Optimal Gateway Selection in VoIP

Richard Weber

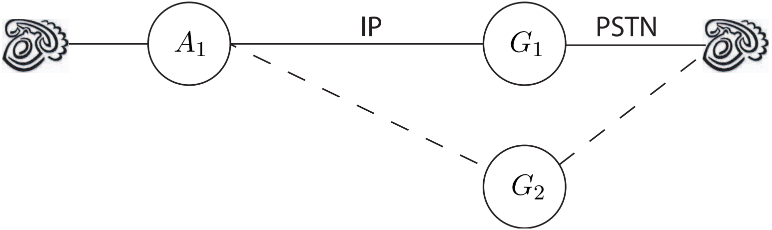
Costas Courcoubetis and Costas Kalogiros

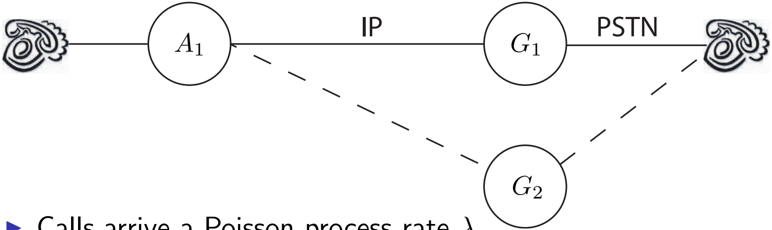
Statistical Laboratory

29 October, 2008

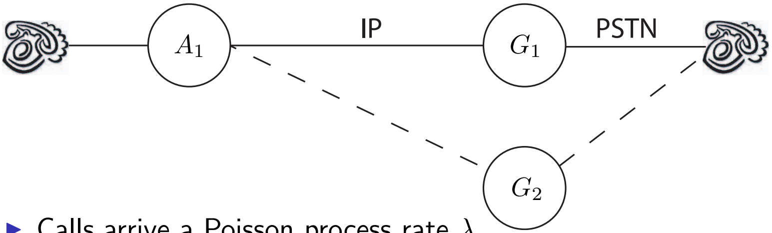
Aggregators and Gateways

Voice over IP is provided by **aggregators**, who terminate calls to the PSTN via **gateways**.

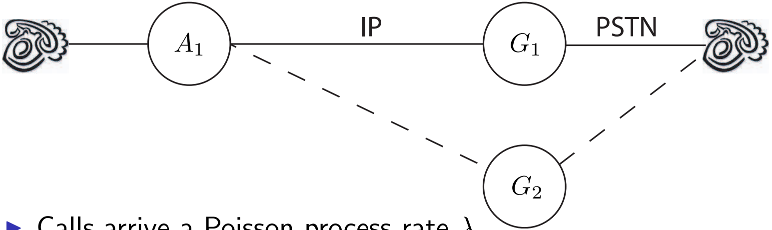




- Calls arrive a Poisson process rate λ.



- Calls arrive a Poisson process rate λ.
- As each call arrives, the aggregator attempts to place it through some subset of the gateways, $S \subseteq \{G_1, \ldots, G_n\}$.



- Calls arrive a Poisson process rate λ.
- As each call arrives, the aggregator attempts to place it through some subset of the gateways, $S \subseteq \{G_1, \ldots, G_n\}$.
- Each gateway $G_i \in S$, reports back (after some delay) whether or not one of its C_i circuits is free.
  - If G_i has a free circuit, then it reserves a circuit and tries to terminate the call at its destination.
    It is in a race with other gateways in S who are also trying to terminate the call.
  - If G_i has no free circuits then it cannot terminate the call.

The Aggregator's Expected Profit

The call is successfully terminated if some gateway $i \in S$ terminates the call before a time T, at which the customer hangs up due to impatience.
Aggregator's reward is

$$r_i = p_0 - p_i \, ,$$

where p_i is the payment he makes to the gateway.

The Aggregator's Expected Profit

The call is successfully terminated if some gateway $i \in S$ terminates the call before a time T, at which the customer hangs up due to impatience.
Aggregator's reward is

$$r_i = p_0 - p_i \,,$$

where p_i is the payment he makes to the gateway.
His expected net profit (assuming the call is terminated before T and all unblocked gateways are equally likely to 'win the race') is

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

where $I_i = 1$ if gateway i has a free-circuit when it is asked by the aggregator to terminate the call. Otherwise $I_i = 0$. $E[0/0] = 0$.

