Analysis of Queuing Delay in Multimedia Gateway Call Routing

Qiwei Huang
UTStarcom Inc, 33 Wood Ave. South
Iselin, NJ 08830, U.S.A

Errol Lloyd
Computer and Information Sciences Department,
Univ. of Delaware, Newark, DE 19716, U.S.A.

Abstract: *The recent emergence of multimedia applications has led to the deployment of various telecommunication technologies (e.g. IP, wireless) in addition to traditional PSTN (Public Switched Telephone Network) circuit switched networks. In this heterogeneous telecommunication world, users make calls without regard to the fact that the originating network may differ from the terminating network, and the call may be routed through one or more intermediate networks. For instance, users from two different PSTN networks may communicate through an IP network using Voice over IP (VoIP) technology. In this framework, multimedia gateways perform call routing, call signaling conversions and media format conversions among different networks. This paper investigates algorithms and queuing delays on multimedia gateways in regard to call routing aimed at minimizing the expected call delay. Both centralized and distributed queuing models are considered, and performance measures are derived.*

Keywords: multimedia gateway, VoIP, call routing, queuing delay, optimization.

1 Introduction

In telecommunications, multimedia gateways have recently emerged as a tool for providing real-time, multi-way communication among different media (e.g. IP, PSTN, wireless etc). The functionalities of multimedia gateways include call routing, call signaling conversions and media conversions [1]. Among these functionalities, this paper focuses on expected queuing delays associated with call routing. In the remainder of this section, we briefly describe multimedia gateway architectures and functionalities. In section 2 we describe the problem that we consider. In section 3 we present our algorithms and analysis.

1.1 General Background

Figure 1 Multimedia gateway interconnections.

A typical multimedia gateway interconnection is shown in Figure 1. There, two multimedia gateways, NYC and LA, are interconnected by three *media networks*: IP, PSTN and wireless. Each end user is connected to a single local network. Each local network is connected to a single gateway.

Multimedia gateways perform three high-level functionalities: *call signaling conversions, media conversions* and *call routing*. Specifically, call signaling conversions are to convert signaling messages in one type of network (such as VoIP call signaling H.225 and H.245) to the signaling messages in another type of network (such as PSTN call signaling Q.931). Similarly, media conversions are to convert the media format provided in one type of network (say, PSTN DS0) to the media format required in another type of network (such as VoIP RTP packets with different codecs). Since call signaling conversions and media conversions are not relevant to this paper, interested readers can refer to [1] and [2] for additional details. This paper studies call routing as detailed below.

1.2 Call Routing

When a call arrives at a multimedia gateway from a local media network, that gateway must allocate that call to one of the non-local media networks associated with the gateway, this is the functionality of call routing. Here, a *channel* is a physical resource that transmits and receives voice/data information. Each call is processed by one channel, the nature of which is network specific: In an IP network, a *channel* refers to a DSP (digital signal processor) channel; In a PSTN network, a *channel* refers to a DS0 channel; In a

wireless network, a *channel* refers to a radio channel. The (gateway) *bandwidth* is the total number of calls that can be processed simultaneously and is equal to the total number of available channels taken over all media. Each media's *bandwidth* is the number of *channels* available on that media. The number of simultaneous calls on each media can't exceed its *bandwidth*.

On multimedia gateways, channels on each media are usually divided statically into two groups. One group is used for incoming calls and the other group is used for outgoing calls. We assume that there is no delay to process incoming calls from other gateways and local networks: incoming calls are either accepted or rejected instantaneously depending on channels' availability. From this assumption, the queuing delay of a multimedia gateway is independent of the queuing delay on the other gateways. Thus, throughout this paper, our focus is on a single multimedia gateway's call routing in its non-local media networks.

1.3 Centralized Queuing Model vs Distributed Queuing Model

Figure 2 illustrates the internal architecture of a multimedia gateway. A central controller is connected to each media controller through a control bus. Each media controller has its physical media interface connected to the media network, and it is responsible for the media access control. The central controller is responsible for the operation, administration and management of the entire system, and it is also responsible for the call routing. Queuing delay occurs when a call can't be processed immediately upon its arrival. There are two types of queuing models: One model is *centralized queuing*: the queuing of the calls is centralized at the central controller. The central controller decides when to route and process a call on which media depending on real time information about channel utilization as provided by each media controller. Specifically, each media controller informs the central controller (through the control bus) as soon as a call on that media is completed. The other queuing model is *distributed queuing*: here, the queuing of the calls is distributed into each media controller. Each media controller decides when to process the calls routed to it (by the central controller) based on its own channel utilization. In section 3.2, we will present two routing algorithms for these two queuing models respectively.

