Optimal Call Routing in VoIP

Costas Courcoubetis
Department of Computer Science
Athens University
of Economics and Business
47A Evelpidon Str
Athens 11363, GR
Email: courcou@aueb.gr

Costas Kalogiros
Department of Computer Science
Athens University
of Economics and Business
47A Evelpidon Str
Athens 11363, GR
Email: ckalog@aueb.gr

Richard Weber
Department of Pure Mathematics
and Mathematical Statistics
University of Cambridge
Wilberforce Road
Cambridge, CB3 0WB, UK
Email: rrw1@cam.ac.uk

***Abstract*—We evaluate from a VoIP provider's point of view possible strategies for selecting PSTN gateways and/or signaling servers (perhaps through ENUM) under blocking uncertainty. Different gateways may have different prices for terminating the VoIP calls to the PSTN and different blocking probabilities. A customer placing a call to the VoIP provider is impatient and may hang-up if the delay in setting up the call is large. Possible strategies for terminating the call to the PSTN include routing the call to the gateway which generates maximum expected revenue from the call, or simultaneously to a set of gateways charging different prices possibly, a strategy called 'forking'. Forking creates a race between the gateways who are trying to terminate the call and thus reduces the average call setup delay, but at the expense of increasing the average termination cost and the overall load of the system. For the above strategies we investigate the trade-off between the average profit generated by a call and call setup delay. We obtain under several assumptions the optimal set of gateways to which to send a call request. We also discuss the effects of forking on the overall call blocking probability of the system and the incentives for gateways and VoIP providers to deploy it. Our results suggest that if forking is enabled then it can be advantageous for gateways to introduce a small signaling charge.**

I. Introduction

VoIP incorporates all forms of call setup, voice transmission and call management using Internet Protocol (IP) technology. Traditionally, VoIP providers were concerned about the quality characteristics of the voice packets; for example delay and jitter. We argue that call setup phase is more important since call routing choices can affect a) the probability that a call will not be blocked due to lack of resources somewhere on the signaling path, b) the QoS that packets will experience, and c) call setup delay (or Post Dial Delay [8]) which is the time until the caller hears a ringtone. Besides, new networking protocols mitigate traditional quality problems by making it easier to assign higher priority to real-time traffic.

The recent trend for convergence of different access network technologies, market deregulation and emergence of dual-mode end-user devices allow the same destination to be reached by alternative paths and interfaces. Furthermore, new protocols, like SIP (Session Initiation Protocol) and ENUM (tElephone NUmber Mapping), give VoIP providers the ability to devise advanced call routing policies in order to achieve their goals of profit maximization, high quality services, etc. SIP is a very popular signaling protocol, implementing service abstraction and allowing flexible value chains and business models. ENUM is a complementary protocol which describes a) a way to convert traditional telephone numbers into SIP addresses for instance and b) a distributed, DNS-based architecture for answering ENUM queries from compatible end-user devices and signaling servers. ENUM comes in two flavours; a public one where end users opt-in and a closed one that is operated by peering VoIP providers as part of their infrastructure.

For example, in VoIP calls towards a PSTN or mobile destination, a provider's routing policy would specify how to choose from a large set of available telephony gateways. These network devices allow the interoperation of Internet and circuit switched networks for placing and receiving calls. But, a call to the same destination may be performed without the need for a gateway, if only a) the provider has issued an ENUM query and b) the destination can be found in ENUM. The situation is more complicated if we consider the fact that large VoIP providers allow incoming calls only from peers, and that gateways can support only a limited number of simultaneous calls.

Another example of interesting routing policies involves calls between users in circuit switched networks. Gateways have the ability to bypass toll charges (especially for long distance and international calls) and thus are attractive for providers. If the call parties are geographically distant then two gateways may have to be used; one for each call leg. But again, gateway operators and telephony providers must have business relationship for a call to be accepted. Furthermore, each alternative can have different performance in terms of call setup delay and not all users are willing to wait for whatever time it will take.

In this paper for simplicity we refer to the specific case of VoIP calls being terminated to the PSTN, but our results apply to the wider set of cases mentioned earlier. In our model we have *aggregators* and *gateway operators*. A gateway operator is a business entity that owns gateways. An aggregator is an entity whose customers place VoIP calls, acting as an Internet Telephony Service Provider (ITSP) or Inter-eXchange Carrier (IXC). His role is to find a route for an incoming call that will establish the connection and meet customer expectations. This route may involve several gateways or be "on net". The reason

we use that term (instead of ITSP) is to capture the fact that incoming calls to an aggregator can be from retail customers as well as wholesale providers (i.e. other aggregators) [1].

So, in a simple case, end-users become customers of aggregators for making VoIP calls, and aggregators sign contracts with gateway owners for establishing calls to PSTN. Aggregators try to optimize the offerings to customers by maximizing their average profit while keeping call setup times low (a substantial factor in VoIP call quality). Similarly, gateway owners try to maximize the utilisation of their equipment by serving calls that bring revenues. Note that gateway operators must do careful capacity planning since high capacity gateways are costlier and have to rent trunks (i.e. in multiples of T1/E1).

In this paper we evaluate from an aggregator's point of view its possible strategies for selecting gateways and/or signaling servers (perhaps through ENUM) under blocking uncertainty. A customer placing a call to the VoIP provider is impatient and may hang-up if the delay in setting up the call is large. Hence it may not be optimal to route the call to the cheapest gateway if its blocking probability is high; for if the call is blocked then routing to another gateway would increase the call setup delay and the risk that the customer hangs up. Possible strategies for routing calls include routing to the gateway generating maximum expected revenue from the call, or simultaneously to a set of gateways, a strategy called 'forking'.

