# A Taxonomy of Rerouting in Circuit-Switched Networks

*Eric W. M. Wong and Andy K. M. Chan, City University of Hong Kong*
*Tak-Shing Peter Yum, The Chinese University of Hong Kong*

**ABSTRACT** In circuit-switched networks it is well known that dynamic routing can provide significant throughput gain over fixed routing. Rerouting is the practice of routing calls currently on alternate paths to direct paths or other less congested alternate paths. Previous studies have shown that rerouting can not only increase the throughput of dynamic routing, but also maintain network stability without the need for trunk reservation. This article presents a taxonomy of rerouting in circuit-switched networks showing the various ways rerouting can be designed. In addition, a comparative study on a number of rerouting schemes are performed in a uniformly loaded, fully connected circuit-switched network.

In recent years, a variety of approaches to alternate path routing networks have been developed in the public switched telephone network (PSTN). AT&T has used a decentralized nonhierarchical routing strategy called *dynamic nonhierarchical routing* (DNHR) for a number of years. DNHR is a time-dependent routing scheme that increases network efficiency by taking advantage of the noncoincidence of busy hours in a large toll network. *Dynamically controlled routing* (DCR), a routing scheme developed by Bell Northern Research, uses a central processor to track the busy-idle status of network trunks and determine the best alternate route choices based on status data collected every 10 s. *Dynamic alternate routing* (DAR) was developed by British Telecom and uses a simple decentralized learning approach to adaptive routing. When the direct trunk group is busy, the two-link alternate path used the previous time is chosen for the overflow call. If the alternate path is busy, the overflow call is blocked, and a new alternate path is selected at random for the next overflow call.

With advances in stored-program control switches and common channel signaling, it is now feasible to monitor trunk occupancies and make routing decision on a call-by-call basis in circuit-switched networks. As a result, many sophisticated dynamic routing schemes (e.g., least-loaded routing, LLR) can therefore be implemented. As an example, AT&T recently adopted a new routing scheme called *real-time network routing* (RTNR). In RTNR, if a direct path is blocked, the call will be routed to the least loaded two-link alternate path. A comprehensive review of dynamic routing can be found in [1, 2].

The major focus of traditional dynamic/alternate-path routing research is to determine the "right" choice of alternate paths for overflow calls. The idea is to increase the network throughput by balancing the traffic load among the alternate paths. Over the years, researchers have learned that simple LLR with trunk reservation can provide significant throughput gain over fixed routing, while other more elaborate approaches (e.g., Markov decision process) can only provide marginal additional gain.

*This project is supported by RGC Comeptitive Earmarked Research Grant under project number 9040427.*

*Editorial liaison: M. Pioro*

In dynamic routing, a routing decision must be made at call arrival based on the network information available at that time. Once a sequence of alternate paths has been chosen, it is final. One method to increase the throughput of traditional dynamic routing is to redistribute network load to eliminate traffic hot spots or bottlenecks. Rerouting (or call repacking) is the practice whereby calls on alternate paths can be rerouted to direct paths or other less congested alternate paths as the situation warrants. Previous studies on rerouting for networks of different kinds can be found in [3–12]. In [5, 9], rerouting is studied in virtual-circuit packet-switched networks and circuit-switched wavelength-division multiplexed all-optical networks, respectively. In [7] an "intentional" rerouting, which is not "forcibly" triggered by a failure, was considered in asynchronous transfer mode (ATM) networks. The article shows that rerouting can be implemented *smoothly* in ATM networks. This means that rerouting can ensure not only first-in first-out (FIFO) service, but also the integrity of cells. Thus, it can be completely transparent to the upper layers of the communicating processes. In [8] a dynamic virtual path rearrangement scheme was proposed, and a strategy based on the scheme was presented for ATM network provisioning. The robustness of this scheme was demonstrated by simulations not only against the forecast errors, but also against changes in transport network and virtual path network structures. Note that as far as fixed route and dedicated bandwidth to users are concerned, a *virtual path* in ATM networks is very similar to a *circuit* in circuit-switched networks, so the results discussed here apply to a large degree to these networks.

In advanced cellular networks, rerouting is realized in the handoff process [13]. In other words, during a handoff a call is rerouted from one base station/mobile switching center to another. In an overlaying cellular network [10, 12] with a mix of micro- and macrocells, a call in a macrocell which is overflowed from a microcell is rerouted back to the microcell whenever possible.

Research on rerouting in circuit-switched networks can be found in [4, 11]. A simple rerouting scheme was studied in [4] through simulation, while in [11] another simple rerouting scheme was analyzed, and numerical examples confirmed the significant throughput gain over the traditional dynamic routing scheme. Moreover, numerical results through analysis and simulation [11] showed that rerouting is another effective means of maintaining the stability of the network besides trunk reservation. It should be noted that besides throughput increase, rerouting can help reduce the effect of link and node failures.

The article is organized as follows. The next section gives a general picture of routing in PSTNs using stored program control switching, common channel signaling, and decentralized control to support sophisticated routing schemes. Some implementation issues are discussed. We then discuss the instability problem and its solutions in dynamic routing. The article presents a taxonomy of rerouting in circuit-switched networks, showing the various ways rerouting can be designed. Next, we discuss a case study from which some interesting observations on rerouting are obtained. Finally, the conclusion is given.