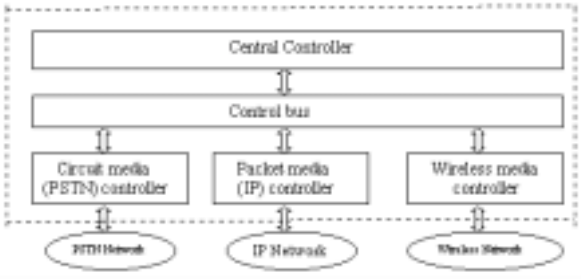

Figure 2 Multimedia gateway internal architecture.

2 Problem Descriptions

As noted above, the *call routing* function of a multimedia gateway is to decide which media will process each call. Relative to call processing, in this paper we make the following standard assumptions: each call should be processed by exactly one channel and cannot be pre-empted; one channel can process only one call at a time; once a call is routed to a media, the call is processed immediately if there is an idle channel, otherwise, the call will be placed into the waiting queue and is processed later when a channel becomes idle on a first come, first served basis.

Considering a single multimedia gateway, the goal of routing is to minimize the expected queuing delay of calls. Associated with this problem, there are two sets of system parameters. One set of parameters is associated with calls: the call arrival rate is a Poisson process with parameter λ and each call has an exponential call length with parameter μ. We assume that the buffer size of the queue (for calls that cannot be processed immediately) is infinite. The other set of parameters is associated with media: there are K non-local media networks connected to the multimedia gateway; the bandwidth of M_i is B_i, which is the maximum number of calls that can be processed by M_i at any given time of instance. Let $B = \sum_{i=1}^{K} B_i$ be the total bandwidth of all media.

Relative to centralized queuing model and distributed queuing model, a *centralized queuing* gateway has one central queue in the central controller (Figure 3). The central controller routes a call immediately if there is an idle channel on any of its media; otherwise, it places the call into the

central queue and routes it later on a first come first served basis as soon as a channel becomes available on any of its media. A *distributed queuing* gateway has one queue in each of its media controllers (Figure 4). After the central controller routes a call to one of the media controllers, the media controller processes the call immediately if there is an idle channel on that media; otherwise it places the call into its queue and processes it later on a first come first served basis as soon as a channel on that media becomes available.

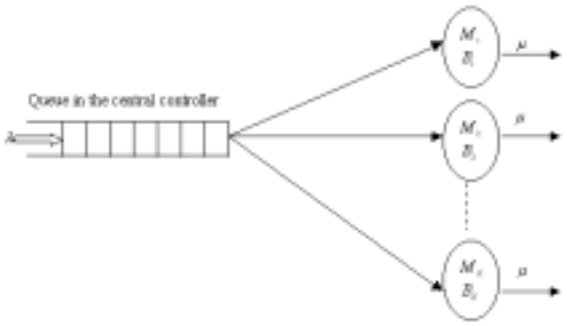

Figure 3 Centralized queuing model.

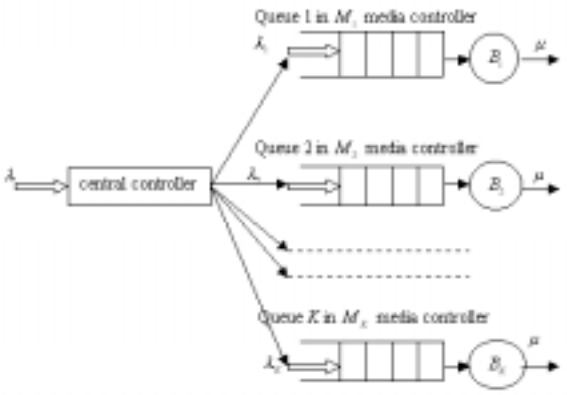

Figure 4 Distributed queuing model.

