Comparisons of QoS from User's Perspective; Which Provides Better Utility to Users, Best–effort or Reservation–based Services?

Akira Watanabe Tetsuya Takine† Masayuki Murata Hideo Miyahara

Graduate School of Engineering Science, Osaka University
Toyonaka, Osaka, 560–8531, Japan
Phone +81–6850–6588, Fax +81–6850–6589
E-mail {watanabe, murata, miyahara}@ics.es.osaka–u.ac.jp

†Graduate School of Infomatic, Kyoto University
Kyoto, 606-8501, Japan
Phone +81–075–753–4758, Fax +81–075–753–4756
E-mail takine@kuamp.kyoto–u.ac.jp

Abstract: The conventional Internet has only provided the best–effort service, which does not offer any QoS (Quality–of–Service) guarantees. However, recent developments of multimedia applications require QoS guarantees for real–time transfers, which eventually introduced reservation–based protocols. However, it is pointed out that reservation–based protocols such as RSVP have several drawbacks such as a scalability problem. In this paper, we introduce user's utility to quantify QoS, and it is used to compare the best–effort and reservation–based services to discuss which service gives a better solution for real–time applications and data applications. By extending our previous results, we discuss the worst utility that the user experiences during the connection in this paper. The tandem network model is also treated to investigate the effect of multiple link systems on both services.

1. Introduction

Recently, multimedia applications on the Internet are actively developed. However, the conventional Internet has only provided the best–effort service, which does not offer any QoS (Quality–of–Service) guarantees. It means that multimedia applications sometimes offer very low–quality presentation to users. Accordingly, a new architecture of ISPN (Integrated Service Packet Network) [1] was proposed to offer the guaranteed QoS for real–time applications on the Internet. Every flow is provided with reserved bandwidth during the connection in ISPN. The mechanism is implemented by the signaling protocol called RSVP [2] and the packet scheduling at the router, such as WFQ [3].

However, several drawbacks of reservation–based networks are pointed out more recently. Those include a protocol overhead and limitation on scalability. Another problem is that the user has a possibility to encounter the connection blocking as the network becomes congested, which is unavoidable in the reservation–based protocol. Thus, it is now recognized that, from another point of view, more important is to accept connections on demand even if the provided QoS is not high. In that sense, a rate–adaptive control mechanism seems to be promising in order to offer not high, but acceptable QoS to allow a flexible use of the network bandwidth. The rate–adaptive control utilizes the capability of controlling the packet generation rate at the source, and can be applied to both networks offering reservation–based and best–effort services.

In the reservation–based network, a signaling protocol called bandwidth re–negotiation is necessary to allocate the bandwidth among the connections with rate–adaptive control. If the network gets short of the bandwidth, it informs established connections that they have to decrease their bandwidths, by which more connections can be accepted at the sacrifice of the decreased QoS level [4]. In [4], the authors consider the MPEG-2 CBR encoding method, by which the generation rate can be changed so as to conform to the assigned bandwidth while keeping user's perceived QoS to be acceptable.

The similar mechanism has already been implemented in real–time applications on the Internet. In such applications, the source monitors the network congestion level based on the feedback information from the receiver, and controls the packet generation rate [5]. A fundamental difference from the reservation–based network is that in the reservation–based service, the number of connections can be limited through connection admission control to guarantee a minimum QoS level while it cannot in the best–effort service. That is, the rate–adaptive mechanism in the best–effort service may improve QoS, but it never guarantees QoS.

Therefore, it becomes important to identify to what extent each service can provide QoS from user's point of view, when real–time applications are introduced. In [6], we introduced user's utility to quantify a level of QoS that each service offers, and to compare the QoS capability of two services. However, we only considered the average value of utility in [6], while more important is that the user experiences the fluctuation of QoS during the connection if the user utilizes the best–effort service. As will described later, the fluctuation of QoS during the connection is also observed in the reservation–based service, but in that case, the minimum QoS can be guaranteed. We therefore compare two services based on the *worst utility* experienced during the connection in this paper by extending our previous approach.