We note that little research work has been done on aggregators' routing strategies. Most researchers have focused on quality of service metrics, i.e. [6], [9]. Closer to our work is [17], [13], in which simple routing policies are evaluated with the aim of minimizing call blocking probability, and [4] where proposed strategies try to balance network efficiency (call rejection probability and conversation QoS), economic efficiency (prioritize callers with high utility) and aggregator's welfare maximization. Last two papers above are based on periodic information about state of own gateways learned through TGREP (Telephony Gateway REgistration Protocol or TRIP-GW), and thus are restricted to cooperative settings (i.e. a single aggregator that owns all gateways). The reason is that TRIP (Telephony Routing over IP) and other standardized inter-domain call routing protocols do not advertise dynamic gateway state information across VoIP providers, so routing decisions are made under blocking uncertainty.

[13] concludes that more calls are accepted when they are directed proportionally to the number of free circuits of a gateway, rather than all directed to the less utilized one. The authors of [17] perform simulations to evaluate the performance of a system composed of two VoIP providers that route calls on each other's gateways only if their own gateways are known to be blocked. They find that when TGREP message propagation delay increases (i.e. $\geq$ 125ms) gateway blocking probablity can increase significantly, since information becomes stale and calls do not always follow a backup route. Finally, [4] proposes that gateways should perform congestion pricing and an aggregator should choose the single one to use by considering their free capacity and network path characteristics, as well as caller's willingness to pay. However, their schemes rely on truthful report of free voice ports and thus are not directly applicable for competitive markets.

Forking creates a race between the providers who are trying to terminate the call and thus reduces the average call setup delay and the blocking probability; but it is at the expense of increasing the overall load on the system (for dealing with congestion issues in SIP see [16]), and the average termination cost, since an expensive gateway may win the race. For the above strategies we investigate the trade-off between the average profit generated by a call and call setup delay. We obtain under certain assumptions an aggregator's optimal routing strategy in general and dynamic scenarios, since our model is not specific to PSTN gateways. This strategy is of a very interesting form: start with the lowest price gateways, and then, as time passes and these turn out to be blocked, add more expensive gateways to the race to increase the probability that the call setup will be successful before the customer hangs-up. We emphasise that the problem of optimal team effort building that we solve has a wider applicability than to VoIP systems. Somewhat similar types of the problem have been considered in [3], [10] and [15], amongst others. In these papers it is supposed, for example, that a customer wishes to source an item and can ask several providers for it. Providers have independent inventories and so either the customer ends up with surplus items (increasing his cost), or he can choose the best of the items supplied.

Finally, we discuss the effects of forking on the overall call blocking probability of the system and the incentives for gateways and VoIP providers to deploy it. An individual call may profit from forking since that gives it a greater probability of being connected, but it increases the load on the system by reserving a circuit at each gateway during the signaling phase. This increases the blocking probability of other incoming calls. We show, in the context of a specific example, that the optimum amount of forking for the overall system of aggregators and gateways is not a Nash equilibrium. It can be turned into an equilibrium by having the gateways charge the aggregators a signaling cost.

The paper is organized as follows. In Section II we describe our model of aggregators and gateways. In Sections III and IV we analyze the optimal policy for routing to a single gateway, and to multiple gateways, respectively. In Section V we consider forking and its incentive problem, and then end with our conclusions.

II. The model

Let us consider a specific destination prefix of the PSTN and let $A = \{A_1, A_2, \cdots, A_k\}$ denote the set of aggregators and $G = \{G_1, G_2, \cdots, G_n\}$ the set of gateways that receive and terminate calls to the above destination number area.

[1]By acting as an intermediary IXC (at a wholesale level), an aggregator can also solve the complex billing and charging issues that arise in such an environment by offering a one shop solution to VoIP customers. This is actually a basic reason why IXCs carry billions of call minutes each year and allowed VoIP to take off.

PSTN Gateway G_j has finite circuits and for each successful termination of a call charges a termination price p_j (which we take as fixed for simplicity). Different gateways may have different prices due to different geographical placement and interconnection agreements for terminating VoIP calls as well as different blocking probabilities. Thus, G_j earns revenue with a rate equal to p_j times the rate at which it successfully terminates calls. In a broader version of the model, G would also include signaling servers that block calls if their policy suggests "accept incoming calls only from peers".

To model call blocking, we assume that a PSTN gateway will block a call if it has run out of circuits due to excessive demand, and this happens with probability b_j. Furthermore, we assume that core PSTN network is over-provisioned and thus the only bottleneck points for terminating calls of a given user are the gateways. For this to hold we make the assumption that a destination has a negligible probability of being busy each time a call to this destination is made through the VoIP system.

In our model blocking at a gateway can only occur because available circuits are exhausted, not because of protocol message processing overload. An extension of this may include congestion effects due to protocol processing. In this case the SIP proxy that handles signaling (processes the various protocol messages) on behalf of the gateway may add extra delays to the processing of the call by the gateway, or even reject new call setup requests if the above load becomes very large. For simplicity, we assume that a gateway (through its proxy server) can process infinitely fast all protocol messages that it receives.

The time it takes to set up a call is crucial for the revenue model of an aggregator since its customers may be 'impatient'. ITU [1] recommends that average call setup delays for local, national and international calls through PSTN should not exceed 3, 5 and 8 seconds respectively.[2] A business customer for example does not like to wait too long for a call to be setup and may hang up, and if this occurs frequently, he will choose another VoIP provider. We model such a customer to have patience of duration T. He is only charged, an amount p_0, if the call is successfully placed before his patience expires. In this case the aggregator obtains a net revenue of $r_j = p_0 - p_j$ if the call was routed through gateway G_j.