3 Routing Performance Results

In this section, we present our main results. In section 3.1, we outline several useful results with respect to *M/M/m* queues. In sections 3.2 and 3.3, we describe two routing algorithms: a greedy algorithm and a traffic splitting algorithm. These two algorithms are designed respectively for the *centralized queuing* model and the *distributed queuing* model. Sections 3.2 and 3.3 also contain analysis of the algorithm performance, and for traffic splitting, a determination of optimal parameters for use in that algorithm. In section 3.4, we compare the performance of the two algorithms.

3.1 Properties of an M/M/m Queue

In this section, we provide (without proof, due to space) some queuing properties for *M/M/m* queues.

Recall that the call arrival rate is a Poisson process with parameter λ and that each call has an exponential call length with the parameter μ. Thus, an *M/M/m* queue [3] models a multimedia gateway that consists of a single queue and can simultaneously process a maximum of $m (m \geq 1)$ calls. Let P_i be the probability that there are *i* calls in the system. The expected waiting time *W* of a call in the queue is given by [3]:

$$W = P_0 \frac{m^m \rho^{m+1}}{\lambda m!(1-\rho)^2} \quad (1.1)$$

Where $\rho = \lambda / \mathrm{m}\mu < 1$ (1.2)

and $P_0 = \left(\frac{m^m \rho^m}{m!(1-\rho)} + \sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!} \right)^{-1}$ (1.3)

Suppose that m and μ are fixed and ρ is a variable. From (1.2), λ is also a variable, and *W* is a function of ρ (or λ) for the given m and μ. The relationship between *W* and ρ is as follows:

Theorem (1.4) $\frac{dW(\rho)}{d\rho} > 0, \frac{d^2W(\rho)}{d\rho^2} > 0 \ (\rho < 1)$.

Corollary (1.5) $\frac{dW(\lambda)}{d\lambda} > 0, \frac{d^2W(\lambda)}{d\lambda^2} > 0 \ (\lambda < \mathrm{m}\mu)$.

Theorem (1.4) and corollary (1.5) show that for fixed m and μ, *W* and $dW/d\rho$ are monotone increasing functions with respect to ρ (ρ<1) and λ (λ<mμ). Now suppose that ρ and μ are fixed, thus λ / m is fixed. From (1.1) and (1.2), $W = P_0 \frac{m^{m-1}\rho^m}{\mu m!(1-\rho)^2}$. Thus *W* is a function of *m*.

Theorem (1.6) If ρ (ρ<1) and μ are fixed, then $W(m)$ is a monotone decreasing function. Let $D(m) = |W(m+1) - W(m)|$ be the changing rate of $W(m)$, then $D(m)$ is monotone decreasing.

Theorem (1.6) shows that for the fixed m and μ, the queuing delay $W(m)$ and the changing rate $D(m)$ of $W(m)$ decrease when *m* and λ increase proportionally (see also *Example II* in section 3.3.2).

3.2 A Greedy Algorithm for Centralized Queuing Model

In this section we describe a simple algorithm for centralized queuing model, derive the expected queuing delay for that algorithm, and prove that the algorithm minimizes the expected queuing delay among all algorithms.

3.2.1 The Greedy Algorithm

A natural approach to centralized queuing model is to utilize a traditional greedy algorithm. Here, upon the arrival of a call, if there is an idle channel on some media, the algorithm routes the call to that media where the call can be processed immediately; otherwise, the call is placed into the central queue and is routed later on a first come first served basis when a channel becomes idle. Specifically:

Algorithm (2.1) Greedy_Routing()

```
{
  while (1)
    if (there is an incoming call)
      if (the central waiting queue is not empty)
          put the call into the central queue;
      else if there is an idle channel on any media
          Routing (the call);
    else put the call into the central waiting queue;

    while (there are idle channels due to call
    completion and the waiting queue is not empty)
      get one call from the central waiting queue;
      Routing (the call);
}
Routing (the call)
{
   for i = 1 to K do
      if there is an idle channel in M_i
          route the call to M_i
}
```

3.2.2 Performance Analysis

Recall that $B = \sum_{i=1}^{K} B_i$ is the total bandwidth of all media. If we let $m=B$, from the greedy algorithm and section 3.1, we know that the expected queuing time of a call is the same as the expected queuing time for an *M/M/m* queue. Thus, we have the following theorem:

Theorem (2.2) Let W_G be the expected queuing time of a call generated by *Greedy_Routing*. Then $W_G = P_0 \frac{m^{m-1}\rho^m}{\mu m!(1-\rho)^2}$ with $\rho = \frac{\lambda}{m\mu} < 1$ and $P_0 = \left(\frac{m^m \rho^m}{m!(1-\rho)} + \sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!} \right)^{-1}$.