## ROUTING IN THE TELEPHONE NETWORK

A national telephone network is typically structured in a three-level hierarchy with levels representing subscriber instruments, central offices, and toll switching offices [13] (Fig. 1). The set of toll switches form the network core which is usually structured as a fully connected mesh; that is, every toll switch has a logical one-hop path to every other toll switch. A typical routing algorithm is as follows:

- If a call's source and destination are within a central office, directly connect them.
- If the call is between central offices within a local exchange carrier, use a one-hop path between central offices.
- Otherwise, send the call to the core.

The only major decision is at an originating toll switch, which chooses either a direct path or a two-hop alternate path to the destination toll switch. The essence of telephone network routing is to determine which one-hop path to choose in the core, and, if this is fully occupied, the order for trying the two-hop paths. How to choose such an alternate call is essential to the performance of routing in circuit-switched networks. In order to improve the routing performance as well as to provide better services to customers, switching and signaling systems have become more and more sophisticated over the years.

Stored program control (SPC) switches can collect local state information throughout the network, process the information, and make appropriate routing decisions. In a network of SPC switches, interswitch signals are generated by the processor in one switch and acted on by the processor in the other [14]. The signals are transferred over a dedicated high-speed data link between the two processors instead of being transmitted over the telephone circuit. This is called *common channel signaling* (CCS). The current standard is the International Telecommunication Union — Telecommunication Standardization Sector's (ITU-T's) Signaling System No. 7 [14].

The CCS network is a dedicated packet-switched network for exchanging signaling information between SPC switches. Since instantaneous link occupancy and switch congestion level information can be gathered at the switches and sent to other switches, sophisticated routing decisions can be made for each telephone connection.

In addition, the reduction in the cost of processing brought about by the microprocessor has enabled the control of switching systems to be decentralized. Instead of all processing being performed by a central processor, routine tasks associated with parts of the system (e.g., the route switch, concentrators) or particular functions (e.g., line scanning, digit reception, signaling) are delegated to separate processors. Decentralized control can speed up information processing time and enables the real-time computation and execution of sophisticated routing decisions.

The latest in the series of telephone routing algorithms is RTNR, which takes advantage of SPC switches, CCS, and decentralized control. In RTNR, each toll switch monitors the loading of every outgoing trunk and computes a list of lightly loaded trunks. When the direct route is full, the originating switch queries the terminating switch through the CCS network for the busy-idle status of all the links connected to the terminating switch. The originating switch then compares its own link busy-idle status information to that received from the terminating switch and determines the least loaded alternative route. RTNR was shown to be very effective, as reported in [13].

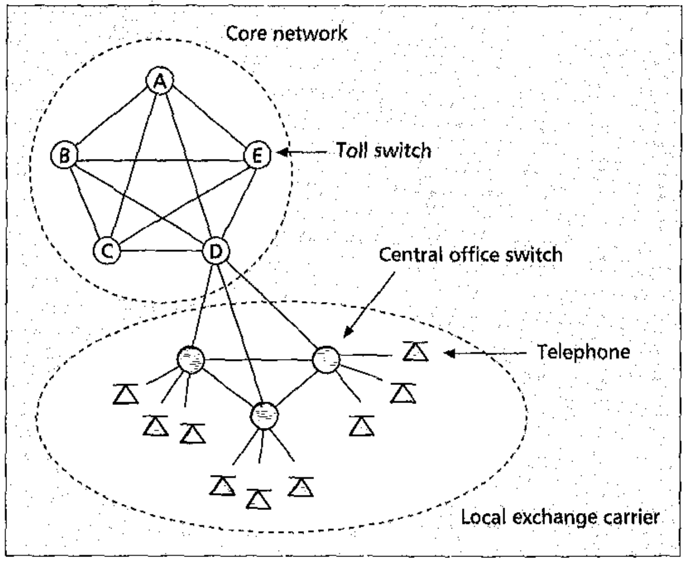



**Figure 1.** *A simplified view of a telephone network.*

Using CCS, signals can be sent while a call is in progress. This enables customers to alter connections after setup (e.g., to transfer a call elsewhere) [14]. It also allows network management to "hand off" calls to other routes for better distribution of network resources. This is called *rerouting*.

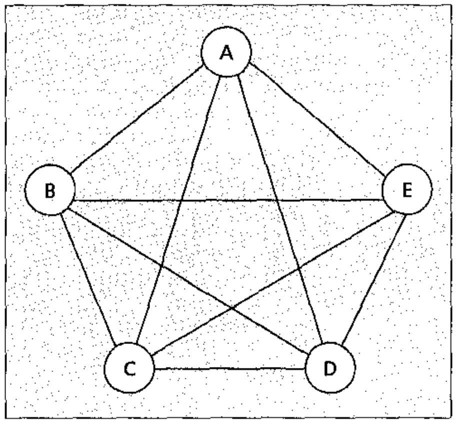
Rerouting involves the finding of an alternate path call and the execution of rerouting it back to its direct path. This is feasible with the current network technology. For example, when a call departs from a toll trunk on link AB, the originating toll switch, say A, can look up its memory to pick up one alternate call designated on toll switch B, say ACB, and route this call to its direct path, AB. Rerouting the call ABC can be facilitated by establishing a new connection on link AB, switching the call ABC to the new connection, and releasing the old connection on path ACB.