We also present new results using the tandem–network model. It is well known that in the circuit–switched network, the long–hop connection encounters the high blocking probability. Since the reservation–based service is essentially the circuit–switched network, QoS offered by the reservation–based service may be much degraded in the large–scaled network. We will use the tandem–network model to quantitatively evaluate it.

This paper is structured as follows. In Section 2, we introduce the services and application models. User's worst utilities according to our definitions are then derived in both services in Section 3. The network model is then extended to the tandem network model in Section 4. In Section 5, we compare two services by numerical examples. We conclude our paper in Section 6.

2. Service and Application Models

In this paper, we consider the following two network service models.

(1) Reservation–based service:

In the reservation–based service, the network reserves physical network resources for the connection before actual communications. For this purpose, some signaling protocol such as RSVP should be equipped with the network. By this mechanism, a part of the network resource is dedicated to the connection, and the QoS guarantees can be realized. If the network resource is short, the connection is blocked.

Network resources may be bandwidth and/or router buffer, and QoS may be represented by throughput, packet delay time and/or packet loss rate. In this paper, we assume that the network reserves the bandwidth for the connection, and therefore, the QoS parameter is throughput. It is a simplest and most realistic form of the reservation–based service.

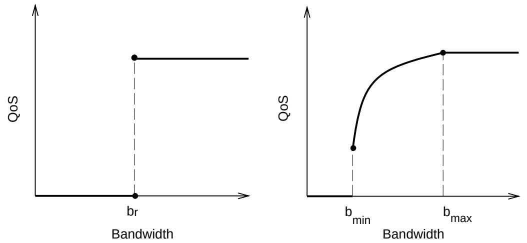
(2) Best–effort service:

In the best–effort service such as the conventional Internet, no QoS is guaranteed for the connection. However, blocking due to lack of the network resources never occurs in this case.

For network applications, we consider interactive real–time applications where resource reservation is necessary if we want QoS guarantees. We can consider the rate–adaptive control in which the packet generation rate is controlled against the network congestion level. That is, the real–time application is divided into the following two applications;

(1) rigid application having no rate–adaptive capabilities,
(2) rate–adaptive application.

Data applications and one–way real–time applications are other important applications, but in this paper, we only consider the interactive real–time application, which is applied to the reservation–based service and the best–effort service. For the case where interactive real–time and data applications co–exist, refer to [6]. To compare the QoS capability of two services, we introduce a notion of *utility*. If we consider SN ratio or MOS as QoS, the utility can be quantitatively represented as a function of the bandwidth [4]. Figure 1(a) shows the QoS function for the rigid applications. If the allocated bandwidth is less than b_r, then the user perceives that it is very uncomfortable, and therefore the utility becomes 0. If the bandwidth of b_r is guaranteed, on the other hand, users are satisfied with comfortable communication. The QoS function for the rate–adaptive application is illustrated in Figure 1(b). If the application is provided with the guaranteed bandwidth of b_{max}, the user can enjoy QoS–rich communication. However, even if the bandwidth is decreased, acceptable QoS is offered to the user as long as the minimum bandwidth of b_{min} is guaranteed. In what follows, QoS functions shown in the figure are represented by $\alpha(\cdot)$ (for rigid real–time applications) and $\beta(\cdot)$ (for adaptive real–time applications).



(a) Rigid real–time application

(b) Rate–adaptive real–time application

Figure 1: QoS functions for applications

In summary, we will compare the utility based on user's perceived QoS by applying

(1) rigid real–time applications, or
(2) rate–adaptive real–time applications

to either of

(i) best-effort service, and
(ii) reservation-based service

in the following sections.

3. Derivation of Worst Utilities

3.1 Network and traffic models

We first consider the worst utility. For this purpose, we assume the single link having the capacity C, which is shared by real–time applications. Connection setup requests from real–time applications arrive at the link following a Poisson distribution with rate λ, and connection holding times are assumed to be exponentially distributed with mean $1/\mu$. For real–time applications, we consider rigid real–time applications in Subsection 3.2 and adaptive real–time applications in Subsection 3.3, respectively.

