Best-Effort Versus Reservations Revisited

Oliver Heckmann[1] and Jens B. Schmitt[2]

[1] KOM Multimedia Communications Lab, TU Darmstadt, Germany
[2] DISCO Distributed Computer Systems Lab, University of Kaiserslautern, Germany

Abstract. In this paper, we walk in the footsteps of the stimulating paper by Lee Breslau and Scott Shenker entitled "Best-effort vs. Reservations: A Simple Comparative Analysis"[1]. In fact, we finally follow their invitation to use their models as a *starting point* and *extend* them to reason about the very basic but still very much debated architectural issue whether quality of service (QoS) mechanisms like admission control and service differentiation are necessary or if overprovisioning with a single service class does the job just as well at lower system complexity. We analytically compare two QoS systems: a QoS system using admission control and a reservation mechanism that can guarantee bandwidth for flows respectively offers service differentiation based on priority queueing for two service classes and a system with no admission control and a single best-effort service class.

Keywords: Quality of Service, Network Architecture.

1 Prelude

The first set of models we use are based on those by Breslau and Shenker. They assume a single bottleneck and a single type of traffic (elastic, strictly inelastic or adaptive) using the bottleneck and then analyse the expected total utility by assuming a certain probability distribution for the number of flows. The main effects investigated with these models are admission control and bandwidth guarantees. As is common and good practice in sciences, we first reproduce the results of Breslau and Shenker and then give some further insights. The second set of models is an original contribution of this paper. Contrary to the other models, they analyse a given load situation and a *traffic mix consisting of elastic and inelastic flows* filling the link at the same time. By incorporating queueing theory and the TCP formula, the second set of models allows us to investigate more sophisticated utility functions and more realistic network behaviour than the first set. The main effects investigated with these models are scheduling and service differentiation.

2 On the Benefit of Admission Control

Shenker and Breslau [1, 7] analyse two fundamentally different QoS systems in their work:

H. de Meer and N. Bhatti (Eds.): IWQoS 2005, LNCS 3552, pp. 151–163, 2005.

1. A best-effort (BE) system without admission control where all flows admitted to the network receive the same share of the total bandwidth.
2. A reservation based QoS system with admission control, where only the flows are admitted to the network that optimally (w.r.t. total utility) fills the network. Their bandwidth is guaranteed by the system.

We start with a fixed load model that assumes a given traffic load for the network.

2.1 Fixed Load

The fixed load model from [7], also published in [1], assumes that there are a number of identical flows requesting service from a link with capacity C. The utility function $u(b)$ of a flow is a function of the link bandwidth b assigned for that flow with:

$$\frac{du(b)}{db} \geq 0 \;\; \forall b > 0\,,\; u(0) = 0\,,\; u(\infty) = 1 \tag{1}$$

A flow rejected by the admission control is treated as receiving zero bandwidth, resulting in zero utility. The link capacity is split evenly among the flows so that the total utility U of k admitted flows is given by $U(k) = k \cdot u(\frac{C}{k})$

If there exists some $\epsilon > 0$ such that the function $u(b)$ is convex but not concave[1] in the neighbourhood $[0, \epsilon]$, then there exists some k_{max} such that $U(k_{max}) > U(k) \;\; \forall k > k_{max}$. In this case, the network is overloaded whenever more than k_{max} flows enter the network; the system with admission control would yield the higher total utility because it could restrict k_{max}.

If the utility function $u(b)$ is strictly concave, then $U(k)$ is a strictly monotonically increasing function of k. In that case, the total utility is maximised by always allowing flows to the network and not using admission control.

Elastic Applications. typically have a strictly concave utility function as additional bandwidth aids performance but the marginal improvement decreases with b. Therefore, if all flows are elastic, the best-effort system without admission control would be the optimal choice.

Looking at the other extreme of the spectrum, there are **strictly inelastic applications** like traditional telephony that require their data to arrive within a given delay bound. Their performance does not improve if data arrives earlier, they need a fixed bandwidth $\tilde{b}$ for the delay bound. Their utility function is given by

$$u(b) = \begin{cases} 0 & b < \tilde{b} \\ 1 & b \geq \tilde{b} \end{cases} \tag{2}$$

which leads to a total utility of

$$U(k) = \begin{cases} 0 & k > C/\tilde{b} \\ k & k \leq C/\tilde{b} \end{cases} \tag{3}$$

[1] This rules out functions simple linear functions $u(b) = a_0 + a_1 \cdot b$ which would, by the way, also violate (1).

