♦ General Method for Network Providers to Choose Applications With Guaranteed QoS

Mikhail A. Panshenskov and Alexander T. Vakhitov

In the presence of the competition for network resources, network providers may experience some difficulty in providing throughput as specified by service-level agreements. Moreover, fulfilling a quality of service (QoS) agreement for a specific application may affect QoS agreements for other applications served by the same network provider. We describe a model to estimate QoS for single and multiple applications and offer a method for a network provider to choose a set of applications with a guaranteed QoS in order to maximize profit with respect to limited network resources. We describe and compare four suboptimal algorithms which can be used when optimal methods cannot be employed. © 2011 Alcatel-Lucent.

Introduction

A network provider often acts as a middleman between end users and application providers. For example, end users often employ a basic service (such as Web access) from a network provider in order to gain access to services from an application provider. One drawback of this scheme is that the user must pay both the application provider and the network provider; a clearer scheme for the user is to pay once per subscription per application. Based on this idea, we envision a future where network providers begin to collaborate more closely with application providers in order to build a single application marketplace (like the Apple App Store*) for end users.

High Capacity Networks

High capacity wireless networks, such as fourth generation (4G) networks, enable a broad range of applications, like high-definition television (HDTV) or HD video calls, which were barely possible in earlier networks. In modern network standards like IEEE 802.16m, channel bandwidth is shared between applications. The activity of one application may affect the quality of service (QoS) of another one. If we consider the provider network as a tree, as illustrated in **Figure 1**, it becomes clear that network contention effects can arise at every internal node as a result of heavy traffic from its children nodes. Such a situation can present a serious problem for service quality, for example, when a voice call, streaming audio, or streaming video application is affected. A possible solution to avoid this situation is to control the bandwidth available for each application, so one application cannot influence the QoS of another.

Related Work

Network providers have two well-developed approaches to QoS guarantees: integrated services (IntServ) [8] and differentiated services (DiffServ) [4, 7]. In the first approach, users can specify different contracts for their own traffic: contracts for microflows

Bell Labs Technical Journal 16(2), 57–62 (2011) © 2011 Alcatel-Lucent. • DOI: 10.1002/bltj.20502

(independent contracts for small portions of traffic) or contracts for virtual private network (VPN)–like tunnels with limited bandwidth. Both types of contracts can be seen as unrealistic because of the high variability in the traffic for individual connections and for tunnels [5]. The DiffServ approach proposed a discrete set of bandwidth values, called QoS classes, which would provide users with the ability to dynamically adjust their QoS level [7]. However, from the network provider's perspective, the scheme is somewhat unpredictable, and users, meanwhile, have demonstrated some reluctance to switch QoS and pay a different price for the same service [4]. In this paper, we view the problem of QoS in terms of satisfying a given bandwidth requirement for a given window within an interval of time. For example, QoS can be defined on a per-hour basis for each of the 24 hours in a day.

The pricing of services is another important question. Research has mostly focused on congestion-based pricing, where the price for transmitting or receiving a chunk of data rises when the network becomes congested. However, as noted in [5], if necessary, a network can be adjusted for higher capacity, so congestion should be the exception, not the norm. The price for a service should justify the network costs plus a reasonable profit on a long term basis [1]. Therefore, pricing based on individual bids for packets, as was proposed in [3], seems highly unreliable and an uncomfortable prospect for industry practitioners. In this paper, we adopt a model where revenue for applications with QoS guarantees is fixed and known.

Panel 1. Abbreviations, Acronyms, and Terms

4G—Fourth generation
ABD—Application bandwidth demand
DiffServ—Differentiated services
HDTV—High-definition television
IntServ—Integrated services
QoS—Quality of service
VPN—Virtual private network