Theorem (2.2) shows that the expected queuing time of a call W_G depends only on λ, μ and the total bandwidth $m=B$, but not on each individual media's bandwidth:

Example I: $K = 2,\ m_1 = 1,\ m_2 = 2,\ \lambda = 2,\ \mu = 1$.

$\rho = \frac{\lambda}{m\mu} = \frac{\lambda}{(m_1 + m_2)\mu} = \frac{2}{3} < 1$. Thus, $W_G = 4/9$.

Although the greedy algorithm seems apparent, it is in fact optimal as the next result shows:

Theorem (2.3) Among all algorithms for centralized queuing gateway, the greedy algorithm minimizes the expected queuing delay. (The proof is omitted due to space).

3.3 The Traffic Splitting Algorithm for Distributed Queuing Model

A "traffic splitting" algorithm is used in *distributed queuing* gateways, which can be modeled as a combination of K $M/M/B_i$ $(i = 1, 2, ..., K)$ queues. The central controller routes the calls (*splits* the traffic) to each media controller with *predetermined probability* to achieve the expected minimum delay. In this section, we first show how to derive the *predetermined probability* to achieve the expected minimum and prove its correctness. Then we describe the traffic splitting algorithm. Finally we present some examples.

3.3.1 Calculation of Predetermined Probabilities

The analysis below focuses on how to obtain the *predetermined probabilities* to minimize the expected delay from the distribution of call arrival rate and call length as well as the bandwidth of each media. Let $W_i (i = 1, 2, ...K)$ be the expected waiting time for a call routed to the media M_i, and W_S be the expected waiting time of a call:

$$W_S = \sum_{i=1}^{K} p_i W_i = \frac{1}{\lambda} \sum_{i=1}^{K} \lambda_i W_i \qquad (3.1)$$

It is obvious that $\sum_{i=1}^{K} p_i = 1$ (3.2)

It follows that $\lambda_i = \lambda p_i$ (3.3)

and $\sum_{i=1}^{K} \lambda_i = \lambda$ (3.4)

We need to find the probability vector $(p_1, p_2, ...p_K)$, or equivalently $(\lambda_1, \lambda_2, ...\lambda_K)$, to minimize W_S. Let media M_i's bandwidth $B_i = m_i$ and $m = \sum_{i=1}^{K} m_i$. Suppose that we have found such a vector. Since the arrival rate of a call at media M_i is a Poisson process with rate $\lambda_i = \lambda p_i$ [3], W_i can be decided according to (1.1), (1.2) and (1.3):

$$W_i = \frac{1}{\frac{m_i^{m_i}\rho_i^{m_i}}{m_i!(1-\rho_i)} + \sum_{n=0}^{m_i-1}\frac{(m_i\rho_i)^n}{n!}} \frac{m_i^{m_i}\rho_i^{m_i+1}}{\lambda_i m_i!(1-\rho_i)^2} \quad (3.5)$$

where $\rho_i = \lambda_i / \mathrm{m}_i\mu < 1$ (3.6)

From (3.4) and (3.6), $\lambda < \mu\sum_{i=1}^{K} m_i$ (3.7)

If K=1, W_S is fixed. Thus we assume K>1. Let $f_i = \frac{1}{\lambda}\frac{\partial \lambda_i W_i}{\partial \lambda_i}$ and $s_i = \frac{\partial f_i}{\partial \lambda_i}$ $(i = 1, 2, ...K)$.