In this case, admission control is clearly necessary to maximise utility. If no admission control is used and the number of flows exceeds the threshold $C/\tilde{b}$, the total utility $U(k)$ drops to zero. The two extreme cases of elastic and strictly inelastic applications show that the Internet and telephone network architectures were designed to meet the needs of their original class of applications.

Another type are the **adaptive applications**; they are designed to adapt their transmission rate to the currently available bandwidth and reduce to packet delay variations by buffering. Breslau/Shenker propose the S-shaped utility function with parameter κ

$$u(b) = 1 - e^{-\frac{b^2}{\kappa+b}} \tag{4}$$

to model these applications. For small bandwidths, the utility increases quadratically ($u(b) \approx \frac{b^2}{\kappa}$) and for larger bandwidths it slowly approaches one ($u(b) \approx 1 - e^{-b}$). The exact shape is determined by κ.

For these flows, the total utility $U(k)$ has a peak at some finite k_{max} but the decrease in total utility for $k > k_{max}$ is much more gentle than for the strictly inelastic applications. The reservation based system thus has an advantage over the best-effort system, but two questions remain: The first is *whether that advantage is large enough to justify the additional complexity* of the reservation based QoS system and the second is, *how likely is the situation where* $k > k_{max}$. These questions are addressed in the next section with the variable load model.

2.2 Variable Load

Model. Breslau and Shenker [1] analyse the likelihood of an overload situation for the strictly inelastic and adaptive applications by assuming a given probability distribution $P(k)$ of the number of flows k. They use two models, a model with a discrete and one with a continuous number of flows k. We base our following analysis on the discrete model and on the algebraic load distribution. [1] also contains results for a Poisson and exponential load distribution, but they do not lead to fundamentally new insights.

For the *algebraic load distribution* $P(k) = \frac{\nu}{\lambda+k^z}$ the load decays at a slower than exponential rate over a large range. It has three parameters ν, λ and z[2]. The algebraic distribution is normalised so that $\sum_{k=0}^{\infty} P(k) = 1$; we analyse $z \in \{2, 3, 4\}$.

Similar to [1], for the following analysis we choose the parameters of the probability distributions so that the expected number of flows $E(k) = \sum_{k=0}^{\infty} k \cdot P(k)$ is 100. For the utility functions, $\tilde{b} = 1$ in (2) and $\kappa = 0.62086$ in (4) as this parameter setting yields $k_{max} = C$ for both utility functions.

The two utility functions analysed should be seen as the extremes of a spectrum. The strictly inelastic utility function does not tolerate any deviation from the requested minimum bandwidth $\tilde{b}$ at all, while the adaptive utility function

[2] λ is introduced so that the distribution can be normalised for a given asymptotic power law z.

embodies fairly large changes in utility across a wide range of bandwidths above and below C/k_{max}.

The expected total utility $\overline{U}_{BE}$ of the best-effort system is

$$\overline{U}_{BE}(C) = \sum_{k=1}^{\infty} P(k) \cdot U(k) = \sum_{k=1}^{\infty} P(k) \cdot k \cdot u(\frac{C}{k}) \tag{5}$$

The QoS system can limit the number of flows to k_{max}. The expected utility $\overline{U}_{QoS}$ of the QoS system is $\overline{U}_{QoS}(C) = \sum_{k=1}^{k_{max}(C)} P(k) \cdot k \cdot u(\frac{C}{k}) + \sum_{k=k_{max}(C)+1}^{\infty} P(k) \cdot k_{max} \cdot u(\frac{C}{k_{max}(C)})$.

To compare the performance of the two QoS systems, the authors of [1] propose the bandwidth gap as a performance metric. The bandwidth gap is the additional bandwidth Δ_C necessary for the best-effort system so that the expected total utilities are equal: $\overline{U}_{QoS}(C) = \overline{U}_{BE}(C + \Delta_C)$

We propose a different metric: the unit-less **overprovisioning factor** OF. It puts the bandwidth gap in relation to the original bandwidth

$$OF = \frac{C + \Delta_C}{C} \tag{6}$$

The overprovisioning factor expresses the bandwidth increase necessary for a best-effort based QoS system to offer the same expected total (respectively average) utility as the reservation based one.