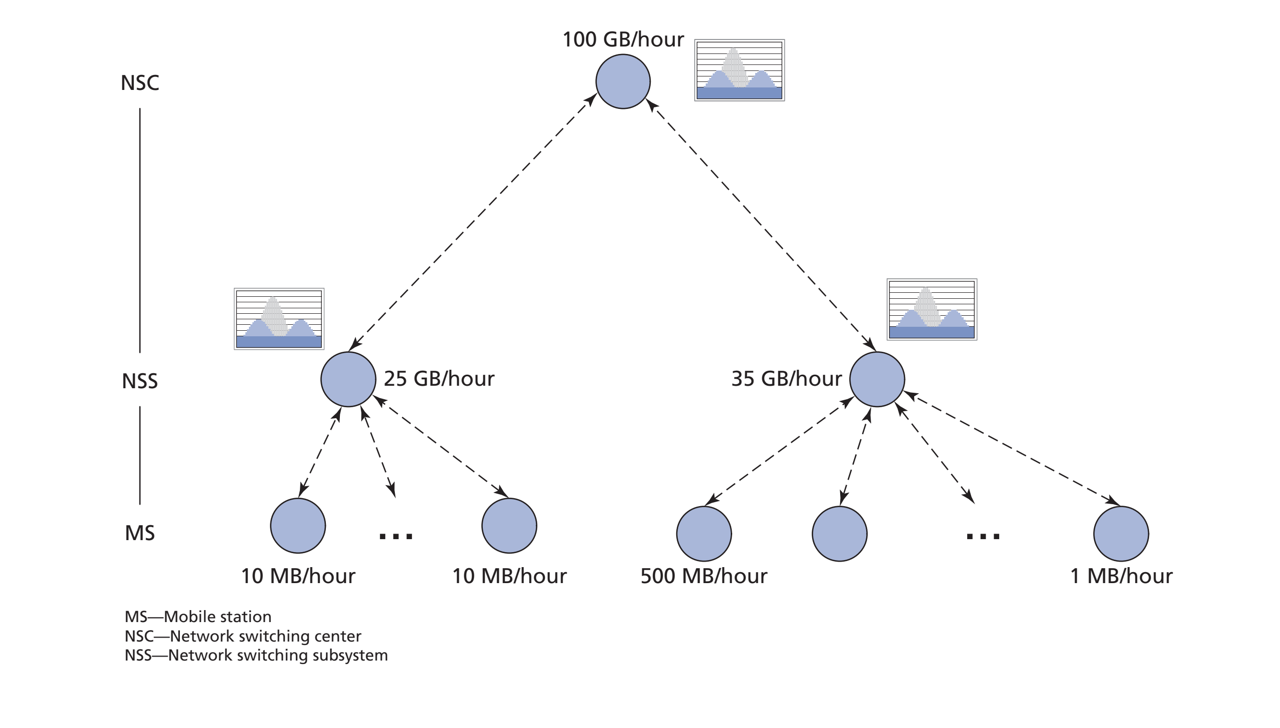

Figure 1.
Provider network.

In a service level agreement, a bid for a specific application involves a pair of application bandwidth demands (ABDs) and a fixed revenue value. This bid can be proposed by an application provider or created jointly with a network provider and should rely on user demand projections and business considerations. In [5], the author conducted experiments which reveal that while individual user traffic is indeterminate, there are certain statistical patterns of group network behavior which repeat every 24 hours. We believe that repeated patterns exist for every application, because we assume the existence of significant numbers of application users.

Problem Setting

Traffic is bounded by the physical capabilities of the network. The problem then is managing traffic with maximal efficiency. We consider profit as a measure of efficiency, and because the cost of network operations does not depend on actual network use, we tackle the problem of profit maximization for the network provider.

Let there be one network and several applications $A = \{a_1 \ldots a_N\}$ which can be installed. User demand for certain applications can be assumed to be either fixed or varying in time. The first case is easier, but we believe the second case is more natural for high-level applications such as social networking, torrent, or video applications. The application bandwidth demand *ABD* depends on the user demand and the characteristics of the application. Let us define *ABD* as $D \downarrow a(t)$ for application a. We assume that ABD is a continuous function defined on some closed interval $[0,T]$ representing an operating time frame such as a day, month, or year. We consider that customer needs for a specific application have been met, and thus quality of service guarantees satisfied, if the network provider can satisfy ABD.

For the network provider, the profitability of each application is defined by the D function, which maps directly to the portion of the application price the network provider receives if it provides the QoS level it has been contracted to provide. The network provider's goal therefore is to choose a subset of all applications $A_{QoS \subseteq A}$ with a guaranteed QoS that will maximize accumulated profit, P:

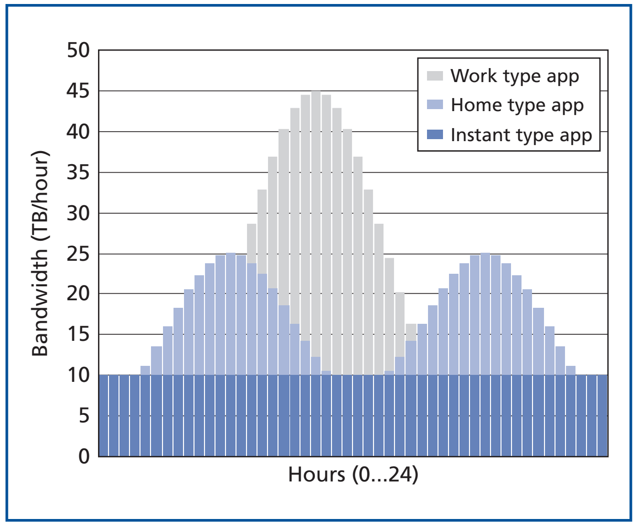

Figure 2.
Bandwidth distribution per application.