## THE STABILITY OF DYNAMIC ROUTING

Dynamic/alternate path routing has been shown to be unstable through analysis and simulation models [15]. In the analytical model, instability means that there are two solutions for a range of offered loads. One solution represents the case that most of the traffic is directly routed, and the other occurs when much of the traffic is alternatively rerouted. An alternately routed call uses two trunks, one on each link of the alternate path, while a direct path call uses only one. For small carried load this presents no problem, and in fact dynamic routing results in a lower blocking probability than fixed routing. For higher loads, the alternately routed calls take up double the resources, block direct path calls in both links, and cause them to use alternate paths as well. This creates an avalanche of alternate path calls and reduces the network throughput to a very low level.

One method of solving this instability problem is to reserve a certain number of trunks for directly routed calls only. The main purpose of trunk reservation is to limit excessive alternate routing during periods of general overload [2].

Rerouting allows calls on alternate paths be rerouted back to their direct paths whenever available; it therefore has the self-stabilizing property. One simple form of rerouting has been analyzed and shown to be stable through case and simulation studies [11]. Later, we will show that a particular form of rerouting can be argued intuitively to be stable. After that, other more elaborate rerouting schemes are shown to be stable by simulation.



■ **Figure 2.** *A five-node fully connected network.*

## A Taxonomy of Rerouting

Since the critical routing decision is on the core part of the telephone network, we focus our study on the most frequently found fully connected core network with alternate paths of only two hops. The rerouting algorithms presented, however, are perfectly general for any kind of mesh network. Figure 2 shows a five-node fully-connected network. Connection requests, or calls, are established by setting up a circuit from the source node/switch to the destination node/switch. This network provides each node pair with a one-hop path (the direct path) and many two-hop paths (the alternate paths).

A call between node pair AB is called a *direct call* if it is carried on link AB. If a call is overflowed from another node pair, say AC, to an alternate path passing through link AB (say path ABC), it is called an *alternate path call on link AB*. That call is also called an *overflow call of node pair AC*. Some other notations are defined as follows:

- $\mathcal{O}_{AB}$ is the set of overflow calls of node pair AB.
- $P_{\mathcal{O}_{AB}}$ is the set of alternate paths carrying calls in $\mathcal{O}_{AB}$.
- $\mathcal{A}_{AB}$ is the set of alternate path calls on link AB.
- $P_{\mathcal{A}_{AB}}$ is the set of alternate paths carrying calls in $\mathcal{A}_{AB}$.
- $N_{\mathcal{A}_{AB}}$ is the set of source-destination node pairs of alternate path calls in $\mathcal{A}_{AB}$.
- $\mathcal{C}_{ABC}$ is the set of alternate path calls in alternate path ABC.

To illustrate, consider link AB in Fig. 3. The two alternate paths as illustrated in this figure are ACB and ADB. The set of alternate path calls are $\mathcal{C}_{ACB}$ and $\mathcal{C}_{ADB}$, respectively. Adding these two sets, we have the set of overflow calls of node pair AB, that is, $\mathcal{O}_{AB} = \mathcal{C}_{ACB} \cup \mathcal{C}_{ADB}$, and hence $P_{\mathcal{O}_{AB}} = \{ACB, ADB\}$. Similarly, the set of alternate calls on link AB consists of those overflowed calls from node pairs AC, AD, BC, and BD, that is, $N_{\mathcal{A}_{AB}} = \{AC, AD, BC, BD\}$. In other words, $\mathcal{A}_{AB} = \mathcal{C}_{ABC} \cup \mathcal{C}_{ABD} \cup \mathcal{C}_{BAC} \cup \mathcal{C}_{BAD}$, and hence $P_{\mathcal{A}_{AB}} = \{ABC, ABD, BAC, BAD\}$.

In the following, we present a taxonomy of rerouting in circuit-switched networks. Rerouting schemes can differ from each other in the following six ways: alternate path selection method, triggering policy, rerouting path selection method, rerouting call selection method, levels of rerouting, and orders of rerouting.

### The Alternate Path Selection Method

The alternate path selection method decides which alternate path to choose when a call has to overflow. We compare two such methods in this article, namely random alternate routing and least-loaded alternate routing. In random alternate routing, if the direct link, say link AB, is full, a two-hop admissible path is selected randomly. Note that a path is called admissible if each link on that path has at least one free trunk. In least-loaded alternate routing, if the direct link is full, a two-hop admissible path with the maximum number of free circuits will be chosen. If there are more than one path having the same maximum number of free trunks on the most congested link, we will select one at random.

To illustrate the two routing rules, take node pair AB in Fig. 3 as an example. Suppose its two alternate paths, ACB and ADB, have {1, 2} and {2, 3} free trunks, respectively. The numbers of free circuits on these two paths are therefore 1 and 2, respectively. In random alternate routing, when the direct link is full, a path is selected randomly from these two alternate paths. In least-loaded alternate routing, the alternate path ADB having the largest number of free circuits is chosen.