Evaluation. The overprovisioning factors for the strictly inelastic and the adaptive utility function and for the algebraic load distributions over a wide range of link bandwidths C are shown in Fig. 1. The reader is reminded of the fact that the expected number of flows $E(k)$ is 100.

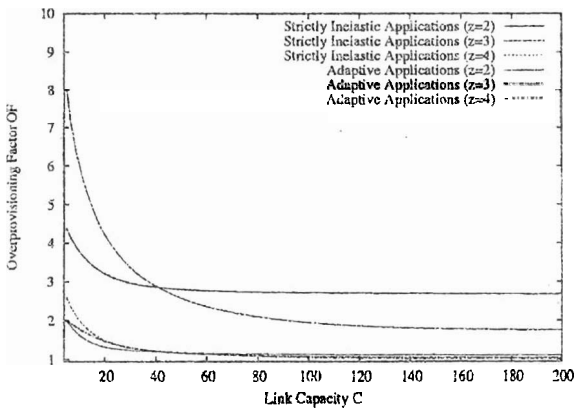


Fig. 1. Results

The algebraic load distribution decays slowly. The lower z, the slower the decay. For the inelastic applications, the very slow decay for $z = 2$ results in a significantly higher overprovisioning factor (2.70 if capacity equals demand and 2.67 if capacity equals twice the demand in the strictly inelastic case) than for the higher values of z (or for the exponential load distribution in [1], where the overprovisioning factor is around 2). For adaptive applications, the overprovisioning factor is close to one (between 1.05 and 1.14 if capacity equals demand).

The results here and in [1] show that the overprovisioning factor is close to unity for adaptive applications and significantly higher than unity for the

inelastic applications. The link capacity significantly influences the performance of both QoS systems and the overprovisioning factor. The reservation based QoS system can provide significant advantages over the pure best-effort system in a well dimensioned network for strictly inelastic applications. For adaptive applications, the advantage is rather low in a well dimensioned network.

2.3 Summary and Conclusions

The analysis above respectively in [1] gives valuable insights but can also be criticised in some points:

- It assumes that only a single type of application utilises the network. If different applications with different requirements utilise a network at the same time, QoS systems can differentiate between them – e.g. by protecting loss sensitive flows or by giving delay sensitive flows a higher scheduling priority – and offer a further advantage over the best-effort system.
- The load distributions (Poisson, exponential, algebraic) used in [1] and above to derive the expected utility for a given bandwidth are not based on empirical studies.
- In addition, it is arguable whether the expected utility really represents the satisfaction of the customers with the network performance:
  If the network performance is very good most of the time but regularly bad at certain times (e.g. when important football games are transmitted), this might be unacceptable for customers despite a good *average* utility.

In the next section, we use a novel approach to avoid these drawbacks and shed more light on the comparison of the two QoS systems.

3 On the Benefit of Service Differentiation

When analysing a mix of different traffic types competing for bandwidth, it is not trivial to determine the amount of bandwidth the individual flows will receive and the delay it experiences. In this section, we present an analytical approach that – contrary to the previous approach – uses queueing theory and the TCP formula as a foundation to calculate the overprovisioning factor for a traffic mix of elastic TCP-like traffic flows and inelastic traffic flows.

3.1 Traffic Types

We assume that two types of traffic – elastic and inelastic – share a bottleneck link of capacity C. For **inelastic traffic**, we use index 1 and assume that there are a number of inelastic flows sending with a total rate r_1. The strictly inelastic traffic analysed in Section 2 did not tolerate any loss. Most multimedia applications, however, can tolerate a certain level of loss. For example, a typical voice transmission is still understandable if some packets are lost – albeit at reduced

quality. We model this behaviour here by making the utility of the inelastic traffic degrading with the packet loss[3] and with excessive delay.

For the **elastic traffic,** we use index 2; it represents file transfer traffic with the characteristic TCP "sawtooth" behaviour: the rate is increased proportional to the round-trip time and halved whenever a loss occurs. We use a TCP formula to model this behaviour; the two main parameters that influence the TCP sending rate are the loss probability p_2 and the RTT respectively the delay q_2. We assume there are a number of greedy elastic flows sending as fast as the TCP congestion control is allowing them to send; their total rate is $r_2 = f(p_2, d_2)$. The utility of the elastic traffic is a function of its throughput.