$$P = \sum_{a_i \in A_{Qos}} P_{ai}$$

The bandwidth restriction at any moment t for a subset of chosen applications A_{QoS} is the network capacity, C:

$$\forall t = \sum_{a_i \in A_{Qos}} D_{ai}(t) \leq C.$$

Figure 2 provides an example of distribution D for different applications. The following section offers a solution to this problem.

Solution

The problem defined above can be reduced to the multidimensional knapsack problem [2] and thus is NP-hard. In the interest of space, this paper does not include a proof for this fact. Because the time complexity for an optimal algorithm is exponential in a number of applications, we offer several suboptimal algorithms, detailed below, and compare the quality of the solutions:

- Greedy profit
- Greedy mean profit
- Greedy max profit
- Probabilistic

Greedy algorithms choose applications of the highest rank by some real-valued ranking function. The *greedy profit* algorithm ranks applications with higher profit higher overall regardless of the amount of bandwidth used:

$$R(ai) = P(ai).$$

In the *greedy mean profit* algorithm, the ranking function is profit divided by the mean amount of bandwidth used:

$$R_{mean}(ai) = \frac{P(ai)P \cdot T}{\int_0^T Dai(t)}$$

Meanwhile, the *greedy max profit* algorithm uses the maximum instead of an integral in its ranking function:

$$R_{\max}(ai) = \frac{P(ai)}{\max\limits_{t \in [0,T]} Dai(t)}.$$

Finally, the *probabilistic* algorithm chooses applications randomly with the probabilities proportional to function $R_{mean}(a_i)$. The random process is repeated a number of times and the optimal solution of all iterations is chosen.

Experiments

We conducted an experiment for each of the four algorithms described in the previous section. In order to simulate real conditions, we considered three different types of applications, as shown in Figure 2, namely: a *work type application* with peak usage in the middle of the day, a *home type application* with peak usage in the morning or evening, and an *instant application* with constant bandwidth demand. The parameters were generated uniformly in the appropriate range of values. The price was generated in the range 1 . . . 50 price units, the network capacity was fixed to 100 bandwidth units, and the number of applications varied in the range of 3 . . . 36.

The results of every execution by every algorithm were divided by the optimal profit value. Thus in **Figure 3** we demonstrate the ratio of the algorithm result to the optimal result. The probabilistic algorithm produced poor results; still we believe that it can be sufficiently improved by using more complex probabilistic schemas. The greedy profit algorithm demonstrated adequate results only for a relatively small number of applications; this is because it does not take ABD into account. Both the greedy mean profit and greedy max profit algorithms demonstrated good results. Still, we predict that since the greedy mean profit is more robust, it is likely to produce better results in the presence of a higher variety of applications.

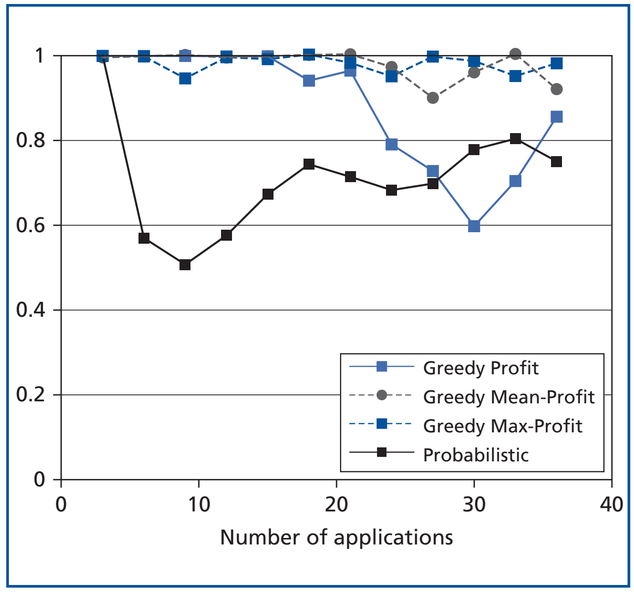

Figure 3.
Algorithms relative quality.