3.2 Rigid real–time applications

3.2.1 Application to the reservation–based service

In this subsection, we consider the case where the rigid application is applied to the reservation–based service.

The rigid application requires a fixed amount of bandwidth (b_r) to be reserved to establish its connection (Figure 1(a)). The connection setup is refused if the remaining capacity of the link is short. That is, the reservation–based service can accept the maximum number $m = \lfloor C/b_r \rfloor$ of connections on the link. It can be modeled as an M/M/m/m queuing model [6]. Once the connection is accepted with the bandwidth b_r, the quantitative level of QoS, $\alpha(b_r)$, is guaranteed during the connection. Thus, the worst utility is equal to the average utility, which is given by

$$U_{r,r}(C) = (1 - L_{r,r})\,\alpha(b_r),$$

where $L_{r,r}$ is an Erlang blocking probability. We should note here that we simply exclude the case of blocking in the above equation. We may have to take account of the negative effect to the user by reservation blocking. We need further research to incorporate such an effect to represent the user's utility.

3.2.2 Application to the best–effort service

When the rigid real–time applications are applied to the best–effort service, all connections are accepted. When the number of real–time connections exceeds $m(= \lfloor C/b_r \rfloor)$, however, the perceived QoS becomes 0. By assuming that users of real–time applications do not give up connections even if the utility falls under the acceptable level, the behavior of real–time applications is modeled by an M/M/∞ queuing system.

To derive the worst utility, we suppose that the tagged connection arrives at time 0 and finds the number k of active connections in the system. The worst utility of the tagged connection is experienced when the tagged connection finds the maximum number of connections in the system during its connection time, which is denoted as n. If $n < m$, then the user of the tagged connection does not perceive the QoS degradation. Otherwise, the QoS becomes 0 in the rigid real–time application. The number n can be determined by analyzing the transient behavior of the system. The probability that the tagged connection finds at most the number n of connections during its connection time is given by

$$r_n(k) = (I - G/\mu)^{-1}\, e, \tag{1}$$

where e is a column vector with all elements 1. The $(n+1) \times (n+1)$ matrix G is given by

$$G = \begin{bmatrix} -\lambda & \lambda & & 0 \\ \mu & -(\lambda+\mu) & \lambda & \\ \ddots & \ddots & \ddots & \\ & (n-1)\mu & -(\lambda+(n-1)\mu) & \lambda \\ 0 & & n\mu & -(\lambda+n\mu) \end{bmatrix} \tag{2}$$

where (i,j) element of e^{Gt} gives the probability that the number of connections in the system is i at time 0, and that the number of active connections does not exceed n until time t, at which the number of active connections becomes j.

Finally, we obtain the worst utility as

$$WU_{r,b}(C) = \sum_{k=0}^{m-1} r_{m-1}(k)\, q(k)\, \alpha(b_r),$$

where $q(k)$ is a steady–state probability that there exists the number k of connections in the M/M/∞ queuing system, which is given as $q(k) = e^{-a} a^k / k!$, where a is traffic load of real–time applications.

3.3 Adaptive applications

3.3.1 Application to the reservation–based service

The adaptive real–time application is tolerant to the assigned bandwidth variation as having been shown in Figure 1(b). That is, the application allows the bandwidth from b_{min} to b_{max}. At the connection setup time, the connection is established if the allocated bandwidth of at least b_{min} is reserved for that connection. Note that we assume that the link bandwidth is fairly shared among real–time applications. For this, the bandwidth re–negotiation is necessary in some way when the connection newly arrives or terminates [4]. Thus, the assigned bandwidth to each connection may be changed during the connection time between b_{min} and b_{max}, i.e., the perceived QoS may fluctuate between $\alpha(b_{min})$ and $\alpha(b_{max})$.

The reservation–based service can accept the maximum of $m_3 = \lfloor C/b_{min} \rfloor$ connections on the link, which leads to an M/M/m_3/m_3 queuing system. The maximum bandwidth is assigned to the connection when the number of active connections m_3 is less than or equal to $\lfloor C/b_{max} \rfloor$.