3.2 Best-Effort Network Model

A best-effort network cannot differentiate between packets of the elastic and inelastic traffic flows and treats both types of packets the same way. The loss and the delay for the two traffic types is therefore equal $p_{BE} = p_1 = p_2$, $q_{BE} = q_1 = q_2$.

Let μ_1 be the average service rate of the inelastic flows, μ_2 the one for elastic flows, λ_1 the arrival rate of the inelastic traffic and λ_2 accordingly the arrival rate of the elastic traffic. The total utilisation ρ is then given by $\rho = \rho_1 + \rho_2 = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2}$ and the average service rate $\overline{\mu}$ by $\overline{\mu} = \frac{\rho_1\mu_1 + \rho_2\mu_2}{\rho_1+\rho_2} = \frac{\lambda_1+\lambda_2}{\rho_1+\rho_2}$.

In the best-effort model, the loss probability p_{BE} is the same for both traffic types and can be estimated with the well-known $M/M/1/B$ loss formula for a given maximal queue length of B packets assuming Markovian arrival and service processes [2]: $p_{BE} = \frac{1-\rho}{1-\rho^{B+1}} \cdot \rho^B$.

For the queueing delay q_{BE} of the bottleneck link, the $M/M/1/B$ delay formula [2] is used: $q_{BE} = \frac{1/\overline{\mu}}{1-\rho} \cdot \frac{1+B\rho^{B+1}-(B+1)\rho^B}{1-\rho^B}$

The arrival rate λ_1 of the inelastic traffic is given by the sending rates r_1 of the inelastic flows (15) while the arrival rate λ_2 of the elastic traffic depends on the TCP algorithm and the network condition. There are several contributions like [5, 6] that describe methods for predicting the average long-term TCP throughput, depending on the loss and delay properties of a flow. For our high-level analysis, we are not interested in details like the duration of the connection establishment, etc. Therefore, we use the plain square-root formula of [3] for this analysis; it allows us to keep the complexity of the resulting model low:

$$\text{throughput} = \frac{\text{MSS}}{\text{RTT} \cdot \sqrt{2/3} \cdot \sqrt{p_2}} \tag{7}$$

with MSS as maximum segment size and RTT as the round trip time. RTT is assumed to be dominated by the queueing delay q_2. The throughput of the queue

[3] It can be seen as an intermediate application between the strictly inelastic and the adaptive traffic of Section 2.

can also be expressed as a function of the arrival process λ_2 and the loss probability p_2:

$$\text{throughput} = \lambda_2(1 - p_2) \tag{8}$$

Introducing parameter t that we call *flow size factor*, (7) and (8) can be simplified to $\lambda_2 = \frac{t}{q_{BE}\cdot\sqrt{p_{BE}}} \cdot \frac{1}{1-p_{BE}}$ t encompasses the $MSS/\sqrt{2/3}$ part of (7) and part of the round-trip-time and is used to put the TCP flows in correct dimension to the inelastic flows which are dimensioned by their fixed sending rate r_1.

As λ_2 is a function of p_{BE} and q_{BE} and at the same time influences p_{BE} and q_{BE}, the network model is a non-linear equation system (see Model 1). It can be solved with standard methods.

3.3 QoS Network Model

To model a QoS system that differentiates between the inelastic and elastic traffic, we use priority queueing. The inelastic traffic receives strict non-preemptive priority in time and (buffer) space over the elastic traffic.

Using the *M/M/1* queueing model the expected waiting time $E(W_1)$ for a packet of an inelastic flow depends on the expected number of packets waiting to be served $E(L_1)$ and the residual service time of the packet currently in the queue. Because non-preemptive queueing is used, the latter can be a type 1 (inelastic flow) or type 2 (elastic flow) packet; because the exponential service time distribution is memoryless, the expected residual service time is $\sum_{i=1}^{2} \rho_i \frac{1}{\mu_i}$:

$$E(W_1) = E(L_1)\frac{1}{\mu_1} + \sum_{i=1}^{2} \rho_i \frac{1}{\mu_i} \tag{9}$$

By applying Little's Law [4] $E(L_i) = \lambda_i E(W_i)$, we get $E(W_1) = \frac{\sum_{i=1}^{2} \rho_i \frac{1}{\mu_i}}{1-\rho_1}$.