In our discussion so far we have provided the motivation for an aggregator to optimize its strategy in selecting amongst the gateways to which it will route incoming calls so as to maximize its total profit, given the fact that there are delays involved in terminating the call, inexpensive gateways may be more likely to block, and users placing calls may hang up if these delays become large.

[2]Similarly, by adding the delay of each mesage exchanged during a GSM call setup gives us on average 3.58 seconds, while using SIP in (Terrestrial) UMTS may need even more time due to possible retransmissions and channel erros attributed to the wireless medium. [2]. Furthermore, in case of Satellite UMTS an average call may suffer multiple times the setup delay of Terrestrial UMTS [12]. Besides, there is always the case that more than one gateways may have to be used in order to terminate a single call, like in the case of PSTN-IP-PSTN calls.

Here, we try to answer the following scheduling problem:

Given an incoming call from a customer, which set of gateways should an aggregator choose and in what time sequence should the requests be sent to these gateways in order to maximize its expected total reward from the call?

The information available to the aggregator may include for each gateway, in addition to the termination charges and the blocking probabilities, the various delays involved (which may depend on the particular gateway). We also assume that the aggregator knows from historic information the parameter of the distribution of the time T that a customer will wait for the destination ringing signal before hanging up due to impatience.

Instead of answering this question in its generality, we consider specific cases by making the appropriate assumptions. In order to do that we will investigate two broad sets of strategies: (a) trying one gateway at a time, and (b) trying multiple gateways for the same call. The latter is possible with SIP by using a procedure known as *forking*.

III. Trying one gateway at a time

This case occurs if an aggregator hasn't configured his proxies to support forking, and a new gateway will be tried only if the previous one reported being blocked. Let r_i denote an aggregator's reward if a call is placed through gateway G_i, and suppose that gateways are indexed so that average reward decreases as

$$(1 - b_1)r_1 \geq \cdots \geq (1 - b_n)r_n \,. \tag{1}$$

Suppose that time moves in discrete steps $t = 1, 2, \ldots$. This is a simplified case in which all gateways take the same time to report back to the aggregator that the setup of the call is successful or that they are blocked. We wish to maximize the expected revenue obtained by the call given that the customer who placed it will hang up after a deterministic time T. Then the strategy of the aggregator depends on the conditional blocking probabilities. If the probability that gateway i is blocked at time $s > t$ given that it was found blocked at time t is the same as the steady state probability b_i, then one should keep trying gateway 1.

A more sophisticated variation of the previous strategy would be to add some delay between redials. When an aggregator sends a call setup request to the gateway it inspects whether it is blocked or not. Since there are message delays between the aggregator and the gateway, there is a minimum time τ between such consecutive inspections due to the above message delays. Given the fact that each time the gateway finds the gateway blocked it increases the call setup time by at least τ, it may be sensible to delay the next inspection. By doing that the probability the gateway is in the blocked state will decrease and be closer to the steady state probability. Hence by adding some delay between inspections, the average time to find the preferred gateway not blocked may decrease. We show the surprising result that adding such delays is never a good idea, for any value of the minimum delay between inspections τ. This property is proved in a more general context, concerning

the inspection of an arbitrary state of a reversible continuous time Markov chain.

Consider any irreducible continuous-time Markov process on a discrete state space. Suppose we inspect such a system at time 0 and find it to be in some state i. For example, this might be the state in which all servers are busy. We now wish to reinspect at times $t, 2t, 3t, \ldots$, until the first time $T = kt$ that we find the system is not in state i. Between our inspection times the Markov process process carries on as usual. The aim is to minimize $E[T]$, subject to $t \geq \tau$. Might we wish to wait for some time $t > \tau$, so as to increase the probability that the state is not i at the next inspection point? In general, this question is open. We provide a proof for the special case that the Markov process is reversible. The answer is that $t = \tau$ is optimal. For example, we might model the number of free circuits in a gateway as a birth death process. Any birth-death process is reversible, and so one should reinspect in such a system as fast as possible.

Theorem 1 *If a continuous-time Markov process is reversible then it is optimal to reinspect as fast as possible. Thus under the constraint that $t \geq \tau$ we should take $t = \tau$.*

Proof. Let $p(t)$ be the probability that the process is in state i at time t, given that it was in state i at time 0. Then if we reinspect every t units of time we first find it not in state i at time T, where

$$E[T] = t + p(t)E[T] = \frac{t}{1 - p(t)}\,.$$

Consider

$$\frac{d}{dt}\left[\frac{t}{1-p(t)}\right] = \frac{1 - p(t) + tp'(t)}{(1-p(t))^2}\,.$$

Kingman [11] states that for a reversible Markov chain $p(t)$ is a completely monotone function, i.e., representable as

$$p(t) = \int e^{-at} dF(a)$$

for some distribution function F. Theorem 1 follows as

$$1 - p(t) + tp'(t) = \int [\, e^{at} - 1 - at\,] e^{-at} dF(a) \geq 0$$

and $e^{at} - 1 - at \geq 0$ for all at. ∎

IV. Trying multiple gateways simultaneously

Another way to increase the probability of placing a call successfully is to use forking, i.e., try more than one gateway simultaneously. In this case, assuming that each gateway with free circuits is equally likely to be the one that connects the call, the aggregator will obtain the average reward amongst those gateways that he tries and which turn to be unblocked. Note that in the case of forking, by delaying call requests sent to particular gateways the aggregator may control better the expected charge and obtain a higher average reward. This is because the first gateway that receives the call request has a bigger chance to be the winner in the resulting race. Hence it may be beneficial to delay the calls to gateways with lower reward. In this section we analyze the basic forking policy in which calls are sent to all selected gateways simultaneously, while in Section V we show that forking is advantageous to aggregators.