### The Triggering Policy

The triggering policy determines under what conditions rerouting is executed. It can be triggered at call arrival time, call departure time, both, or any time the rerouting processor is available. In this article we focus on comparisons of the arrival, departure, and arrival-plus-departure (mixed) triggering policies.

In the arrival triggering policy, a rerouting process is initiated when a connection request for a direct call is blocked, say on node pair AB. The rerouting process will try to select an alternate path call in $\mathcal{A}_{AB}$ and, for example, reroute it back to its direct link if a free trunk is available there. If rerouting is not successful, an alternate path selection procedure is then triggered. With the use of this policy, rerouting should be performed before alternate path routing, if needed.

In the departure triggering policy, a rerouting process is initiated when a connection is released from a path, say between node pair AB. An overflow call will be selected in $\mathcal{O}_{AB}$ and rerouted back to the direct link of node pair AB. In the mixed triggering policy, a rerouting process is initiated both when a connection request for a direct call is blocked and when a connection is released from a path.

To illustrate, consider Fig. 3. With arrival triggering, when a call to link AB is blocked, one of the calls in $\mathcal{A}_{AB}$ will be selected and rerouted back to its respective direct link (i.e., link AD, AC, BC, or BD). With departure triggering, a call departure on node pair AB will trigger a rerouting of a call in $\mathcal{O}_{AB}$ (e.g., from path ACB or ADB) back to link AB.

### The Rerouting Path Selection Method

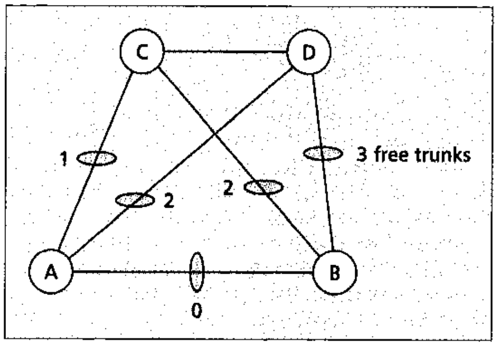
When a call is to be rerouted, it can be sent back to its direct path, to another alternate path, or first to direct paths and, if full, to another alternate path. In this article we focus on the first and third methods for arrival triggering.

### The Rerouting Call Selection Method

The rerouting call selection method determines the particular call to reroute. We study two such methods here: random selection and busiest-first selection.

In random selection, an overflow call is selected randomly from the set of available calls. Take Fig. 3 as an example. In arrival triggering, when a call to link AB is blocked, a node pair in $N_{\mathcal{A}_{AB}}$ (either AC, AD, BC, or BD) is selected at random, and a corresponding alternate path call, if existing, in $\mathcal{A}_{AB}$ is rerouted to its direct link. In departure triggering, when a call departs on node pair AB, a path in $P_{\mathcal{O}_{AB}}$ (either ACB or ADB) is selected randomly, and the corresponding overflow call (in either $C_{ACB}$ or $C_{ADB}$) is rerouted back to its direct link AB.

The busiest-first selection method is similar except that in arrival triggering, an alternate path call is selected (randomly) from a path so that the minimum number of free trunks on links associated with $N_{A_{AB}}$ and $P_{O_{AB}}$ is maximized. If there are more than one candidate path, one will be selected randomly. Alternatively, in departure triggering, when a call is terminated on node pair AB, the busiest alternate path (i.e., with the least number of free circuits) from $P_{O_{AB}}$ is selected and an overflow call (of node pair AB) from that path is rerouted back to direct link AB. If there are more than one candidate call, one will be selected randomly.



■ **Figure 3.** *Direct and alternate paths.*

Taking Fig. 3 again as an example, we assume $P_{A_{AB}}$ = {ABC, BAD} and $P_{O_{AB}}$ = {ACB, ADB}. In other words, we have $N_{A_{AB}}$= {AC, BD}. With arrival triggering, when a call is blocked on direct link AB, it is easy to observe that if an alternate call in BAD is chosen to reroute back to BD, the number of free trunks on links associated with $N_{A_{AB}}$ and $P_{A_{AB}}$ is 1 (counted from link AC which is chosen from links AB, AC, AD, BC, and BD) which is larger than if a call in ABC is rerouted, which is 0 (counted from link AC). With departure triggering rerouting, when a call departs in link AB, its overflow call on path ACB gets rerouted to link AB since the busiest alternate path in $P_{O_{AB}}$ is ACB.

### LEVELS OF REROUTING

Since an alternate path call lies on an alternate path with two links, the rerouting of an alternate path call will free up one trunk in each of these two links. Further rerouting can therefore be performed in these two links.

In this article we study such a scheme for departure triggered rerouting called departure triggered chain rerouting, where the rerouting process is propagated down to as many levels as possible. For example, in Fig. 3 the rerouting of a call on path ACB to link AB will free up one trunk each in links AC and BC. Further rerouting can then be performed on these two links, say a call on path ABC to link AC, a call on path BAC to link BC, and so on.

