobile networks. We now discuss how resources can be managed efficiently and fairly through proper pricing, load monitoring and admission controls.

Currently, most data plans provided by different operators across the world are flat-rate plans differing from each other in terms of the monthly data allowance or the data cap. A higher per bit charge is applied after the data limit is exhausted [14]. However, once QoS is implemented, user subscriptions must be priced proportionally to their achieved quality.

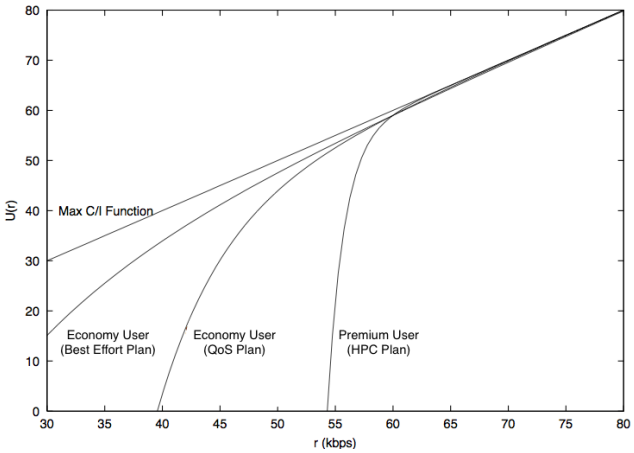


Figure 1. Throughput-Based Utility Function

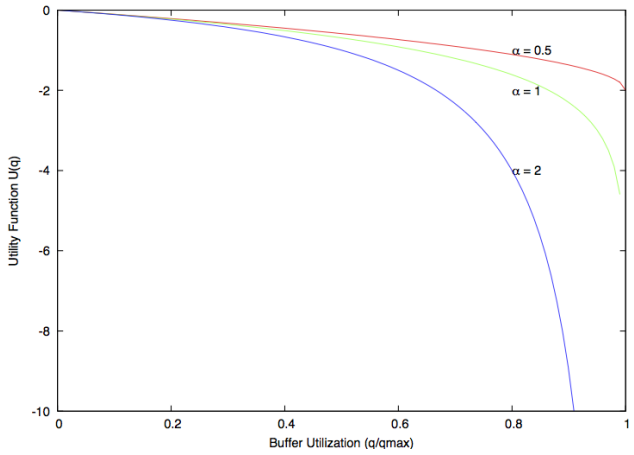


Figure 2. Delay-Based Utility Function

Network operators can use tiered data pricing, along with a suitable congestion detection and admission control mechanisms to implement the above framework. For instance, we propose two different kinds of data plans: A Premium plan devised for HPC users, and an Economy plan for normal users. The economy plan can be futher categorized into a QoS plan and a best-effort plan, differing from each other in terms of the QoS level provided. For example, the users who wish to access interactive multimedia and video-streaming applications can opt for a QoS plan, while those who just want to access emails or other web based content can go for a best effort plan. The per bit price charged will be higher for the QoS plan as compared to the best effort plan, and the total price paid will depend on the QoS level as well as the monthly data cap.

In case of the Premium plan, high QoS (high throughput, low latency and low PLR) can be guaranteed, along with a high data allowance (say 50GB per month). The cost per bit will be much higher for the users who subscribe to the Premium plan when compared to the Economy plans. This is so because significantly more resources have to be reserved for providing more stringent QoS. When this data limit is exhausted, the user can still achieve high QoS but will be charged an even higher cost per bit. This can help to ensure fairness.

Based on the throughput based utility function and the delay based utility function, different values of the constants $\beta$ and $\alpha$ can be chosen to provide different delay budgets for different user classes as defined above. Figure 1 illustrates the use of a throughput-based utility function for different values of $\beta$, chosen to provide different levels of QoS for different user classes. If sufficient bandwith is available, the requirements of all the users can be satisfied. However, when resources become scarce, then a larger penalty is incurred for the premium HPC users, than for the economy users. Similarly, the delay based utility function has been plotted for different values of $\alpha$, as shown in Figure 2. In this case, as the value of $\alpha$ increases, the penalty incurred on reaching the maximum delay also increases. In this case, different values of $\alpha$ can be chosen to satisfy the different QoS requirements for different classes. For instance, a value of $\alpha$ =2 can be chosen for the premium users requiring a lower delay and the values of $\alpha$ =1 and $\alpha$ =0.5 can be used for the economy class users with lower QoS requirements.