According to the corollary (1.5), we have

$$f_i = \frac{1}{\lambda}\frac{\partial \lambda_i W_i}{\partial \lambda_i} = \frac{1}{\lambda}(W_i + \lambda_i \frac{\partial W_i}{\partial \lambda_i}) > 0 \quad (3.8)$$

$$\text{and } s_i = \frac{\partial f_i}{\partial \lambda_i} = \frac{1}{\lambda}(2\frac{\partial W_i}{\partial \lambda_i} + \lambda_i \frac{\partial^2 W_i}{\partial \lambda_i^2}) > 0 \quad (3.9)$$

We let $\frac{\partial W_S}{\partial \lambda_j} = 0$ $(j = 1, 2, ...K-1)$ (3.10)

From (3.4), we treat $\lambda_i (i = 1, 2, ...K-1)$ as $K-1$ independent variables, λ_K depends on $\lambda_i (i = 1, 2, ...K-1)$. Thus $\lambda_K = \lambda - \sum_{i=1}^{K-1} \lambda_i$ (3.11)

and $\frac{\partial \lambda_K}{\partial \lambda_i} = -1$ (for $i < K$) (3.12)

We simplify (3.10) using (3.12),

$$\frac{\partial W_S}{\partial \lambda_j} = \frac{1}{\lambda}\left(\frac{\partial \lambda_K W_K}{\partial \lambda_K}\frac{\partial \lambda_K}{\partial \lambda_j} + \sum_{i\neq j, i=1}^{K-1}\frac{\partial \lambda_i W_i}{\partial \lambda_j}\right) + f_j$$

$$= f_j - f_K = 0 \quad (j = 1, 2, ...K-1) \quad (3.13)$$

We have the following theorem and corollary:

Theorem (3.14) There exists a unique $(\lambda_1, \lambda_2, ...\lambda_K)$ to (3.5), (3.6), (3.7), (3.11) and (3.13) that minimizes W_S. (The proof is omitted due to space)

Corollary (3.15) When $m_1 = m_2 = ... = m_K$, the expected waiting time W_S is minimized iff $\lambda_1 = ... = \lambda_K = \lambda / K$.

Generally, the exact solution of the vector $(\lambda_1, \lambda_2, ...\lambda_K)$ is unattainable since (3.13) are non-linear equations. But an approximate value of the vector $(\lambda_1, \lambda_2, ...\lambda_K)$ can be obtained by using Newton's method for the non-linear equations described in [4], and we will not discuss it here.

3.3.2 The Algorithm

From theorem (3.14), we describe the traffic splitting algorithm as follows:

Algorithm (3.16) Traffic_Splitting()

{

calculate $(\lambda_1, \lambda_2, ...\lambda_K)$ *from (3.5), (3.6),* (3.11) *and* (3.13) *by using Newton's method for the non-linear equations described in* [4].

for each call, do

route the call to M_i *with probability* λ_i / λ

}

Next, we give two examples to calculate the probability to achieve the expected minimum delay:

Example II: $K = 2$, $m_1 = 1$, $m_2 = 2$, $\lambda = 2$, $\mu = 1$.

$\lambda / (\mathrm{m}_1 + m_2)\mu = 2/3 < 1$. From (3.5), we have $W_1 = \frac{\lambda_1}{(1-\lambda_1)}$, $W_2 = \frac{\lambda_2^2}{4-\lambda_2^2}$. From (3.1), we have

$$W_S = \frac{\lambda_1}{\lambda}W_1 + \frac{\lambda_2}{\lambda}W_2 = \frac{\lambda_1^2}{2(1-\lambda_1)} + \frac{\lambda_2^3}{2(4-\lambda_2^2)} \quad (3.17)$$

From (3.11), we have $\lambda_1 = 2 - \lambda_2$ (3.18)

Thus, $W_S = \frac{(2-\lambda_2)^2}{2(\lambda_2 - 1)} + \frac{\lambda_2^3}{2(4-\lambda_2^2)}$ (3.19)

Let $dW_S / d\lambda_2 = 0$, $3\lambda_2^3 - 8\lambda_2^2 + 28\lambda_2 - 32 = 0$ (3.20)

Using Newton's method, we have $\lambda_2 = 1.411,\ \lambda_1 = 0.589,\ p_2 = 0.7055,\ p_1 = 0.2945$, $W_S = 1.124$. Note that although the second media has double the capacity of the first media, *more than* two thirds of the calls are routed to that media in an optimal solution. The reason is as follows: if we route exactly two thirds of the calls to the second media, then $\rho_1 = \rho_2 = 2/3$. From theorem (1.6), we know that the changing rate of the expected queuing delay in the first media is larger than that in the second media, which means that if we route more calls to the second media, we can further decrease the expected queuing delay.