We analyze first a case of equal gateway response times and deterministic T, then a case of different response times and exponentially distributed T. Finally, we provide an optimal control formulation for a more abstract version of the dialling problem, an optimal 'team formation' problem. These results point to the following intuitive policy: an aggregator should first try the most profitable gateways and if blocked, as time becomes more critical augment the team of competing gateways by including less profitable ones.

A. The single period problem

Suppose time is discrete, i.e., gateways take the same time of $\tau = 1$, to report back to the aggregator the outcomes of their call setup attempts. Consider first the case in which the caller hangs up at time $T = 1$, so that there is only one chance to connect the call. Let I_i be distributed as a binomial $B(1, 1-b_i)$ random variable. That is, $I_i = 0$ and $I_i = 1$ with probabilities b_i and $1 - b_i$ respectively. The expected reward obtained from attempting to place the call through a set of gateways S is

$$g\,(S) = E\left[\frac{\sum_{i\in S} I_i r_i}{\sum_{i\in S} I_i}\right], \tag{2}$$

where we adopt the convention that $E\left[\frac{0}{0}\right] = 0$. This expected profit is the average of the sum of all rewards from gateways that were invited and were found non-blocked.

It is desired to choose S to maximize $g(S)$. We shall prove various results about this optimization problem under one or more of the following conditions, the second of which we have met as (1).

$$b_1 \geq \cdots \geq b_n \tag{3}$$

$$(1 - b_1)r_1 \geq \cdots \geq (1 - b_n)r_n \tag{4}$$

$$r_1 \geq \cdots \geq r_n \tag{5}$$

Note that (3)–(4) imply (5). A special case in which all these hold is $(1 - b_1)r_1 = \cdots = (1 - b_n)r_n$ and $r_1 \geq \cdots \geq r_n$. This case can be motivated by the notion that if there are many aggregators trying to place calls then they will tend to send their traffic to a gateway where $(1 - b_i)\, r_i$ is greatest and the effect of this will be to increase the blocking probability b_i of the cheapest gateways. So in equilibrium we might suppose that the $(1 - b_i)\, r_i$ are all equal.

Theorem 2 *Suppose that* (3)–(5) *hold. Then $g(S)$ is maximized by S amongst the collection of sets*

$$L = \{1\}\,, \{1, 2\}\,, \{1, 2, 3\}\,, \ldots, \{1, 2, 3, \ldots, n\}\,. \tag{6}$$

Proof.

Consider a set S and $i, j \notin S$ with $i < j$, Let $c_i = 1 - b_i$, $R = \sum_{k\in S} I_k r_k$, $N = \sum_{k\in S} I_k$, $S_i = S + \{i\}$ and $S_j =$

$S + \{j\}$. Then

$$\begin{aligned} & g(S_i) - g(S_j) \\ &= (b_i - b_j)g(S) + c_i E\left[\frac{R + r_i}{N+1}\right] - c_j E\left[\frac{R + r_j}{N+1}\right] \\ &= (b_i - b_j)\left(g(S) - E\left[\frac{R}{N+1}\right]\right) \\ &\quad + (c_i r_i - c_j r_j) E\left[\frac{1}{N+1}\right]. \end{aligned}$$

The right hand side is nonnegative, by assumptions (3)–(4) and the fact that $g(S) = E[R/N] \geq E[R/(N+1)]$. So it is as good to add i to S as add j to S. This implies that $g(S)$ is maximized by some $S \in L$. ∎

It is interesting to compare our problem to one of Chade and Smith [5], who aim to maximize the expected maximum reward (rather than average reward) of the available items amongst a queried set S, minus a cost that is increasing in the size of S, i.e.,

$$\ell(S) = E\Big[\max_{i \in S}\{I_i r_i\}\Big] - c(|S|)\,.$$

For example, a student might have information on universities' rankings and acceptance rates, and wish to choose a set to which to apply so as to maximize the expected value of the highest ranked university that accepts him, minus a cost of making the applications. Chade and Smith comment that $\ell(S)$ is a submodular set function and that maximizing such a function is NP-hard in general (see [14]). However, $\ell(S)$ can be maximized by a marginal improvement algorithm that runs in polynomial time of $O(n^2)$.

By contrast, if we do not assume that (3)–(5) hold, then $g(S)$ is not maximized by a marginal improvement algorithm, or any other efficient algorithm that we have been able to discover. We suspect the problem is NP-hard. To illustrate the difficulty, consider $\{r_1, r_2, r_3\} = \{0.97, 0.88, 0.87\}$ and $\{b_1, b_2, b_3\} = \{0.36, 0.07, 0.02\}$. Then $\{c_1 r_1, c_2 r_2, c_3 r_3\} = \{0.6208, 0.8184, 0.8526\}$. It turns out that $g(S)$ is maximized by $S = \{1, 3\}$. This is not a set of largest $(1 - b_i)r_i$, or of largest r_i.

We now turn to the problem of identifying which of the sets in L is optimal when (3)–(5) hold. Let

$$g_j = g\left(\{1, 2, 3, \ldots, j\}\right). \tag{7}$$

The greatest g_j is particularly easy to find because g_j is unimodal, increasing up to a maximum and then decreasing in j. In fact, we show in the following theorem that the sequence $\{g_j\}$ is quasiconcave, which means

$$g_j \geq \min\{g_{j-1}, g_{j+1}\}, \tag{8}$$

for all $j \in \{2, \ldots, n-1\}$. This implies that the sequence $\{g_j\}$ is unimodal.