At any particular instance of time, only one of the constraints typically forms the resource bottleneck and only this constraint needs to be taken into account. However, as it is not known in advance which constraint will form the bottleneck; all constraints (throughput, delay and PLR) must be included. However, in some cases, such as HPC applications, all the constraints such as throughput, delay and PLR may apply, and in this case, the premium parameters for both throughput and delay will be applicable.

In addition to differential pricing as described above, management of QoS in the case of congestion must also be addressed. In order to detect congestion different load detection algorithms can be chosen by the operators. For instance, as proposed in [12], the average value of the priority over all users, (i.e., the average utility function gradient) can be calculated to estimate the current network load. If this exceeds some threshold then various actions can be taken (e.g., block new connection requests or reduce QoS constraints etc.). The utility function is defined using the notion of barrier functions, as explained in equations (7) and (8), in order to satisfy the throughput and delay constraints respectively. If we consider equation (7), then different values of $\beta$ can be chosen for each subscription class. When sufficient resources are available, the QoS requirements of all users can be satisfied. However in the case of congestion, a high penalty is incurred for regular users when compared to the premium users. This ensures that the premium users continue to enjoy higher priority and hence higher throughput even when the system is congested. Similarly, suitable parameter values can be chosen for the other constraints such as delay, PLR, etc.

Hence, when congestion occurs, users from different service classes are throttled relative to their guaranteed QoS values and subscription class.

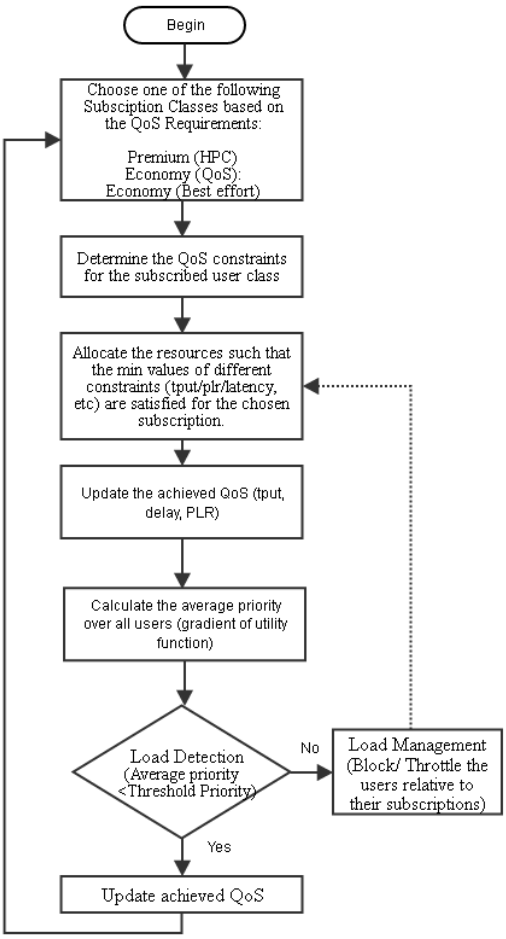


Figure 3. QoS-based Charging and Resource Allocation

Similarly, when the network situation improves, the QoS for each user is restored based on their subscription classes. For example, the Premium user's throughput is increased more relative to that of an Economy plan user. The overall process is shown in Figure 3. In addition, to avoid high costs for the Premium users when not using high performance mobile computing, they can be allowed to switch to the lower subscription classes at any time. This dynamic adjustment of subscription class provides cost benefits to users.