Departure triggered chain rerouting is a form of "ultimate" rerouting in the sense that we try to reroute back as many alternate calls as possible to their direct links. It is tempting to conclude that departure triggered chain rerouting gives minimum call blocking. However, it will be shown later that this is not true.

Chain-triggering rerouting is inherently stable. To see that, imagine a call from A to C overflowing from link AC to path ABC. This occurs only when links AB and BC both have unoccupied trunks. These two links shall have spare trunks only if there are currently no overflow calls originating from links AB and BC (because otherwise these overflow calls would have been rerouted back to links AB and BC by the chain-triggering action as soon as spare trunks are available). In other words, the overflow call is occupying only the unwanted bandwidth on path ABC. Even if any overflow calls of links AB and BC are accepted after the setup of that call, the network will force those overflow calls back to their direct paths after the completion of that call. Therefore, it is a tendency for the network to always move toward the "high-throughput low-blocking" operating point [15]. To enter the "low-throughput high-blocking" operating point, the network needs to have a tendency to move more and more calls to alternate paths. This tendency is obviously missing in chain-triggering rerouting.

### ORDERS OF REROUTING

With the use of the arrival triggering policy, when a new call finds its direct link (say AC) full, an alternate call on link AC will be identified for rerouting back to its direct link. If all direct links (e.g., AB) of the alternate calls on link AC are full, rerouting fails. However, it is possible to trigger the rerouting of an alternate path call[1] (say on link AB) to make room for the original rerouted call, and hence make room for the new call. For example, in Fig. 3 if a call on path ACB cannot be rerouted to link AB (assumed full), it is possible to reroute a call on link AB (say a call on path ABD to link AD) to free up a trunk for the rerouting of a call on path ACB. This practice is called *second-order rerouting*. Higher-order rerouting can be performed similarly.

### PERFORMANCE COMPARISONS

In a national circuit-switched network which is well dimensioned, the direct path of a node pair is always the preferred path to carry a call because it consumes the least amount of bandwidth resources as well as switching, processing, and signaling effort. The use of an alternate path is more "expensive" in the sense that it consumes two trunks to carry a call and requires more switching effort. The amount of switching effort is proportional to the alternate path traffic rate (i.e., the rate of overflow calls carried by the network). The selection of alternate paths also requires additional processing and signaling effort of the network. This additional effort is proportional to the alternate path attempt rate (or offered load). We take these rates to be the measures of switching and signaling efforts. Similar measures for rerouting are the reroute traffic rate (i.e., the rate of calls successfully rerouted) and the reroute attempt rate.

Our performance study is based on computer simulation of a five-node and a seven-node fully connected networks with trunk group size $N$ = 50. The performance comparisons presented in this section are based on the above four measures on a link (in units of erlangs) plus the call blocking probability. We let $D$ denote the traffic load in erlangs between any two node pairs and $m$ be the number of alternate paths for a particular direct link. For a $K$-node fully connected network, $m = K - 2$. The simulation time is adjusted so that all simulation points have 95 percent confidence intervals within 1 percent of the mean.

As can be imagined, many rerouting rules can be defined. We attempt only to identify a few interesting ones and compare their performance. Table 1 shows the five performance measures of random alternate routing (RAR), least-loaded alternate routing (LAR), random alternate routing with random rerouting (RRR), and LLR with busiest-first rerouting (LBR) in a five-node network. Both rerouting schemes use arrival triggering mechanism. Trunk reservation parameters $r$ = 2 and $r$ = 0 (i.e., without trunk reservation) are assumed, and $D$ = 45.5 is chosen. The latter represents a network utilization factor of $D/N$ = 0.91. For $r$ = 2, it is seen that both rerouting schemes give lower call blocking probabilities than the traditional alternate routing schemes. With rerouting, the alternate path traffic and alternate path attempt rates are both lower than without rerouting. However, this gain is offset

[1] *Note that we could also trigger the rerouting of a direct path call, say AB, to an alternate path after rerouting failure.*

| | Blocking | Alt. path traffic | Reroute traffic | Alt. path attempts | Reroute attempts |
|---|---|---|---|---|---|
| | | | a) $r = 2$ | | |
| RAR | 5.88% | 2.60 | 0 | 5.27 | 0 |
| LAR | 5.74% | 2.51 | 0 | 5.12 | 0 |
| RRR | 3.09% | 2.32 | 1.42 | 3.73 | 5.14 |
| LBR | 3.07% | 2.29 | 1.35 | 3.68 | 5.03 |
| | | | b) $r = 0$ | | |
| RAR | 15.8% | 7.67 | 0 | 14.86 | 0 |
| LAR | 14.7% | 7.09 | 0 | 13.78 | 0 |
| RRR | 2.51% | 4.05 | 2.82 | 5.19 | 8.01 |
| LBR | 2.47% | 3.79 | 2.54 | 4.91 | 7.46 |