To determine the worst utility in this case, we consider the transient behavior of the M/M/m_3/m_3 queuing system in a similar way presented in Subsection 3.2.2. Let $\overline{r}_n(k)$ be the probability that the tagged connection finds *at most* the number n of connections in the system during its connection time, when the connection finds k connections at its arrival time. It is given by changing $(n+1, n+1)$ element of G in Eq. (2) to $-n\mu$ and applying it to Eq. (1). Then the probability that the tagged connection finds the maximum number n of active connections during its connection time is given by

$$\begin{aligned} s_n(k) &= \overline{r}_n(k) - \overline{r}_{n-1}(k), \qquad n > k, \\ s_n(n) &= \overline{r}_n(n). \end{aligned}$$

The worst utility $WU_{a,b}$ in this case is then given as

$$\begin{aligned} WU_{a,r}(C) = & \sum_{k=0}^{m_1-1} \sum_{j=k}^{m_1-1} s_k(j)\, p(j)\, \beta(b_{max}) \\ & + \sum_{k=m_1}^{m_2-1} \sum_{j=m_1-1}^{k} s_k(j)\, p(j)\, \beta(C/k). \end{aligned}$$

where $p(j)$'s are steady state probabilities of the M/M/m_3/m_3 queueing system;

$$p(j) = (a^j/j!)/(\sum_{l=0}^{m_3} a^l/l!).$$

3.3.2 Application to the best–effort service

It may be a too simple assumption, but we assume that the bandwidth is fairly shared by adaptive real–time connections by considering that the connections are equipped with some ideal rate–adaptive mechanism. Note that several rate–adaptive mechanisms for real–time applications have already been proposed in the literature, while in the actual situation, those mechanisms may not be able to adjust its rate fairly.

The worst utility $WU_{a,b}(C)$ can be obtained by changing $s_n(k)$ in Eq. (3) to $t_n(k)$ given as

$$t_n(k) = r_n(k) - r_{n-1}(k), \qquad n > k,$$
$$t_n(n) = r_n(n),$$

where $r_n(k)$ was obtained in Eq. (1).

4. An Extension to Tandem Network Model

4.1 Network and traffic models

In this section we treat the tandem network model with N links as shown in Figure 2. The designated (foreground) connections with N hops uses Path 0, and the single–hop (background) connections arriving at link i uses Path i. Connections arrive at Path i ($0 \leq i \leq N$) according to a Poisson distribution with rate λ_i. The capacity of each link is C, and the holding times of connections on Path i are assumed to be exponentially distributed with mean $1/\mu_i$, respectively. We assume that the connection setup times including propagation delays between nodes are negligible since those must be large enough compared with the connection holding times. Note that from a viewpoint of user's utility, the connection setup time may be another important factor, but it is beyond scope of our current paper.

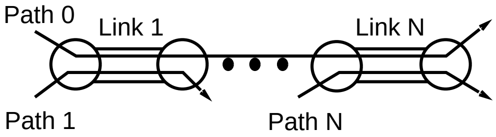


Figure 2: extended network model

Since our concern is how the utility is degraded dependent on the number of hop counts, we will derive the utility of long–hop connections in the below.

4.2 Rigid real–time applications

4.2.1 Application to the reservation–based service

The long–hop connection on Path 0 is only accepted if the bandwidth of b_r is reserved on all links. Recalling that the maximum number of acceptable connections is $m = \lfloor C/b_r \rfloor$ on each link, we obtain the steady state probability $P_{u_0,\cdots,u_N}$ that the number of connections on Path i is u_i as

$$P_{u_0,\cdots,u_N} = P_{0,\cdots,0} \prod_{i=0}^{N} \frac{a_i^{u_i}}{u_i!}, \tag{3}$$

where $P_{0,\cdots,0}$ is given as

$$P_{0,\cdots,0} = 1/\sum_{j=0}^{m}[\frac{a_0^{u_0}}{u_0!} \times \prod_{i=1}^{N}(\sum_{u_i=0}^{m-j}\frac{a_i^{u_i}}{u_i!})]. \tag{4}$$