To determine the average queueing delay q_1, we need the expected sojourn time $E(S_1) = E(W_1) + 1/\mu_1$: $q_1 = E(S_1) = \frac{1/\mu_1 + \rho_2/\mu_2}{1-\rho_1}$

For the second queue, the determination of the expected sojourn time is more complicated. The expected waiting time $E(W_2)$ and the sojourn time $E(S_2) = q_2$ for a packet of type 2 is the sum of

- the residual service time $T_0 = \sum_{i=1}^{2} \rho_i \frac{1}{\mu_i}$ of the packet currently in the queue because the queue is non-preemptive,
- the service times $T_1 = E(L_1)/\mu_1$ for all packets of priority 1
- and the service times $T_2 = E(L_2)/\mu_2$ for all packets of priority 2 that are already present waiting in the queue at the point of arrival of the new packet of type 2 and are therefore served before it
- plus the service times $T_3 = \rho_1(T_0 + T_1 + T_2)$ for all packets of priority 1 that arrive during $T_0 + T_1 + T_2$ and that are served before the packet of type 2 because they are of higher priority.

The waiting time is $E(W_2) = T_0 + T_1 + T_2 + T_3$, for the sojourn time respectively queueing delay the service time has to be added $q_2 = E(S_2) = E(W_2) + 1/\mu_2$. By applying (9) and Little's Law [4] we get $q_2 = E(S_2) = \frac{(1+\rho_1)\sum_{i=1}^{2} \rho_i \frac{1}{\mu_i}}{(1-\rho_1-\rho_1\rho_2)(1-\rho_1)} + \frac{1}{\mu_2}$.

A packet of type 1 is not dropped as long as there are packets of type 2 waiting in the queue that could be dropped instead. With respect to loss, the arrival process 1 with arrival rate λ_1 thus experiences a normal $M/M/1/B$ queue with a loss probability for a packet of type 1 of $p_1 = \frac{1-\rho_1}{1-\rho_1^{B+1}} \cdot \rho_1^B$.

We make the simplifying assumption that λ_1 is small enough so the loss for queue 1 is negligible $p_1 \approx 0$. For the low priority queue, the loss probability is then given by

$$p_2 = \frac{(1-\rho_1-\rho_2)}{1-(\rho_1+\rho_2)^{B+1}} \cdot (\rho_1+\rho_2)^B \cdot \frac{\lambda_1+\lambda_2}{\lambda_2} \tag{10}$$

The first part of (10) represents the total loss of the queueing system; the second part $\frac{\lambda_1+\lambda_2}{\lambda_2}$ is necessary because the packets of type 2 experience the complete loss.

The priority queueing based QoS network model is summarised in Model 2, it is using the same parameters as Model 1. Like the best-effort network model, it is a non-linear equation system.

3.4 Utility Functions

Inelastic Traffic. The inelastic traffic represents multimedia or other real-time traffic that is sensitive to loss and delay. Therefore, the utility u_1 of the inelastic flows is modelled as strictly decreasing function of the loss probability p_1 and the deviation of the delay q_1 from a reference queueing delay q_{ref}: $u_1 = 1 - \alpha_p p_1 - \alpha_q \frac{q_1 - q_{ref}}{q_{ref}}$.

As a reference queueing delay q_{ref} we use the queueing delay (19) of the QoS network model as that is the minimum queueing delay achievable for this traffic under the given circumstances (number of flows, link capacity, non-preemptive service discipline, etc).

Elastic Traffic. The elastic traffic represents file transfer traffic. The utility of this traffic depends mostly on the throughput as that determines duration of the transfer. The utility u_2 is therefore modelled as function of the throughput d_2: $u_2 = \beta \cdot d_2 = \beta \cdot \frac{t}{q_2 \cdot \sqrt{p_2}}$.

We determine the parameter β so that $u_2 = 1$ for the maximum throughput that can be reached if $\lambda_1 = 0$; both network models lead to the same β if there is no inelastic traffic.