The Aggregator's Problem

Aggregator wishes to maximize expected reward $g(S)$.

Which set of gateways S should the aggregator ask to terminate the call, and in what time sequence should his requests be sent to these gateways?

'**Forking**' is the strategy of asking more than one gateway to terminate the call.

Trying Gateways One at a Time

Suppose we try just one gateway at a time. Depending on assumptions, various orders are optimal. For example, we might suppose

1. Customer gives up after time $T \sim$ exponential(β);
2. Blocking probability of gateway i is b_i.
3. Round trip delay between aggregator and gateway i is τ_i.
4. Time for gateway i to terminate a call to the destination (given it has a free circuit) is σ_i.

Trying Gateways One at a Time

Suppose we try just one gateway at a time. Depending on assumptions, various orders are optimal. For example, we might suppose

1. Customer gives up after time $T \sim$ exponential(β);
2. Blocking probability of gateway i is b_i.
3. Round trip delay between aggregator and gateway i is τ_i.
4. Time for gateway i to terminate a call to the destination (given it has a free circuit) is σ_i.

Then is best to try gateway i before j if

$$r_i \frac{(1-b_i)e^{-\beta(\tau_i+\sigma_i)}}{1-b_i e^{-\beta\tau_i}} \geq r_j \frac{(1-b_j)e^{-\beta(\tau_j+\sigma_j)}}{1-b_j e^{-\beta\tau_j}} .$$

Forking: Optimizing the Forking Set

Suppose we can try more that one gateway at a time (forking).

Suppose $T = 1$ and all gateways take the same time, $\tau_i = 1$, to report back whether or not they are blocked; σ_i are i.i.d., so each gateway is equally likely to 'win the race'. The aggregator has one attempt in which to find a gateway that can terminate the call. He forks to a set of gateways, S, seeking to maximize

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

Forking: Optimizing the Forking Set

Suppose we can try more that one gateway at a time (forking).

Suppose $T = 1$ and all gateways take the same time, $\tau_i = 1$, to report back whether or not they are blocked; σ_i are i.i.d., so each gateway is equally likely to 'win the race'. The aggregator has one attempt in which to find a gateway that can terminate the call. He forks to a set of gateways, S, seeking to maximize

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

or

$$g^\theta(S) = g(S) - \theta b(S)\,,$$

where $b(s) = \prod_{i \in S} b_i$ is the probability no gateway has a free circuit.

$$g(S) = E\left[\frac{\sum_{i \in S} I_i r_i}{\sum_{i \in S} I_i}\right] = \sum_{U \subseteq S, U \neq \emptyset} \frac{1}{|U|} \prod_{i \notin U} b_i \prod_{i \in U} (1 - b_i) \sum_{i \in U} r_i \,.$$

$$g(S) = E\left[\frac{\sum_{i \in S} I_i r_i}{\sum_{i \in S} I_i}\right] = \sum_{U \subseteq S\,, U \neq \emptyset} \frac{1}{|U|} \prod_{i \notin U} b_i \prod_{i \in U} (1 - b_i) \sum_{i \in U} r_i\,.$$

Conjecture. The problem of finding the optimal S is NP-hard.

A Related, but Easier Problem

A student who is applying to universities, at some cost of applying, and can ultimately select the best offer he receives. He wishes to maximize

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

A Related, but Easier Problem

A student who is applying to universities, at some cost of applying, and can ultimately select the best offer he receives. He wishes to maximize

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

This can be solved efficiently by a marginal allocation algorithm:

$$S = \{\}$$

$\texttt{while}\ \max_{i \notin S}\{\,\ell(S + \{i\})\,\} > \ell(S)$

$$S = S + \arg\max_{i \notin S}\{\,\ell(S + \{i\})\,\}$$

$\texttt{endwhile}$

Simplifying Conditions

Let us assume the following.

(a) $b_1 \geq \cdots \geq b_n$.

(b) $(1 - b_1)r_1 \geq \cdots \geq (1 - b_n)r_n$.

(c) $r_1 \geq \cdots \geq r_n$.

Note that (a)–(b) imply (c).