To derive the utility for the long–hop connection on Path 0, we first determine its blocking probability, $LL_{r,r}$. The connection blocking occurs when the number of connections is equal to m on at least one of links. That is, we have

$$LL_{r,r}(C) = \sum_{\{u_0,\cdots,u_N\} \text{ such that } u_0+u_i=m\ (1\leq i\leq N)} P_{u_0,\cdots,u_N}.$$

The utility for long–hop connections is then given as

$$LU_{r,r}(C) = (1 - LL_{r,r}(C))\alpha(b_r).$$

4.2.2 Application to the best–effort service

When the rigid real–time application is applied to the best–effort service, we assume that all connections are accepted. Thus, each link can be modeled by an independent M/M/∞ queue. We thus have a steady state probability as

$$Q_{u_0,\cdots,u_N} = \prod_{i=0}^{N} q_i(u_i), \tag{5}$$

where $q_i(u_i)$ is given as

$$q_i(u_i) = e^{-a_i}a_i^{u_i}/u_i!.$$

If the number of connections on at least one of links is larger than m, then available bandwidth of long–hop connections on Path 0 is less than b_r and the utility becomes 0 as having been described in Subsection 3.2.2, Thus, the utility for long–hop connections is given as

$$LU_{r,b}(C) =$$
$$\frac{1}{a_0}\sum_{u_0=1}^{m}\sum_{u_1=0}^{m-u_0}\cdots\sum_{u_N=0}^{m-u_0} u_0\,Q_{u_0,\cdots,u_N}\,\alpha(b_r).$$

4.3 Adaptive applications

4.3.1 Application to the reservation–based service

Next, we consider the utility of the adaptive applications applied to the reservation–based service. Recalling that the maximum number of allowable connections on each link is $m_3 = \lfloor C/b_{min} \rfloor$ (see Subsection 3.3.1), the steady state probability $R_{u_0,\cdots,u_N}$ that the number u_i of connections is active on Path i is given by changing m in Eq. (3) to m_3.

The various bandwidth allocations to connections can be considered for given state $\{u_0,\cdots,u_N\}$. In this paper, we assume that the bandwidth re–negotiation protocol allocates the bandwidth to achieve the Max–Min fairness [7]. Its implementation in the actual network may not be easy, but in our current tandem network model, the allocated bandwidth is easily determined as follows. First, choose the link having the maximum number of connections, and let the number denote as u_{max}. Then, the bandwidth allocated to the connections on Path 0 is determined as $C/(u_0 + u_{max})$, i.e., the capacity of that link is equally divided. Connections through other links fairly share the remaining bandwidth $C - u_0\, C/(u_0 + u_{max})$. That is, the bandwidth allocated to other connections on Path i $(1 \leq i \leq N)$ becomes

$$\frac{u_{max}}{u_i} \frac{C}{u_0 + u_{max}}.$$

Since the utility of long–hop connections on Path 0 is $\beta(C/(u_1 + u_{max}))$ for given u_1, the average utility is given as

$$LU_{a,r}(C) =$$
$$\frac{1}{a_0} \sum_{u_0=1}^{m_2} \sum_{u_1=0}^{m_2-u_0} \cdots \sum_{u_N=0}^{m_2-u_0} u_0\, R_{u_0,\cdots,u_N}\, \beta(\frac{C}{u_0 + u_{max}}).$$

4.3.2 Application to the best–effort service

We finally consider the case where the adaptive application is applied to the best–effort service. The adaptive application utilizing the best–effort service should determine its rate by itself according to the feedback information. However, we again introduce the simple assumption that each connection ideally changes its rate so that the Max-Min fairness can be eventually achieved. We notice that it is an ideal assumption, but our concern is not its realization, but to compare the utilities of best–effort and reservation–based services.