## IV. Future Work and Conclusions

Wireless networks have evolved to support resource intensive applications such as interactive multimedia and video streaming, etc., as well as traditional services such as email, web and voice, over the same network infrastructure. In order to support HPC applications, which require stringent QoS requirements, over the existing wireless mobile networks there is a need for a QoS based framework and a flexible pricing plan based on it, to maximize resource utilization while providing high user satisfaction. In this paper we propose a framework for provisioning high QoS for supporting HPC services and applications over existing wireless mobile networks. In addition, we discuss how proper charging and resource allocation can be used to balance between resource consumption and the QoS provided. The proposed framework provides a simple yet flexible solution for supporting high QoS requirements of HPC applications over existing wireless networks. In future work we plan to simulate realistic scenarios to illustrate the benefits of the proposed approach.

## Acknowledgment

This work has been supported by "Collaboration Platform Service Technology Development and Application: K-16-L01-C02-S03" at Korea Institute of Science and Technology Information (KISTI), Daejeon, Korea and by the Institute for Information and communications Technology Promotion (IITP) grant funded by the Korean Government (MSIP) : No. B0717-16-0033, Study on the multi-dimensional Future Network System Architecture for diversity of Services, terminals and Networks.

## References

[1] "Mobile Broadband Capacity Constraints and the need for Optimization", Rysavy Research, February 2010.

[2] C. V. Deepu, N. Kurkure, P. Dinde, A. Das, A. Gupta and G. Misra, "e-Onama: Mobile high performance computing for engineering research," 2013 Third International Conference on Innovative Computing Technology (INTECH), London, 2013, pp. 532-536.

[3] "Supercomputing on a cell phone", http://news.mit.edu/2010/supercomputer-smart-phones-0901, September 2010.

[4] "NVIDIA:Mobile phones, tablets and HPD (cloud)-Stream Computing," https://streamcomputing.eu/blog/2012-05-12/nvidia-mobile-phones-tablets-and-hpc-cloud/, May 2012.

[5] "Interconnect Analysis: 10GigE and InfiniBand in High Performance Computing", White Paper, http://www.hpcadvisorycouncil.com.

[6] J. Wang, S. Basu, C. McArdle and L. P. Barry, "Large-scale hybrid electronic/optical switching networks for datacenters and HPC systems," 2015 IEEE 4th International Conference on Cloud Networking (CloudNet), Niagara Falls, ON, 2015, pp. 87-93.

[7] S. Narravula, H. Subramoni, P. Lai, R. Noronha and D. K. Panda, "Performance of HPC Middleware over InfiniBand WAN," 2008 37th International Conference on Parallel Processing, Portland, OR, 2008, pp. 304-311.

[8] A. K. Kodi and A. Louri, "Energy-Efficient and Bandwidth-Reconfigurable Photonic Networks for High-Performance Computing (HPC) Systems," in IEEE Journal of Selected Topics in Quantum Electronics, vol. 17, no. 2, pp. 384-395, March-April 2011.

[9] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019," Cisco Whitepaper, www.cisco.com.

[10] F. Kelly, "Charging and rate control for elastic traffic," European Trans. On Telecommunications, vol. 8, pp. 33-37, 1997.

[11] P. Hosein, "Scheduling of VoIP traffic over a time-shared wireless packet data channel," 2005 IEEE International Conference on Personal Wireless Communications, 2005, pp. 38-41.

[12] Ramneek, P. Hosein and W. Seok, "Load metric for QoS-enabled cellular networks and its possible use in pricing strategies," 2014 IEEE Symposium on Wireless Technology and Applications (ISWTA), Kota Kinabalu, 2014, pp. 30-35.

[13] P. A. Hosein, "QoS control for WCDMA high speed packet data," 4th International Workshop on Mobile and Wireless Communications Network, 2002, pp. 169-173.

[14] Soumya Sen, Carlee Joe-Wong, Sangtae Ha, and Mung Chiang, "A survey of smart data pricing: Past proposals, current plans, and future trends," ACM Comput. Surv. 46, 2, Article 15, November 2013.