3.5 Evaluation

The default parameter values we use for the following evaluation are given in Table 1. The effect of parameter variation is analysed later. The motivation behind the utility parameter α_p is that the utility of the inelastic flows should be zero for 10% losses (if there is no additional delay); for the parameter α_q the motivation is that the utility should be zero if the delay doubles compared to the minimal delay of the QoS system. β is chosen so that the utility of the elastic

Table 1. Default Parameter Values for the Evaluation

Parameter	Value
μ_1	83.3 pkts/s
μ_2	same as μ_1
α_q	1
α_p	10
β	see Section 3.4
B	10 pkts
t	t_0, $5t_0$, $10t_0$
r_1	[0, ..., 40] pkts/s
w_1	[1, 2, 5]
w_2	1

flow is 1 for the maximum throughput as explained in Section 3.4. During the evaluation we vary w_1, r_1 and t. For the choice of w_1, we assume that for the total utility evaluation, the inelastic flows are more important than the elastic flows because they are given priority over the elastic flows and it seems reasonable to expect users to also have a higher utility evaluation for one real-time multimedia flow (e.g. a phone call) than for a file transfer. An indication for that is the fact that the price per minute for a phone call nowadays is typically much higher than the price per minute for a dial-up Internet connection used for a file transfer. As evaluation metric we again use the **overprovisioning factor**[4].

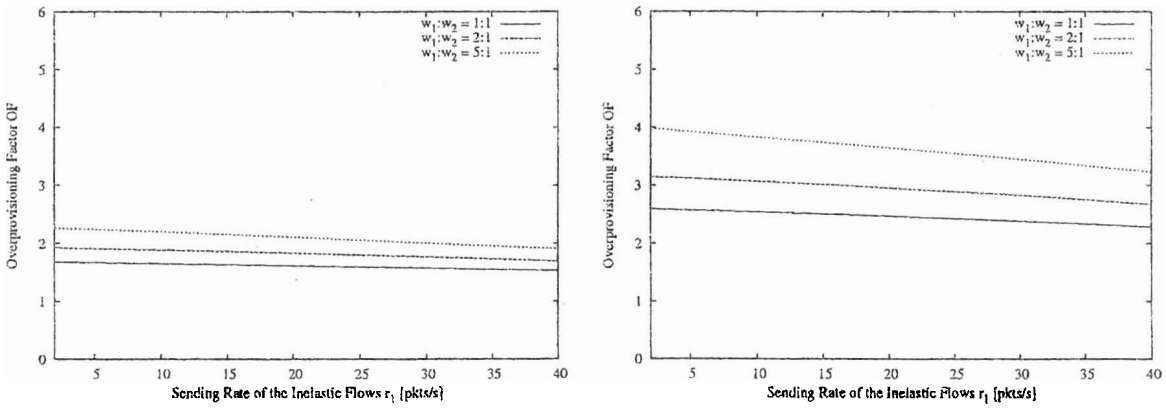
Basic Results. The overprovisioning factors OF for different flow size factors[5] t and for different weight ratios $w_1 : w_2$ are depicted on the y-axis in the graphs of Fig. 2. The total sending rate r_1 of the inelastic flows is shown on the x-axis.

[4] For a given r_1 and t, we determine the solution vector (p_1, q_1, p_2, q_2) of the QoS network Model 2. The utility values $u_1 = f(p_1, q_1)$ and $u_2 = f(p_2, q_2)$ and the weighted average utility U_{ref} are derived from the solution vector with $w_1, w_2 > 0$: $U_{ref} = \frac{w_1 u_1(p_1,q_1)+w_2 u_2(p_2,q_2)}{w_1+w_2}$

The best-effort system based on Model 1 is overprovisioned by a factor OF. The bandwidth respectively service rates μ_1 and μ_2 are increased by that factor OF. Additionally, the buffer space B is increased by the same factor. U_{ref} is used as a reference value and OF is increased by a linear search algorithm until $U_{BE}(OF^*) = U_{ref}$.

[5] To derive an anchor point for t, we arbitrarily determine a t_0 that leads to $\rho_1 = 20\%$ and to $\rho_2 = 60\%$ using the QoS network model. This represents a working point with $\lambda_1 = 0.2 \cdot \mu_1$ with a total utilisation of 80%. Every fourth packet is a multimedia packet, creating a typical situation where a QoS system would be considered. If t is increased to $t = 5t_0$ and λ_1 kept constant, then the proportion of of multimedia packet to file transfer packet drops to 1 : 3.4. At the same time, the aggressiveness of TCP against the inelastic flows increases in the best-effort network model as can be seen in the evaluation results below (Fig. 2).