Theorem 3 *Suppose* (5) *holds. Then* $\{g_1, g_2, \ldots, g_n\}$ *is a quasiconcave sequence.*

Proof. If $\{g_j\}$ were *not* quasiconcave, we would need $g_j > g_{j+1}$ and $g_{j+2} > g_{j+1}$ for some $j \geq 1$. Let $S = \{1, \ldots, j\}$, and as previously, let $R = \sum_{k \in S} I_k r_k$ and $N = \sum_{k \in S} I_k$. For any ℓ, m, define

$$K_\ell = \frac{R + r_\ell}{N+1}\,, \quad K_{\ell m} = \frac{R + r_\ell + r_m}{N+2}$$

$$G_\ell = E[K_\ell]\,, \quad G_{\ell m} = E[K_{\ell m}]\,.$$

Then

$$\begin{aligned} & g_j > g_{j+1} \implies g_j > G_{j+1} \\ & g_{j+2} - g_{j+1} > 0 \\ & \quad \implies b_{j+1}(G_{j+2} - g_j) + c_{j+1}(G_{j+1,j+2} - G_{j+1}) > 0 \end{aligned}$$

Combining these we find that we need

$$b_{j+1} G_{j+2} + c_{j+1} G_{j+1,j+2} > G_{j+1}\,.$$

The assumption of (5) that $r_1 \geq \cdots \geq r_n$ means that $K_{j+1} \geq K_{j+2}$ and $K_{j+1} \geq K_{j+1,j+2}$. This means that the above cannot hold, since the left hand side is at most G_{j+1}. Thus the sequence $\{g_j\}$ is quasiconcave. ∎

B. Forking with different gateway response times

Let us consider the continuous time problem in which once an aggregator sends a call request to a gateway, it takes exponential time with parameter λ_j for a response of the gateway to reply that it was successful or that the gateway was found blocked. Again, reward is obtained only if the call is connected within an time T that is exponentially distributed with parameter β, and suppose that I_j is distributed as a binomial $B(1, 1-b_j)$ random variable. In this case the optimal policy is stationary. This is in contrast with the nonstationary policies that are optimal when T is not exponentially distributed, like in Section IV-C below.

If we are only able to ask each gateway once, then the expected return is

$$h(S) = E\left[\frac{\sum_{j \in S} I_j \lambda_j r_j}{\beta + \sum_{j \in S} I_j \lambda_j}\right].$$

If we may retry the gateways, and their blocking probabilities are stationary, then we seek a set S to maximize $f(S)$, where

$$f(S) = \frac{\sum_{j \in S} \lambda_j \left[(1 - b_j)\, r_j + b_j f\,(S)\right]}{\beta + \sum_{j \in S} \lambda_j}\,. \tag{9}$$

Note that due to the stationarity of the solution, a gateway that is found blocked will be invited again, unless the caller has abandoned the call. Solving (9) we obtain

$$f(S) = \frac{\sum_{j \in S} \lambda_j\,(1 - b_j)\, r_j}{\beta + \sum_{j \in S} \lambda_j\,(1 - b_j)} = \frac{\sum_{i \in S} \alpha_i r_i}{\beta + \sum_{i \in S} \alpha_i}\,, \tag{10}$$

where $\alpha_j = \lambda_j\,(1 - b_j)$. We have seen that the maximization of $g(S)$ is a difficult problem unless conditions (3)–(5) hold. However, the maximization of $f(S)$ is easy, as the following theorem shows.

Theorem 4 *If* (5) *holds then the f-maximizing set must be in L.*

Proof. Suppose the theorem is not true and that the uniquely optimal set is $S_j = S + \{j\}$ which includes j but not i, where $i < j$, and thus $r_i \geq r_j$. Let $r(S) = \sum_{k \in S} \alpha_k r_k$ and $\alpha(S) = \beta + \sum_{k \in S} \alpha_k$. Then if S_j is to be uniquely optimal we need

$$f(S_j) > \max\{f(S_{ij}), f(S)\}\,.$$

The right hand side is least when $r_i = r_j$, so we require

$$\frac{r(S) + \alpha_j r_j}{\alpha(S) + \alpha_j} > \max\left\{\frac{r(S) + \alpha_i r_j + \alpha_j r_j}{\alpha(S) + \alpha_i + \alpha_j}, \frac{r(S)}{\alpha(S)}\right\}.$$

A little routine algebra shows that this is impossible. The first inequality above requires $r(S) - r_j\alpha(S) > 0$, whereas the second requires $r(S) - r_j\alpha(S) < 0$. ∎

By a similar argument as in Theorem 3 one can prove a result about the quasiconcavity of f over increasing sets in L.

C. An optimal control problem

We now consider a model that is more abstract, but which generalizes Section IV-B by permitting the time T at which the caller hangs to have a more general distribution than exponential with parameter β. Suppose that the aggregator knows that all the gateways are unblocked and wishes to choose, as a function of time, when to ask each gateway to start attempting the call setup. Once asked, the time that gateway i takes to set up the call is exponentially distributed with parameter μ_i. In order that gateways with greater r_i should have a greater probability of being the one to connect the call, we suppose that the aggregator can signal to each gateway i that at time t he should attempt the connection only at some fraction $u_i(t)$ of his maximum rate μ_i, where $u_i(t) \leq 1$. This may be impractical, but we shall shortly see that under certain conditions the optimal solution is a practical one, in which $u_i(t)$ switches from 0 to 1 at a single time, i.e., the time at which the aggregator asks gateway i to join others in also trying to set up the call.