■ **Table 1.** *A comparison of different routing schemes with 91 percent loading in a five-node network.*

| | a) Five-node network | | | b) Seven-node network | | |
|---|---|---|---|---|---|---|
| *D/N* | 0.85 | 0.91 | 0.97 | 0.85 | 0.91 | 0.97 |
| (i) Arrival triggering | 0.57% | 2.5% | 5.9% | 0.15% | 1.6% | 5.0% |
| (ii) Second-order arrival | 0.55% | 2.5% | 5.8% | 0.15% | 1.6% | 4.9% |
| (iii) Alternate-to-alternate | 0.54% | 2.4% | 5.7% | 0.14% | 1.5% | 4.7% |
| (iv) Departure triggering | 0.63% | 2.6% | 6.0% | 0.22% | 1.8% | 5.2% |
| (v) Chain departure | 0.66% | 2.7% | 6.1% | 0.23% | 1.9% | 5.3% |
| (vi) Arrival + departure | 0.65% | 2.7% | 6.1% | 0.22% | 1.8% | 5.3% |

■ **Table 2.** *The blocking of various triggering schemes for RRR under different loading,* D/N.

by the addition of the reroute traffic and reroute attempt rates. Comparing the two rerouting rules, we find that all five performance measures are very close to each other. For $r = 0$ (i.e., without trunk reservation), the performance of traditional alternate routing becomes much worse in all five measures. It is obviously due to the instability problem [15]. However, for $r = 0$, both rerouting schemes give lower blocking probabilities when compared to $r = 2$ at the expense of increases in the other four measures. This means that rerouting can stabilize the network just like trunk reservation, but with lower blocking probability. This is because trunk reservation limits the number of alternate path calls on a link, and hence induces a bit of unnecessary blocking. On the other hand, rerouting allows these alternate path calls without causing problems since these calls could be moved back to the direct link if necessary. However, in the use of rerouting, trunk reservation can be used to reduce the amount of switching, processing, and signaling load offered to the network.

Tables 2 and 3, respectively, show the blocking probabilities and the other four performance measures of RRR with:
- Arrival triggering
- Second-order arrival triggering
- Arrival triggering with rerouting alternate path calls to other alternate paths (if rerouting to the direct path is not successful)
- Departure triggering
- Departure triggered chain rerouting
- Mixed (arrival-plus-departure) triggering schemes

It is found that the arrival, second-order arrival, and alternate-to-alternate triggering schemes outperform the departure triggering and chain departure schemes in all measures. This is because departure triggering schemes often make unnecessary reroutings and sometimes, due to the wrong anticipation of congestion, wrong rerouting decisions. The Appendix gives an example to illustrate the reason behind this. For the arrival triggering schemes, it is seen that with the additional options of second-order rerouting and rerouting alternate path calls to other alternate paths, the call blocking probabilities can be slightly reduced at the expense of additional switching and signaling efforts. For the departure triggering schemes, it is interesting to observe that departure triggered chain rerouting has higher call blocking probability than simple departure triggering. This is because chain triggering creates more congestion around the direct links and causes higher blocking. The Appendix gives an example to elaborate this point. Moreover, we find that the mixed triggering scheme has larger call blocking probability when compared with the arrival or departure triggering schemes, although the mixed scheme initiates more reroute attempts than the sum of those of the individual ones.

Table 4 shows the rate (in erlangs) of direct and alternate calls per link of five rerouting schemes for a five-node network under various traffic conditions. We see that the average numbers of direct calls (per link) for all schemes are virtually the same, while the average numbers of alternate calls are quite different.

Table 5 compares the various triggering schemes with an idealized rerouting scheme in a four-node network with trunk group size $N = 20$. The idealized rerouting scheme works as follows:
- Rerouting of alternate calls to direct paths is always performed whenever possible.
- Rerouting of an alternate call to other alternate paths is allowed.
- Only two-hop alternate paths are allowed. As a result, rerouting of direct calls to alternate paths to make room for a new call is not allowed because the new call kicks away a

| | Alt. path traffic | Reroute traffic | Alt. path attempts | Reroute attempts |
|---|---|---|---|---|
| (i) Arrival triggering | 4.05 | 2.82 | 5.19 | 8.01 |
| (ii) Second-order arrival | 4.11 | 2.92 | 5.24 | 13.09 |
| (iii) Alternate to alternate | 4.60 | 3.49 | 9.17 | 8.08 |
| (iv) Departure triggering | 10.00 | 9.36 | 11.23 | 44.92 |
| (v) Chain departure | 10.19 | 9.57 | 11.41 | 31.69 |
| (vi) Arrival + departure | 9.96 | 9.33 | 11.19 | 56.18 |

■ **Table 3.** *The routing effort of various triggering schemes for RRR with 91 percent loading in a five-node network.*

direct call to use an alternate path and is itself using a two-link alternate path. In other words, doing so will use up three trunks to accommodate a new call.
- All levels and all orders of rerouting are allowed.