Example III: $K = 2,\ m_1 = 8,\ m_2$=8, $\lambda = 8,\ \mu$=1. Immediately from corollary (3.15), we know that when $\lambda_1 = \lambda_2 = \lambda/2 = 4$, (λ_1, λ_2) minimizes the expected delay for a call. Thus $p_1 = p_2 = 0.5$.

3.4 Performance Comparison: Centralized Queuing vs Distributed Queuing

In this section, we compare the expected queuing delay of the greedy algorithm (for centralized queuing model) and the traffic splitting algorithm (for distributed queuing model), as described in the prior two sections.

Note that the queuing delays that arise in centralized queuing gateway (modeled by an *M/M/m* queue where $m = \sum_{i=1}^{K} B_i$) versus distributed queuing gateway (modeled by K $M/M/B_i$ $(i = 1, 2, ..., K)$ queues) result from slightly different sources. In centralized queuing model, the delay is due to the postponement of routing on the central controller until a channel becomes available on some media. There, no queuing delay results from each media controller. In contrast, in distributed queuing model, there is no delay due to routing on the central controller. Rather, queuing delays occur as a result of waiting to process the call until a channel becomes available on a particular media. As one might expect, we have the following theorem, which establishes that centralized queuing model can never be worse than distributed queuing model. The proof is omitted due to space.

Theorem (3.21) Let K (K>1) be the number of media and let media M_i's bandwidth $B_i = m_i$ and $m = \sum_{i=1}^{K} m_i$. If $\lambda/m\mu < 1$, then $W_G < W_S$.

Examples I and *II* (shown earlier) are examples of applying the greedy and traffic splitting algorithms for the same parameters. It can be easily seen that $W_G < W_S$ in these two examples. The following theorem (the proof is omitted due to space) shows that when each media has the same bandwidth, W_S is at least K times W_G:

Theorem (3.22) Given K (K>1) media, where each media has same bandwidth B, let $m = \sum_{i=1}^{K} B = KB$ and $W_{\min} = \min W_S$. Then if $\lambda/m\mu < 1$, $\frac{W_G}{W_S} < \frac{1}{K}$; if $\lambda/m\mu \to 1$, $W_G/W_{\min} \to 1/K$.

Example IV: $K = 2,\ m_1 = m_2$=2, $\lambda = 3.6,\ \mu$=1.

$$\rho_1 = \frac{\lambda}{(m_1 + m_2)\mu} = \frac{3.6}{4} = 0.9 < 1.\ W_{\min} = 4.263,$$

$W_G = 2.102$. Thus $r_1 = W_G/W_{\min} = 0.493 < 1/2 = 1/K$

Example V: $K = 2,\ m_1 = m_2$=2, $\lambda = 3.96,\ \mu$=1.

$$\rho_2 = \frac{\lambda}{(m_1 + m_2)\mu} = \frac{3.96}{4} = 0.99 < 1,\ W_{\min} = 49.25,$$

$W_G = 24.45$. Thus $r_2 = W_G/W_{\min} = 0.497 < 1/2 = 1/K$

From these two examples, we can see that r_2 is closer to 1/2 than r_1 since ρ_2 is closer to 1 than ρ_1.

Since centralized queuing model can never be worse than distributed queuing model, then what are the advantages of distributed queuing model? By using distributed queuing model, the central controller doesn't need to know the real time channel usage information from each media controller, and the queuing functionality is located inside each media controller. Thus the workload of the central controller can be alleviated.

References

[1] ITU-T *H.323 Packet-Based Multimedia Communications Systems.* Series H: 11/2000.

[2] ITU-T *H.248 Media Gateway Control Protocol*, Series H: A.M.S.., 06/2000.

[3] R. Nelson, *Probability, Stochastic Procedures, and Queuing Theory*. Springer-Verlag, 2000.

[4] L. V.Fausett, *Applied Numerical Analysis Using MetLab*. Prentice Hall, 1999.