Let $x(t)$ be the probability that no gateway has connected the call by time t. Then

$$\dot{x}(t) = -\sum_i \mu_i u_i(t) x(t)\,.$$

Consider a problem of maximizing the expected reward obtained by time T (the time at which a customer gives up, and which for now we take to be deterministic).

$$\int_0^T \sum_i \mu_i r_i u_i(t)\, x(t)\, dt\,.$$

This simple control problem can be solved by Pontryagin's Maximum Principle. The Hamiltonian is

$$\begin{aligned} H &= \sum_i \mu_i r_i u_i(t) x(t) - \eta(t) \sum_i \mu_i u_i(t) x(t) \\ &= x(t) \sum_i \mu_i (r_i - \eta(t)) u_i(t)\,. \end{aligned}$$

This is maximized by taking $u_i(t) = 0$ or 1 as $r_i < \eta(t)$ or $> \eta(t)$ respectively. Also, $\eta(T) = 0$ (by a transversality condition and fact that $x(T)$ is unconstrained) and

$$\dot{\eta}(t) = -\partial H/\partial x = -\sum_i (r_i - \eta(t)) u_i(t) \mu_i\,.$$

So $\eta(t)$ is nonnegative and decreasing in t. Assuming once more that $r_1 \geq \cdots \geq r_n$ this implies that the set of gateways that should be attempting to set up call at time t, namely $S(t) = \{i : u_i(t) = 1\}$, is always of the form $\{1, \ldots, j(t)\}$, where $j(t)$ is nondecreasing in t.

Suppose now that T has a distribution with p.d.f. $g(t)$ and c.d.f. $G(t)$. We are seeking to maximize

$$\begin{aligned} &\int_0^\infty \int_0^T \sum_i \mu_i r_i u_i(t)\, x(t)\, dt\, g(T)\, dT \\ &\quad = \int_0^\infty \sum_i \mu_i r_i u_i(t)\, x(t)(1 - G(t))\, dt\,, \end{aligned}$$

where the equality follows from integration by parts. Things are exactly the same as before, but now we have $u_i(t) = 0$ or $u_i(t) = 1$ as $r_i < \xi(t)$ or $r_i > \xi(t)$, respectively, where $\xi(t) = \eta(t)/(1 - G(t))$. The set $S(t)$ is of the form $\{1, \ldots, j(t)\}$, but we no longer have the fact that $j(t)$ is monotonically nondecreasing.

However, $j(t)$ is nondecreasing if the hazard rate of T, namely $h(t) = g(t)/(1 - G(t))$, is nondecreasing (as is most realistic in practice). To see this, note that

$$\begin{aligned} \dot{\eta}(t) &= -\sum_i \mu_i (r_i(1 - G(t)) - \eta(t)) u_i(t)\,. \\ \dot{\xi}(t) &= h(t)\xi(t) - \sum_i \mu_i (r_i - \xi(t)) u_i(t)\,. \end{aligned} \tag{11}$$

In the special case of T that is exponentially distributed with parameter β, one can check that the solution is

$$\eta(t) = \theta e^{-\beta t}, \text{ where } \xi(t) = \theta = \max_S \frac{\sum_{i \in S} \mu_i r_i}{\beta + \sum_{i \in S} \mu_i}\,.$$

More generally we can argue that $\xi(t)$ must be nonincreasing in t. We do this for a slightly different problem. Suppose that for a given T_0 we are seeking to maximize

$$\int_0^{T_0} \sum_i \mu_i r_i u_i(t)\, x(t)(1 - G(t))\, dt\,.$$

This is the expected reward we can obtain by time T_0, where $1 - G(T_0) > 0$. Suppose there is a t for which $\dot{\xi}(t) > 0$. Since $h(t)$ is nondecreasing, it follows from (11) that for small ϵ we must also have $\dot{\xi}(t + \epsilon) > 0$. Thus, if $\xi(t)$ ever reaches a point where it is increasing then it is increasing everywhere following that point. However, this is inconsistent with $\xi(t) > 0$ and $\xi(T_0) = 0$. Having reached the conclusion that $\xi(t)$ is nonincreasing in t, we conclude that $S(t) = \{1, \ldots, j(t)\}$, where $j(t)$ is monotone nondecreasing in t.

V. Is forking desirable?

When aggregators deploy a forking strategy, each gateway that receives a call setup request must reserve a circuit to the PSTN before requesting the PSTN to terminate the call. This circuit cannot be released and used by other calls unless the gateway hears from the PSTN that the destination is busy or the call terminates after a successful establishment. Thus forking imposes a cost on gateways which is not directly charged to aggregators. An individual call setup may benefit by forking, but it creates a negative externality to the rest of the system.

This suggests the question: *Is forking desirable? How do we avoid the inefficient equilibrium resulting from a 'Tragedy of the commons'?* In this section, we will try to give some preliminary answers for a few simple scenarios.

A. The case of two gateways

As in [4] for the case of Probabilistic Selection Based on Utilization, and due to the system complexity (we cannot use Erlang B formula) let us suppose there are just two gateways and one aggregator (or a set of aggregators). Call traffic to the aggregator is Poisson with rate λ. The aggregator randomly selects calls to use forking, resulting in a rate $\lambda_f \leq \lambda$ of forked calls (a call setup is sent to both gateways). Non-forked calls occur with rate $\lambda_{nf} = \lambda - \lambda_f$ and are sent to one of the two gateways at random with probability $1/2$. Calls are processed by the gateways as follows. Each call goes through two phases: first, a signaling phase and second, if signaling is successful, a conversation phase. During each phase one circuit is reserved from the gateway that is involved. We assume that both phases have exponentially distributed durations, with parameters μ_1 and μ_2 respectively. A forked call is not blocked when one of the two gateways has a free circuit. If both gateways have a free circuit then because of the race the service time of the above signaling phase is the minimum of two independent exponential random variables, each with parameter μ_1, and so is exponential with parameter $2\mu_1$. In our model we assume that when the race ends, the gateway who is the winner notifies the aggregator who in turn notifies the other gateway to stop trying to complete the signaling phase.