As can be seen from the graphs, the higher the ratio $w_1 : w_2$ is – that is, the more important the inelastic flows are for the overall utility evaluation – the higher the overprovisioning factor becomes. This can be expected, because for small overprovisioning factors the utility u_1 of the inelastic flows is smaller in the best-effort system than the QoS system where they are protected from the elastic flows because they experience more loss and delay. Thus, the higher u_1 is weighted in the total utility function U, the more bandwidth is needed in the best-effort system to compensate this effect.



(a) Flow Size Factor $t = t_0$ (b) $t = 5t_0$

Fig. 2. Overprovisioning Factors for the Configuration of Table 1

Comparing the two graphs, it can be seen that as the flow size factor is increased more overprovisioning is needed. Increasing the flow size factor represents increasing the number of elastic (TCP) senders and the aggressiveness of the elastic flows. In the best-effort system where the inelastic flows are not protected, a higher flow size factor increases the sending rate of the elastic flows on cost of additional loss and delay for the inelastic flows that in return has to be compensated by more capacity leading to a higher overprovisioning factor.

Keeping the flow size factor constant, with an increase of the sending rate r_1 the overprovisioning factor decreases; the decrease is stronger the higher the flow size factor is. For a weight ratio of $w_1 : w_2 = 2 : 1$ for example the overprovisioning factor drops from $r_1 = 2$ to 40 by 12.0% for $t = t_0$ and 14.9% for $t = 5t_0$. This phenomenon can be explained the following way: When comparing the resulting utility values u_1 and u_2 of the QoS system with the best-effort system ($OF = 1$), the utility value of the inelastic flows u_1 drops because they are no longer protected. At the same time, the utility value of the elastic flows u_2 increases because they no longer suffer the full loss. The increase of u_2 is stronger than the decrease of u_1 the higher r_1 is, therefore for higher r_1 less overprovisioning is needed.

The following discussions – unless stated otherwise – are based on a weight ratio $w_1 : w_2 = 2 : 1$ and a flow size factor of $t = 5t_0$.

Different Bottleneck Resources. Increasing the buffer space B has two adverse effects; it decreases the loss rate and increases the potential queueing delay. An increase of B results in an increase of the overprovisioning factor OF. This is an indication that for the utility calculation, the queueing delay has a stronger effect than the loss rate. This is not surprising because for the $M/M/1/B$ formulas, the loss becomes quickly negligible for larger B.

To confirm this, we reduced the queueing delay effects by setting $\alpha_q = 0.05$ and repeated the experiment. Now, with an increase of B from 10 over 15 to 20 the adverse effect can be observed: the overprovisioning factor drops from 1.76 over 1.68 to 1.66 for $r_1 = 10$.

To conclude, the effect of the buffer size depends on the ratio of α_p to α_q in the utility function.

Next, the reference buffer space B and at the same time the bandwidth (respectively the service rates μ_1 and μ_2) are doubled; r_1 was increased accordingly.

Compared to the previous experiment, the overprovisioning factors only increased insignificantly for $t = 5t_0$. In the best-effort system – as can be seen from (14) – for large B, the queueing delay q_{BE} becomes inverse proportional to the service rate $\overline{\mu}$ and therefore the bandwidth. For large B, the loss p_{BE} exponentially approaches zero as can be seen from (13). Via (16), this leads to a massive increase the elastic rate λ_2 and overall utilisation ρ. This explains why the buffer space has a larger influence than the service rate. Similar arguments hold true for the QoS system.

3.6 Conclusions

The experiments of this section evaluated the relative performance advantage of a QoS system offering service differentiation over a plain best-effort system. The systems have two resources, buffer and bandwidth. We used two types of traffic - elastic and inelastic traffic - which share a bottleneck link. The evaluation is based on an aggregated utility function. Our results are overprovisioning factors that show how much the resources (bandwidth and buffer) of the best-effort system have to be increased to offer the same total utility that the QoS system provides.

Compared to the approach of Breslau and Shenker (Section 2), the overprovisioning factors of the models in this section are generally higher. This is explained by the fact that the models of Section 2 do not consider different traffic types sharing the bottleneck resources. Therefore, they miss one important aspect of QoS systems which is service differentation between flow classes.

In today's Internet the overwhelming part of the traffic is TCP based file transfer traffic. As realtime multimedia applications spread and are supported, their initial share of traffic will be low. In our models this can be represented by rather low sending rates r_1 (few inelastic flows) and a high flow size factor

t (many elastic flows). Interestingly, our results show that especially for this combination the overprovisioning factors are the highest. Therefore, to support the *emerging* realtime traffic applications, QoS architectures have their greatest advantages.