Theorem 1 *Suppose (a)–(c) hold. Then $g^\theta(S)$ is maximized by choosing S amongst the collection of sets*

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

Identifying the Optimal S

Let

$$g_i = g(\{1, 2, \ldots, i\}) \, .$$

Theorem 2 *Suppose (c) holds, i.e.,* $r_1 \geq \cdots \geq r_n$*. Then* $\{g_1, \ldots, g_n\}$ *is a quasiconcave sequence. That is,*

$$g_i \geq \max\{g_{i-1}, g_{i+1}\} \quad \textit{for all } j \in \{2, 3, \ldots, n-1\}.$$

This implies that $\{g_1, \ldots, g_n\}$ is unimodal, and so we can find the optimal S easily.

The Optimal S when Allowed Repeated Attempts

Theorem 3 *Suppose we may make k attempts to place the call. Then, assuming (a)–(c) hold, we should at each successive attempt fork to a set in L of nondecreasing size. Moreover, the expected reward is unimodal over increasing sets in L.*

Let V_k be the maximal expected revenue obtainable in k attempts. The dynamic programming equation is

$$V_k = \max_S \{g(S) + b(S)V_{k-1}\}\,,$$

with $V_0 = 0$. Apply previous results with $\theta = -V_{k-1}$.

Different Gateway Response Times

Suppose it takes a time $\tau_j \sim \text{exponential}(\mu_j)$ for gateway j to reply that it is or is not blocked, and a further time $\sigma_j = 0$ to connect the call. Reward is obtained if the call is connected by time $T \sim \text{exponential}(\beta)$.

If we can only ask each gateway once, the expected reward is

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

If we may retry a gateway when it reports it is blocked, and their blocking probabilities are stationary, then we seek S to maximize

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

Theorem 4 *If (c) holds then the f-maximizing set must be in L, i.e., of the form $\{1,\dots,j\}$ for some j.*

Arbitrarily distributed T

Suppose all gateways are unblocked and T has p.d.f. g.

$$x(t) = P(\text{call not yet terminated by time } t).$$

Consider an optimal control problem of maximizing

$$\int_0^\infty \int_0^T \sum_i \mu_i r_i u_i(t)\, x(t)\, dt\, g(T)\, dT$$

where

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

and $u_i(t)$ is the proportion of its maximum possible effort that we ask gateway i to put into trying to connect the call.

Theorem 5 *If (c) holds, then at time t we should be asking a set of gateways $\{1, 2, \ldots, j(t)\}$ to connect the call. If the hazard rate of T is nondecreasing, then $j(t)$ is nondecreasing.*

The Dialing Problem

Suppose we dial a switchboard and hear,

All our operators are busy, please try again later.

Suppose it takes time τ to redial. We could redial at times $\tau, 2\tau, 3\tau, \ldots$, until we get through. Or we could try at times $t, 2t, 3t, \ldots$, for some $t > \tau$. Suppose we wish to minimize the expected time until we get through, say W.

Should we redial as fast as possible?

The Dialing Problem

Suppose we dial a switchboard and hear,

All our operators are busy, please try again later.

Suppose it takes time τ to redial. We could redial at times $\tau, 2\tau, 3\tau, \ldots$, until we get through. Or we could try at times $t, 2t, 3t, \ldots$, for some $t > \tau$. Suppose we wish to minimize the expected time until we get through, say W.

Should we redial as fast as possible?

$p_{0,0}(t) = P(0 \text{ operators free at time } t \mid 0 \text{ operators free at time } 0)$.

$$W = t + p_{0,0}(t)W = \frac{t}{1 - p_{0,0}(t)}\,.$$

So we should redial as fast as possible if $dW/dt \geq 0$, i.e., if

$$(1 - p_{0,0}(t)) + t\frac{d}{dt}p_{0,0}(t) \geq 0\,.$$

Suppose the switchboard operates as an Erlang loss system with c circuits. In principle, we can solve

$$\begin{aligned}
\frac{d}{dt}p_{0,0}(t) &= \lambda p_{0,1}(t) \\
\frac{d}{dt}p_{0,i}(t) &= (c-i+1)\mu p_{0,i-1}(t) + \lambda p_{0,i+1}(t)\,, \quad 0 < i < c \\
\frac{d}{dt}p_{0,c}(t) &= \mu p_{0,c-1}(t)
\end{aligned}$$

with $p_{0,0}(0) = 1$ and $p_{0,i}(0) = 0$, $i \neq 0$.