Because this scheme possesses all the flexibilities, its blocking performance can serve as a lower bound for the various triggering schemes and allows us to assess how much additional blocking improvement is possible by making rerouting schemes more and more complex.

|  | D/N = 0.85 |  | D/N = 0.91 |  | D/N = 0.97 |  |
|---|---|---|---|---|---|---|
|  | Direct | Alternate | Direct | Alternate | Direct | Alternate |
| (i) Arrival triggering | 41.37 | 1.85 | 43.16 | 2.45 | 44.31 | 2.69 |
| (ii) Second-order arrival | 41.40 | 1.82 | 43.18 | 2.38 | 44.38 | 2.59 |
| (iii) Alternate to alternate | 41.46 | 1.70 | 43.31 | 2.21 | 44.55 | 2.37 |
| (iv) Departure triggering | 41.88 | 0.78 | 43.65 | 1.28 | 44.72 | 1.67 |
| (v) Chain departure | 41.88 | 0.76 | 43.68 | 1.23 | 44.76 | 1.60 |
| (vi) Arrival + departure | 41.89 | 0.77 | 43.66 | 1.27 | 44.73 | 1.66 |

**Table 4.** *The rates in erlangs of direct and alternate calls per link in a five-node network.*

From Table 5, we can conclude that the simple arrival and departure triggering schemes have blocking performance very close to that of the idealized rerouting scheme, indicating that more advanced rerouting schemes (e.g., second-order rerouting and rerouting alternate path calls to other alternate paths) cannot improve performance very much.

|  | D/N = 0.95 | D/N = 1.00 |
|---|---|---|
| (i) Arrival triggering | 10.06% | 13.10% |
| (ii) Second-order arrival | 10.00% | 12.99% |
| (iii) Alternate to alternate | 9.94% | 12.86% |
| (iv) Departure triggering | 10.20% | 13.20% |
| (v) Bound | 9.82% | 12.75% |

**Table 5.** *A blocking comparison of various triggering schemes.*

## Conclusions

This article presents a comprehensive study of rerouting schemes in circuit-switched networks and examines the complexity of implementing those schemes. We start from a taxonomy of rerouting by showing the various ways rerouting schemes can be designed. Due to the large number of possible rerouting schemes, we can only pick a few interesting ones for performance comparisons. Based on our limited scope of study on uniformly loaded fully connected circuit-switched networks, we can conclude the following:

- Random alternate path routing with random rerouting is as good as any of the more complicated variations.
- Arrival triggering schemes are better than departure triggering schemes. This suggests that rerouting should be performed only when necessary (i.e., when a new call is going to overflow) in order not only to reduce rerouting effort but, more important, to reduce call blocking (since wrong rerouting decisions can be avoided).
- Departure triggered chain rerouting has higher call blocking than simple departure triggering.
- The mixed (arrival-plus-departure) triggering scheme initiates more rerouting attempts while performing worse than the arrival or departure triggering schemes. It is therefore not worth considering.
- The additional options of second-order rerouting and rerouting an alternate path call to another alternate path when rerouting to the direct path is not possible can only reduce the call blocking probability by a very small amount. In other words, the simple arrival triggering scheme performs very well in reducing blocking, and there is little room for further improvement.
- Trunk reservation will increase the blocking probability when rerouting is used. Since rerouting has the self-stabilizing property, trunk reservation is not necessary for that purpose. However, trunk reservation can be used to regulate the rates of rerouting *and* alternate routing, and thereby the switching, processing, and signaling load imposed on the network.

## References


[1] G. R. Ash, *Dynamic Routing in Telecommunications Networks*, McGraw-Hill, 1997.

[2] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*, Springer, 1995.

[3] M. H. Ackroyd, "Call Repacking in Connecting Networks," *IEEE Trans. Commun.*, vol. 27, no. 3, Mar. 1979, pp. 589–91.

[4] A. Girard and S. Hurtubise, "Dynamic Routing and Call Repacking in Circuit-Switched Networks," *IEEE Trans. Commun.*, vol. 31, no. 12, Dec. 1983, pp. 1290–94.

[5] R. H. Hwang and J. F. Kurose, "On Virtual Circuit Routing and Rerouting in Packet-Switched Networks," *Proc. IEEE ICC '91*, 1991, pp. 1318–23.

[6] B. R. Venkatraman and R. E. Newman-Wolfe, "Performance Analysis of a Method for High Level Prevention of Traffic Analysis Using Measurements from a Campus Network," *Proc. 10th Ann. Comp. Sec. App. Conf.*, 1994, pp. 288–97.

[7] R. Cohen, "Smooth Intentional Rerouting and Its Applications in ATM Networks," *Proc. IEEE Infocom '94*, 1994, pp. 1490–97.

[8] A. Yamashita, R. Kawamura, and H. Hadama, "Dynamic VP Rearrangement in an ATM Network," *Proc. IEEE GLOBECOM '95*, 1995, pp. 1379–83.

[9] K. C. Lee and V. O. K. Li, "A Circuit Rerouting Algorithm for All-Optical Wide-Area Networks," *Proc. IEEE Infocom '94*, 1994, pp. 954–61.

[10] R. Beraldi, S. Marano, and C. Mastroianni, "A Reversible Hierarchical Scheme for Microcellular Systems with Overlaying Macrocells," *Proc. IEEE INFOCOM '96*, 1996, pp. 51–58.

[11] E. W. M. Wong, A. K. M. Chan, and T. S. P. Yum, "Re-routing in Circuit-Switched Networks," *Proc. IEEE INFOCOM '97*, 1997, pp. 1375–81.

[12] V. B. Iversen, "Traffic Engineering of Cellular Wireless Systems," Network Info. Processing Sys., IFIP TC6, Sofia Antipolis, France, , Oct. 1997.