4 Caveat

Both sets of models in this paper necessarily have their limitations because they are based on analytical methods that by nature only allow a certain degree of complexity to be still solvable. The influence of the network topology has been neglected so far. Neither of the approaches uses a fully realistic traffic model that accounts for packet sizes, realistic variability of the packet interarrival times and so on.

Discussions may go on ...

Model 1. Best-Effort Network Model

r_1	Total sending rate of the inelastic flows [pkts/s] (given)
t	Flow size factor of the elastic flows [pkts] (given)
μ_1	Service rate of the inelastic traffic [pkts/s] (given)
μ_2	Service rate of the elastic traffic [pkts/s] (given)
B	Queue length [pkts] (given)
p_{BE}	Loss probability
q_{BE}	Queueing delay [s]
λ_1	Arrival rate of the inelastic traffic at the bottleneck [pkts/s]
λ_2	Arrival rate of the elastic traffic at the bottleneck [pkts/s]
ρ	Utilisation of the queue
$\overline{\mu}$	Average service rate [pkts/s]

Equations

$$\overline{\mu} = \frac{\lambda_1 + \lambda_2}{\rho} \tag{11}$$

$$\rho = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} \tag{12}$$

$$p_{BE} = \frac{1-\rho}{1-\rho^{B+1}} \cdot \rho^B \tag{13}$$

$$q_{BE} = \frac{1/\overline{\mu}}{1-\rho} \cdot \frac{1 + B\rho^{B+1} - (B+1)\rho^B}{1-\rho^B} \tag{14}$$

$$\lambda_1 = r_1 \tag{15}$$

$$\lambda_2 = \frac{t}{q_{BE} \cdot \sqrt{p_{BE}}} \cdot \frac{1}{1-p_{BE}} \tag{16}$$

Model 2. QoS Network Model

p_1	Loss probability of the inelastic flows
q_1	Queueing delay of the inelastic flows [s]
p_2	Loss probability of the elastic flows
q_2	Queueing delay of the elastic flows [s]
ρ_1	Utilisation of the queue with inelastic flows
ρ_2	Utilisation of the queue with elastic flows

Equation (15) and

$$\rho_1 = \lambda_1/\mu_1 \tag{17}$$

$$\rho_2 = \lambda_2/\mu_2 \tag{18}$$

$$q_1 = \frac{1/\mu_1 + \rho_2/\mu_2}{1-\rho_1} \tag{19}$$

$$q_2 = \frac{(1+\rho_1)\sum_{i=1}^{2}\rho_i\frac{1}{\mu_i}}{(1-\rho_1-\rho_1\rho_2)(1-\rho_1)} + \frac{1}{\mu_2} \tag{20}$$

$$p_1 = \frac{(1-\rho_1)}{1-\rho_1^{B+1}}\cdot\rho_1^B \approx 0 \tag{21}$$

$$p_2 = \frac{(1-\rho_1-\rho_2)}{1-(\rho_1+\rho_2)^{B+1}}\cdot(\rho_1+\rho_2)^B\cdot\frac{\lambda_1+\lambda_2}{\lambda_2} \tag{22}$$

$$\lambda_2 = \frac{t}{q_2\cdot\sqrt{p_2}}\cdot\frac{1}{1-p_2} \tag{23}$$

References

1. Lee Breslau and Scott Shenker. Best-Effort versus Reservations: A Simple Comparative Analysis. In *Proceedings of the ACM Special Interest Group on Data Communication Conference (SIGCOMM 1998)*, pages 3–16, October 1998.
2. Leonard Kleinrock. *Queueing Systems – Theory.* Wiley-Interscience, New York, Vol. 1, 1975. ISBN: 0471491101.
3. T. V. Lakshman and U. Madhow. The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. *IEEE/ACM Transactions on Networking*, 5(3):336–350, 1997.
4. J. D. Little. A proof of the queueing formula $l = \lambda w$. *Operations Research*, 9(3):383–387, March 1961.
5. M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *Computer Communication Review*, 27(3), July 1997.
6. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings of the ACM Special Interest Group on Data Communication Conference (SIGCOMM 1998)*, pages 303–314, September 1998.
7. Scott Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, September 1995.