By the above assumption, the utility of connections on Path 0 is given by $\beta(C/(u_0 + u_{max}))$ and its utility is determined as

$$LU_{a,b}(C) =$$
$$\frac{1}{a_0} \sum_{u_0=1}^{\infty} \sum_{u_1=0}^{\infty} \cdots \sum_{u_N=0}^{\infty} u_0\, Q_{u_0,\cdots,u_N}\, \beta(\frac{C}{u_0 + u_{max}}),$$

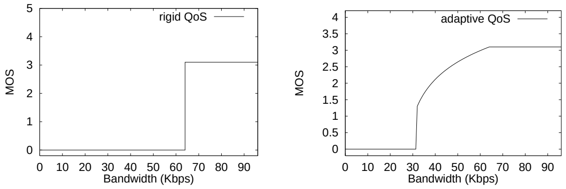
where $Q_{u_0,\cdots,u_N}$ was given by Eq. (5).

5. Numerical Examples

In comparing the reservation–based and best–effort services, we use a *utility difference* which represents how much the utility obtained by the reservation–based service is larger than the one by the best–effort service. That is, the utility difference $XD_y(C)$ ($X = W$ for the worst utility and $X = L$ for long–hop connections that we will use later in this section; $y = r$ for rigid real–time applications and $y = a$ for adaptive real–time applications) is given by

$$XD_x(C) = XU_{y,r}(C) - XU_{y,b}(C).$$

Larger values of the utility difference give more preference to the reservation–based service. QoS functions that we will use in the numerical examples are shown in Figure 3, which model the digitized voice.



(a) Rigid application $\alpha(\cdot)$

(b) Adaptive application $\beta(\cdot)$

Figure 3: QoS functions

We first compare the worst utilities of reservation–based and best–effort services using the single link model (Section 3.). For numerical examples, we fix the traffic load ($a = \lambda/\mu$) to be 20 Erlang while the link capacity C is changed. Figures 4 and 5 present worst utilities of rigid and rate adaptive real–time applications and utility differences. For comparison purpose, the average utilities ($U_{r,r}$, $U_{r,b}$ and DU_r) obtained in [6] are also shown. From Figure 4, we can observe that when the link capacity is not large, the worst utility of the best–effort service becomes almost 0, and the reservation–based network is much preferable. As the link capacity becomes large, however, an introduction of the reservation–based network is not necessary since the utility difference reaches 0.

If we use the rate–adaptive application, the utility difference becomes smaller than the case of the rigid application (Figure 5), and it seems that the reservation–based network is not necessary. However, its premise is that the link capacity is adequately prepared by estimating the traffic load of real–time applications. It is impossible in the current (and probably future) Internet. Moreover, we can observe from the figure that the worst utilities are much smaller than the average utilities. We need more link capacities to build the high–quality Internet.

We next present the results of the tandem network model analyzed in Section 4.. In the following numerical examples, we fix the traffic loads of connections (Paths 0 through N) to be 10 Erlang while the link capacity C (identically set on all links) and the number of links are changed.

Figures 6 and 7 show utility differences of long–hop connections for rigid and rate–adaptive applications, respectively. The horizontal axis shows the number of hops of long–hop connections. The four values of the link capacity C are used in the figures; 0.5, 1.0, 1.5, and 2 Mbps. From Figure 6, we can observe different behaviors by the link capacity. When the link capacity is not large ($C = 0.5, 1.0$ Mbps), the utility differences are decreasing by the larger number of hop counts. That is, the advantage of the reservation–based service becomes small. It is because the blocking probability of the connection requests is large due to the small link capacity. As the link capacity is adequately provided ($C = 1.5$ Mbps in the current case), the preference of the reservation–based service becomes significant. However, the link capacity becomes large enough

($C = 2$ Mbps), the utility difference again becomes small, and the best–effort service may be a good choice when we consider the introduction cost of the reservation–based service.

The same tendency can be found in Figure 7 which shows the case of the rate–adaptive applications. However, the utility differences are much smaller than the ones of rigid applications (Figure 6) as one may expect. That is, the figure clearly indicates that the rate–adaptive applications help improving QoS of real–time applications in the current Internet which only provides the best–effort service. However, we should again claim that the best–effort service can only improve QoS not guarantee QoS of the established connections, which is an important factor for real–time applications.