[13] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network*, Addison Wesley, 1997, pp. 345–50.

[14] J. E. Flood, *Telecommunications Switching, Traffic and Networks*, Prentice Hall, 1995.

[15] M. Schwartz, *Telecommunication Network: Protocols, Modeling and Analysis*, Addison Wesley, 1988.


## Biographies


ERIC W.M. WONG (ewong@ee.cityu.edu.hk) joined City University of Hong Kong as an assistant professor in electronic engineering in 1994. He received his B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong in 1988 and 1990, respectively, and his Ph.D. degree in electrical and computer engineering from the University of Massachusetts at Amherst in 1994. His research interests are in high-speed networks, video on demand, satellite communications, and dynamic routing. The most notable of these involves the analytical modeling of the LLR scheme in circuit-switched networks. The model drastically reduces the computational complexity of designing and dimensioning telephone systems. The model has also served as a core for the analysis of many other advanced routing schemes, such as Real-Time Network Routing recently implemented in the AT&T telephone network.

TAK-SHING PETER YUM (yum@ie.cuhk.edu.hk) worked in Bell Telephone Laboratories for two and a half years and taught in National Chiao Tung University, Taiwan, for two years before joining the Chinese University of Hong

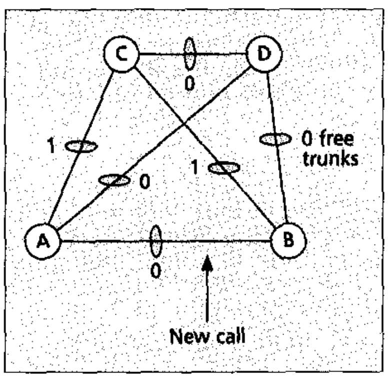
Kong in 1982. He has published original research on packet switched networks with contributions in routing algorithms, buffer management, deadlock detection algorithms, message resequencing analysis, and multiaccess protocols. In recent years he branched out to work on the design and analysis of cellular networks, lightwave networks, and video distribution networks. He believes that the next challenge is designing an intelligent network that can accommodate the needs of individual customers. His research benefits a lot from his graduate students. Six of them are now professors at local institutions. His recent hobby is reading history books. He enjoys playing bridge and badminton.

ANDY K. M. CHAN (kmchan@ee.cityu.edu.hk) received his B.Eng. and M.Phil. degrees in information engineering from the Chinese University of Hong Kong in 1992 and 1994, respectively. He is now working as a part-time research assistant at City University of Hong Kong, and his research interest is in network performance analysis.

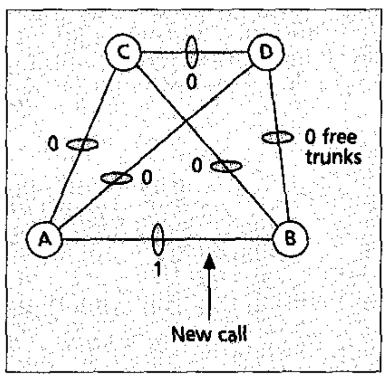
# APPENDIX

## ARRIVAL TRIGGERING PERFORMS BETTER THAN DEPARTURE TRIGGERING

Consider the situation in Fig. 4, where a direct call has just departed from a fully occupied link AC (freeing up a channel), and a new call has arrived on the fully occupied link AB subsequently. We assume that all calls on link AB are direct path call. In the case of departure triggering, an alternate call ADC, if any, will be rerouted back to its direct link AC, and the new call to link AB will be blocked since its direct path and alternate paths are all blocked. The case of arrival triggering is quite different. No call will be rerouted back to link AC after a call departure in link AC. Therefore, the new call could overflow to alternate path ACB. In this example, it is apparent that the rerouting of call ADC is unwarranted since it could lead to the loss of a new call. The result is not surprising since the direct path corresponding to an alternate call is normally very congested (otherwise, there would be no overflow calls in the very beginning). Therefore, unwarranted rerouting could only contribute more congestion on the direct path and potentially create more blocking. One the other hand, call ADC can stay there until it terminates or until, say, a new call arrives on link CD. In the latter case, call ADC could then be routed to link AC if arrival triggering is used.




■ **Figure 4.** *Arrival triggering vs. departure triggering.*




■ **Figure 5.** *Departure triggering vs. chain triggering.*

## DEPARTURE TRIGGERING REROUTING PERFORMS BETTER THAN DEPARTURE TRIGGERED CHAIN REROUTING

Consider the situation in Fig. 5, where a direct call has just departed from a full link AB, and a new call has arrived on link AB subsequently. We assume that all calls on link AB are direct path calls. In the case of departure triggering, let an alternate call ADB be rerouted back to link AB. In the case of chain triggering, let an additional call ACD be routed to link AD since there is a channel just freed up due to the rerouting of ADB. It is obvious that the new call on link AB will be lost in the case of chain triggering, but can survive by overflowing to alternate path ADB in the case of the simple departure triggering. Just as stated above, departure triggering will create more congestion on the direct link. Hence, departure triggered chain rerouting will aggregate even more traffic around the already congested direct link and causes higher blocking.