For this system we show the effect of forking on the blocking probability of calls that fork, on calls that do not fork, and on the average call blocking probability of the system. We observe that as the percentage of forked calls increases, the blocking probabilities of calls that fork and of calls that do not fork both increase. But since the forked calls have a much smaller blocking probability, the overall effect is to decrease the blocking probability of the average call.

In Figure 1 we display results for a system of 2 gateways with $c = 4$ circuits per gateway. It shows the effect on the blocking probabilities of the duration of the signaling phase, as we have solid lines for $\mu_1 = 4$ lying above dotted lines for $\mu_1 = 20$. The calculations have been done by computing the steady-state probabilities of the Markov process. It is represented with a state-space of (x, y_1, y_2, z_1, z_2), where x is the number of forked calls for which setups are being attempted by both gateways, y_i is the number of calls for which setups are being attempted only by gateway i, and z_i is number of conversations in progress through gateway i. There are the constraints $x + y_i + z_i \leq 4$. This gives a 371 state Markov process. We can make the following observations.

(i) The average blocking probability is minimized by maximizing forking to 100%. (But this happens because there are only two gateways; as we shall see shortly, in an example with 6 gateways, the optimal amount of forking may be less than 100%.) A forked call affects the system in two ways. On the one hand it produces signaling work for both gateways. On the other hand, the time that it spends in the setup phase is less in each gateway, due to the race. Forking also gains because forked calls are blocked only if both gateways are blocked, whereas non-forked calls are blocked if one gateway is blocked. E.g., suppose that p_0, p_1, p_2 are the stationary probabilities with which 0, 1 or 2 gateways are blocked. Then forked calls are blocked with probability p_2, whereas unforked calls are blocked with probability $p_2 + \frac{1}{2}p_1$.

(ii) Since forked calls are greedy, they more effectively use circuits from the gateways. Non-forked calls block more because they are less flexible.

These results are supported by those from a discrete event simulator, for different parameter values and more gateways, and which where ommited due to space constraints [7].

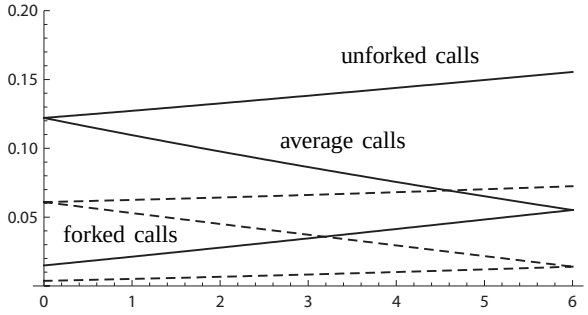

Fig. 1. Blocking probabilities of forked, unforked and average calls as λ_f varies from 0 to 6, with $\lambda_f + \lambda_{nf} = 6$, and $\mu_1 = 4$, $\mu_2 = 2$ (solid lines), and $\mu_1 = 20$, $\mu_2 = 2$ (dashed lines).

B. Incentivizing an optimal amount of forking

Let us consider a simple scenario in which there are 6 gateways, each with just 1 circuit[3]. This system can be represented as a Markov process with 75 states and so it is relatively easy to find out what happens under various forking policies. Suppose calls arrive at rate $\lambda = 1$. If a call setup phase is attempted simultaneously by j gateways then this phase lasts a time that is exponentially distributed with parameter $j\mu_1$. The conversation phase is equally likely to begin in each of these j gateways, and lasts a time that is exponentially distributed with parameter μ_2. Let b_k be the call blocking probability when all arriving calls are forked to k randomly chosen gateways. Note that because some gateways

[3]Such gateways can be home equipment (known as ATA) participating in fwdOUT community (www.fwdout.com) either as callers or callees.

to which a call is forked may be blocked, the setup of a call may actually be attempted by less than k gateways. We find, that with $\lambda = 1$, $\mu_1 = 4$, and $\mu_2 = 2$, the smallest blocking probability is obtained for $k = 4$. It is interesting that the minimum is achieved when all arriving calls are forked to the same number of gateways, rather than, say, some proportion using $k = 3$ and the remainder using $k = 4$.

Suppose that the input traffic is generated by many aggregators. Then both gateways and aggregators are better off when the throughput is maximized. However, there is an obstacle to achieving this. A 'tragedy of the commons' problem arises because an individual aggregator has no incentive to restrict his forking to $k = 4$. The probabilities that his call will be blocked if he forks it to 1, 2, 3, 4, 5, or 6 gateways are 0.1261, 0.0388, 0.0192, 0.0097, 0.0045, and 0.0029, respectively. Thus, he will wish to fork to all 6 gateways. If all aggregators do this then the blocking probability increases from 0.0097 to 0.0418.

One way to incentivize the aggregators to choose a forking parameter less than $k = 6$ is to require an aggregator to pay a charge γ_0 to each unblocked gateway to which he forks a call. So if an aggregator forks his call to k gateways, and j of these are unblocked, then he makes revenue $r - j\gamma_0$ if $j \geq 1$, and 0 if $j = 0$. Suppose all calls are forked to k gateways. Let m_k be the mean number of unblocked gateways that begin attempting a call setup. Assume that all gateways charge the same amount p per call connected, and the aggregator has a profit per call connected of $r = p_0 - p$. Thus his revenue per call that he attempts to place is $R_k = (1-b_k)r - m_k\gamma_0$, where b_k is the blocking probability when all calls are forked to k gateways. For the data of this example:

$$R_1 = 0.8889r - 0.8889\gamma_0 \qquad R_4 = 0.9913r - 3.5043\gamma_0$$
$$R_2 = 0.9776r - 1.7556\gamma_0 \qquad R_5 = 0.9868r - 4.3833\gamma_0$$
$$R_3 = 0.9902r - 2.6287\gamma_0 \qquad R_6 = 0.9582r - 5.2813\gamma_0$$