Another interesting criterion for an algorithm is its overall robustness. We consider an algorithm A more robust than algorithm B, if after a small change in the application bandwidth demand D of one application, the set $A_{QoS}(A)$ is changed less than the set $A_{QoS}(B)$. The change of application set A_{QoS} can be measured with respect to the size of the symmetric difference in the sets. Assuming that the bandwidth demand D of an application increases for ϵ at every moment in time, we considered the change A_{QoS} for every algorithm above. Greedy profit is the most robust algorithm since its result, A_{QoS}, does not depend on D. The probabilistic algorithm is the least robust since its result may change even if its input remains unchanged. As for the greedy max profit and greedy mean profit algorithms, relative robustness cannot be strictly identified and depends on the bandwidth demand function D. However, in some special cases, such as when a family of functions D can be presented as $a = f_0 + \beta$, the greedy max profit algorithm is more robust than the greedy mean profit algorithm.

Conclusions

This paper has described situations in which a next-generation network provider could potentially fail to guarantee quality of service at a level to which it had contracted. In order to tackle the problem as described, we offer to limit the set of applications with guaranteed QoS and describe a method for choosing such a set in order to maximize network provider profits while satisfying network constraints. To solve the problem we considered four different suboptimal algorithms, which were compared for the simulated set of applications. The greedy mean profit and greedy max profit algorithms demonstrated the best results and most robust behavior.

In the future, application bandwidth demand estimation methods should also be presented and analyzed. The probabilistic nature of traffic can be explored and integrated into the choice of application model, as is done in [6] with overbooking methods. The dependency of user demand on application price may also be introduced, to explain pricing policies used by network providers to control demand and congestion. Another possible generalization of the model can be in prioritization of applications, as the current model classifies them only as accepted or declined. There are no significant difficulties in considering more than two application classes.

*Trademarks

App Store is a trademark of Apple, Inc.

References

[1] W. J. Baumol and J. G. Sidak, Toward Competition in Local Telephony, MIT Press, Cambridge, MA, 1994.

[2] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, San Francisco, CA, 1979.

[3] J. K. MacKie-Mason and H. R. Varian, "Pricing the Internet," Public Access to the Internet (B. Kahin and J. Keller, eds.), Prentice Hall, Englewood Cliffs, NJ, 1995, pp. 269–314.

[4] K. Nichols and S. Blake (eds.), "Differentiated Services Operational Model and Definitions," IETF Internet Draft, Feb. 1998, <http://tools.ietf.org/html/draft-nichols-dsopdef-00>.

[5] J. W. Roberts, "Internet Traffic, QoS, and Pricing," Proc. IEEE, 92:9 (2004), 1389–1399.

[6] B. Urgaonkar, P. Shenoy, and T. Roscoe, "Resource Overbooking and Application Profiling in Shared Hosting Platforms," Proc. 5th Symp. on Operating Syst. Design and Implementation (OSDI '02) (Boston, MA, 2002), pp. 239–254.

[7] X. Wang and H. Schulzrinne, "Pricing Network Resources for Adaptive Applications in a Differentiated Services Network," Proc. 20th IEEE Internat. Conf. on Comput. Commun. (INFOCOM '01) (Anchorage, AK, 2001), vol. 2, pp. 943–952.

[8] P. P. White, "RSVP and Integrated Services in the Internet: A Tutorial," IEEE Commun. Mag., 35:5 (1997), 100–106.

(Manuscript approved May 2011)

MIKHAIL A. PANSHENSKOV was a software design engineer in Alcatel-Lucent's Mobile Application Division in Namur, Belgium, from 2007 to 2010. He is now a full-time researcher at Saint Petersburg State University, Russia, working under the chair of the university's Parallel Algorithms department. His industrial work experience includes extensive research with telecommunication services and networks from high-level design down to the low-level implementation, as well as scheduling and parameter estimation in distributed networks. He holds the equivalent of an M.S. degree in software engineering from Saint Petersburg State University and has published a number of papers in journals, international conferences, and workshop proceedings.

ALEXANDER T. VAKHITOV is a research group leader at Saint Petersburg State University, Russia. He has worked on innovative algorithms in the field of stochastic optimization in order to enhance multi-core distributed systems, image processing systems, and practical expert systems. He has also worked on a number of image processing algorithms. He received the equivalent of an M.S. degree in software engineering and a Ph.D. degree in mathematics, both from Saint Petersburg State University. Dr. Vakhitov has published more than 20 papers in journals and international conference and workshop proceedings on topics in control theory, distributed systems, image processing, and expert systems. ◆