More generally, suppose we have a continuous time Markov process which is found to be in state 0 at time 0. We wish to reinspect at times $t, 2t, 3t, \ldots$, and minimize the expected time until we first find it not in state 0, subject to choosing $t \geq \tau$.
In general, it can be optimal to take $t > \tau$. Now

$$p_{0,0}(t) = \sum_k \alpha_k e^{-\nu_i t} \,.$$

More generally, suppose we have a continuous time Markov process which is found to be in state 0 at time 0. We wish to reinspect at times $t, 2t, 3t, \ldots$, and minimize the expected time until we first find it not in state 0, subject to choosing $t \geq \tau$.
In general, it can be optimal to take $t > \tau$. Now

$$p_{0,0}(t) = \sum_k \alpha_k e^{-\nu_i t}\,.$$

Suppose all α_k and ν_k are real, all $\alpha_k > 0$ and $\sum_k \alpha_k = 1$. Then

$$(1 - p_{0,0}(t)) + t\frac{d}{dt}p_{0,0}(t) = \sum_k \alpha_k \left(1 - (1 + \nu_k t)e^{-\nu_k t}\right) \geq 0\,,$$

and so fast dialing is optimal.

Theorem 6 *Suppose a continuous time Markov process is reversible. Then for any state* 0*, we can write*

$$p_{0,0}(t) = \sum_k \alpha_k e^{-\nu_k t},$$

where all α_k *and* ν_k *are real, all* $\alpha_k > 0$ *and* $\sum_k \alpha_k = 1$*.*

Corollary. *Fast dialing is optimal for the Erlang loss model of a switchboard.*
(as this is a reversible Markov process.)

Is Forking Desirable?

An individual call setup may benefit by forking, but it creates a negative externality to the rest of the system because it increases the blocking probability for other call setups.

Is forking desirable? How do we avoid the inefficient equilibrium resulting from this 'Tragedy of the commons'?

A Numerical Example

Consider a case of one aggregator and two gateways.

- Calls arrive Poisson with rate λ.
- A rate λ_f of calls are forked, and $\lambda_{nf} = \lambda - \lambda_f$ are unforked.
- Two phases: (i) a signalling phase ($\sim$ exponential(μ_1)) and (ii), if signalling is successful, a conversation phase ($\sim$ exponential(μ_2)).
- During each phase one circuit is reserved in the gateway involved.
- A forked call is not blocked if at least one of the two gateways has a free circuit. If both gateways have a free circuit then signalling phase is distributed exponential$(2\mu_1)$.
  (*The gateway who is the winner notifies the aggregator who in turn notifies the other gateway to stop trying to complete the signalling phase.*)

A Numerical Example

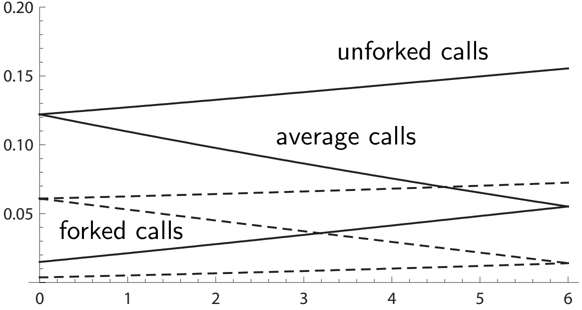


Figure: Blocking probabilities of forked, unforked and average calls as λ_f varies from 0 to 6, with $\lambda_f + \lambda_{nf} = 6$, and $\mu_1 = 4$, $\mu_2 = 2$ (solid lines), and $\mu_1 = 20$, $\mu_2 = 2$ (dashed lines).

Incentivizing an optimal amount of forking

Consider 6 gateways, each with just 1 circuit.
This can be represented as a Markov process with 75 states.

- Calls arrive at rate $\lambda = 1$.
- If a call setup phase is attempted simultaneously by j gateways it lasts time $\sim$ exponential$(j\mu_1)$.
- Conversation phase is equally likely to begin in each of these j gateways, and lasts a time $\sim$ exponential(μ_2).
- $\mu_1 = 4$, $\mu_2 = 2$.

b_k = blocking probability when all arriving calls are forked to k randomly chosen gateways.