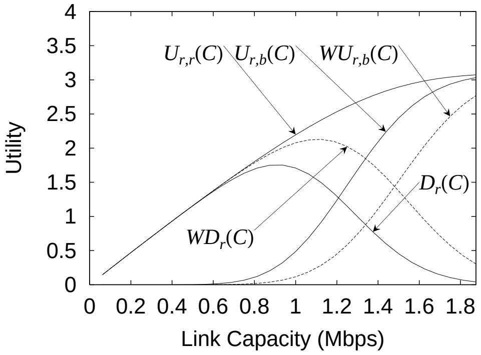


Figure 4: Average and worst utilities of rigid applications

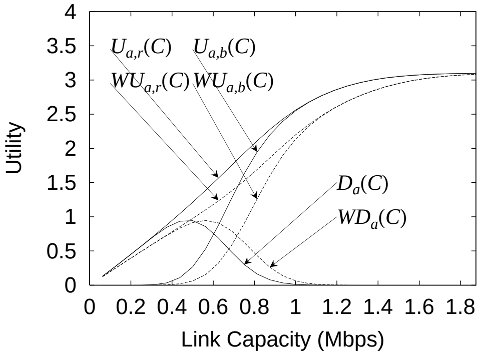


Figure 5: Average and worst utilities of rate–adaptive applications

6. Concluding Remarks

In this paper, we have introduced user's worst utility to quantify user's perceived QoS to compare the QoS capabilities of reservation–based and best–effort services. The network is then extended to the tandem link model. By comparing two services, we have discussed which service gives a better solution for real–time applications. Our observation is that the reservation–based service is necessary unless the adequate network dimensioning framework is provided in the best–effort service. At least now, we do not have a solution, and it requires future research works.

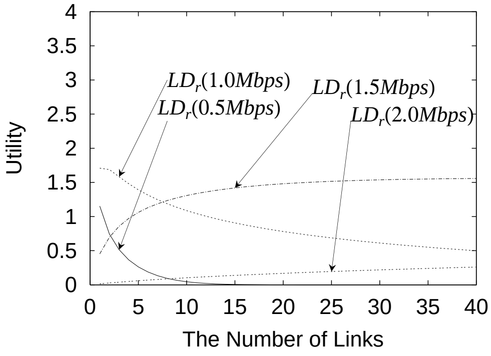


Figure 6: Utility differences of rigid applications

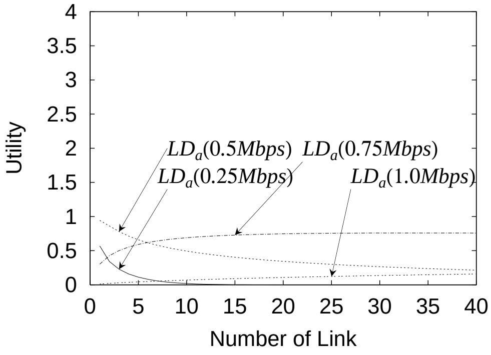


Figure 7: Utility differences of rate–adaptive applications

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture," *Internet RFC 1633*, July 1994.

[2] P. White, "RSVP and integrated services in the Internet: A tutorial," *IEEE Commun. Mag.*, vol. 34, pp. 100–106, May 1997.

[3] A. Parekh and R. Gallanger, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, pp. 344–357, June 1993.

[4] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "MPEG-2 rate control algorithm for ATM networks with bandwidth re-negotiation," in *Proceedings of Fifth IFIP IWQoS*, pp. 291–302, May 1997.

[5] J-C.Bolot and T.Turletti, "Exprerience with control mechanisms for packet video in the Internet," *CCR ACM Sigcom*, vol. 28, January 1998.

[6] A. Watanabe, M. Murata, and H. Miyahara, "Which provides better utility to users, best–effort service or reservation–based service," *Proceedings of ITC-CSCC'99*, July 1999.

[7] D. Bertsekas and R. Callager, *DATA NETWORKS*, ch. 9, pp. 524–529. Prentice-Hall Interrational Edition.