If we take $\gamma_0 \in [\,0.0005, 0.0011\,]r$ then we induce an optimal amount of forking since $R_4 > \max\{R_1, R_2, R_3, R_5, R_6\}$. Suppose $r = 10$ and $\gamma_0 = 0.007$. The revenue per call is then 9.668, which exceeds the 9.582 that is achieved at the equilibrium of $k = 6$, induced by $\gamma_0 = 0$. The gateways are also better off since with $\gamma_0 = 0.007$, $m_4 = 3.5043$, $b_4 = 0.0087$, and $b_6 = 0.0418$, their revenue is increased by $(1 - b_4)p + m_4\gamma_0 - (1 - b_6)p = 0.0245 + 0.0031p$ per arriving call, relative to what they would have obtained with $\gamma_0 = 0$. One can check that $k = 4$ is the only stable point in the game that results as each aggregator attempts to optimize his forking strategy in response to the forking strategy adopted by others. In the following matrix R_{ij} is the revenue obtained by forking a single call to j gateways when all other calls are being forked to i gateways. The greatest entry in each row is shown in bold. Note that $i, j = (4, 4)$ is the only equilibrium point. Furthermore, when gateways have available circuits they always inform the aggregator that they process the call, so they cannot lie afterwards about their state when they realize that the destination was busy. Gateways lying that are blocked is not beneficial as long as their profit from a successfully terminated call is greater than γ_0.

$$R = \begin{pmatrix} 8.827 & 9.752 & \mathbf{9.800} & 9.750 & 9.689 & 9.627 \\ 8.717 & 9.653 & \mathbf{9.772} & 9.744 & 9.690 & 9.631 \\ 8.701 & 9.576 & 9.718 & \mathbf{9.725} & 9.682 & 9.627 \\ 8.700 & 9.504 & 9.635 & \mathbf{9.668} & 9.659 & 9.615 \\ 8.705 & 9.436 & 9.522 & 9.546 & 9.561 & \mathbf{9.574} \\ 8.741 & 9.378 & \mathbf{9.380} & 9.328 & 9.271 & 9.213 \end{pmatrix}$$

VI. Conclusions

In this work we formulated a model for routing calls in VoIP. We analyzed the optimal server selection strategies of the VoIP providers under blocking uncertainty by considering the trade-off between call setup delay and termination charge when more than a single gateway is invoked to terminate the same call (the case of 'forking'), as well as when forking is not supported by an aggregator. Finally, we analyzed the consequences of forking and showed with an example that a 'tragedy of the commons' problem can arise because individual VoIP providers have the incentive to fork more than is optimal for the system. Our results suggest that if forking is enabled then it can be advantageous for gateways to introduce a small signaling charge.

Acknowledgement

This research project is co-financed by E.U.-European Social Fund (80%) and the Greek Ministry of Development-GSRT (20%).

References

[1] "ITU-T Recommendation I.352," International Telecommunication Union, Tech. Rep., 1993.

[2] "A Comparison Between GERAN Packet-Switched Call Setup Using SIP and GSM Circuit-Switched Call Setup Using RIL3-CC, RIL3-MM, RIL3-RR, and DTAP," Nortel Networks, Tech. Rep., 2000.

[3] L. d. Boer, G. v. Dijkhuizen, and J. Telgen, "A basis for modelling the costs of supplier selection: The economic tender quantity," *The Journal of the Operational Research Society*, vol. 51, no. 10, 2000.

[4] G. D. Caesar, M.C. and R. Katz, "Resource management for ip telephony networks," *IWQoS*, 2002.

[5] H. Chade and L. Smith, "Simultaneous search," *Econometrica*, vol. 74, no. 5, pp. 1293–1307, 09 2006.

[6] R. Cohen and G. Nakibly, "A traffic engineering approach for placement and selection of network services," in *INFOCOM*, 2007, pp. 1793–1801.

[7] C. Courcoubetis, C. Kalogiros, and R. Weber, "Optimal call routing in voip," Tech. Rep., 2009, see http://nes.aueb.gr/users/ckalog.html.

[8] T. Eyers and H. Schulzrinne, "Predicting internet telephony call setup delay," in *In IPTel 2000*.

[9] A. Hari, V. Hilt, and M. Hofmann, "Intelligent media gateway selection in a voip network," *Bell Labs Technical Journal*, pp. 4757–, 2005.

[10] S. P. Ketchpel and H. Garcia-Molina, "Competitive sourcing for internet commerce," in *ICDCS '98*, 1998.

[11] J. Kingman, "Markov transition probabilities. II: Completely monotonic functions." *Z. Wahrscheinlichkeitstheor. Verw. Geb.*, vol. 9, 1967.

[12] V. Y. H. Kueh, R. Tafazolli, and B. G. Evans, "Performance analysis of session initiation protocol based call set-up over satellite-umts network," *Computer Communications*, vol. 28, no. 12, pp. 1416–1427, 2005.

[13] K. Kumaran and A. Sahoo, "Modeling and performance analysis of telephony gateway registration protocol," in *LCN 2007*, pp. 575–582.

[14] L. Lovasz, *Submodular Functions and Convexity*, A. Bachem, M. Grotschel, and B. Korte, Eds. Berlin: Springer, 1982.

[15] O. Madani, "Efficient information gathering on the internet," in *FOCS '96*. Washington, DC, USA: IEEE Computer Society, 1996, p. 234.

[16] M. Ohta, "Overload protection in a SIP signaling network," *International Conference on Internet Surveillance and Protection*, vol. 0, p. 11, 2006.

[17] M. Schlesener and V. Frost, "Performance evaluation of telephony routing over ip (trip)," *IPOM 2003*, pp. 47–53.