Incentivizing an optimal amount of forking

Consider 6 gateways, each with just 1 circuit.
This can be represented as a Markov process with 75 states.

- Calls arrive at rate $\lambda = 1$.
- If a call setup phase is attempted simultaneously by j gateways it lasts time $\sim$ exponential$(j\mu_1)$.
- Conversation phase is equally likely to begin in each of these j gateways, and lasts a time $\sim$ exponential(μ_2).
- $\mu_1 = 4$, $\mu_2 = 2$.

b_k = blocking probability when all arriving calls are forked to k randomly chosen gateways.

b_k is minimized 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 $k = 4$.

A Game of Many Aggregators

Suppose there are many aggregators. Both gateways and aggregators are better off when the throughput is maximized.

However, there is a ‘tragedy of the commons’ because no individual aggregator has no incentive to restrict his forking to $k = 4$.

A Game of Many Aggregators

Suppose there are many aggregators. Both gateways and aggregators are better off when the throughput is maximized.

However, there is a 'tragedy of the commons' because no individual aggregator has no incentive to restrict his forking to $k = 4$.

Suppose we require an aggregator to pay γ_0 to each unblocked gateway to which he forks a call. So if he forks a call to k gateways, and j of these are unblocked, then he has revenue $r - j\gamma_0$ if $j \geq 1$, and 0 if $j = 0$.

Revenue per call 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.

Taking $\gamma_0 \in [\,0.0059, 0.0109\,]r$ then we induce an optimal amount of forking since $R_4 > \max\{R_1, R_2, R_3, R_5, R_6\}$.

Equilibrium of the Game

Let R_{ij} be the revenue obtained by forking a single call to j gateways when all other calls are being forked to to i gateways. The greatest entry in each row is shown in bold.

$$R = \begin{pmatrix} 8.827 & 9.752 & \mathbf{9.800} & 9.750 & 9.689 & 9.627 \\ 8.717 & 9.653 & \mathbf{9.772} & 9.744 & 9.690 & 9.631 \\ 8.701 & 9.576 & 9.718 & \mathbf{9.724} & 9.682 & 9.627 \\ 8.678 & 9.489 & 9.624 & \mathbf{9.658} & 9.649 & 9.604 \\ 8.751 & 9.459 & 9.537 & 9.561 & 9.578 & \mathbf{9.594} \\ 8.743 & 9.380 & \mathbf{9.381} & 9.329 & 9.272 & 9.214 \end{pmatrix}$$

Equilibrium of the Game

Let R_{ij} be the revenue obtained by forking a single call to j gateways when all other calls are being forked to to i gateways. The greatest entry in each row is shown in bold.

$$R = \begin{pmatrix} 8.827 & 9.752 & \mathbf{9.800} & 9.750 & 9.689 & 9.627 \\ 8.717 & 9.653 & \mathbf{9.772} & 9.744 & 9.690 & 9.631 \\ 8.701 & 9.576 & 9.718 & \mathbf{9.724} & 9.682 & 9.627 \\ 8.678 & 9.489 & 9.624 & \mathbf{9.658} & 9.649 & 9.604 \\ 8.751 & 9.459 & 9.537 & 9.561 & 9.578 & \mathbf{9.594} \\ 8.743 & 9.380 & \mathbf{9.381} & 9.329 & 9.272 & 9.214 \end{pmatrix}$$

$k = 4$ is the (unique) Nash equilibrium in the game that results as each aggregator attempts to optimize his forking strategy in response to the forking strategy adopted by others.

Summary

- We have analyzed some optimal gateway selection and forking strategies in simple models.

Summary

- We have analyzed some optimal gateway selection and forking strategies in simple models.
- We have found a solution to the 'dialing problem'.

Summary

- We have analyzed some optimal gateway selection and forking strategies in simple models.
- We have found a solution to the 'dialing problem'.
- We have observed that a 'tragedy of the commons' problem can arise because individual VoIP providers may choose to fork more than is optimal for the system taken as a whole.

  It can be advantageous for both aggregators and gateways if there is the imposition of a small signalling charge.