Connectivity, Performance, and Resiliency of IP-Based CDMA Radio Access Networks

Tian Bu, Mun Choon Chan, and Ram Ramjee

Abstract—IP-based Radio Access Networks (RAN) are expected to be the next generation access networks in UMTS and CDMA networks. The question of connectivity, i.e., how best to connect base stations to the Radio Network Controllers (RNC) in an IP-based RAN, has not been addressed by researchers. Furthermore, given a connection configuration, an RNC selection algorithm that assigns an incoming call to an RNC is also necessary. This paper examines RAN connectivity and its impact on the performance and resiliency of the wireless network using different RNC selection algorithms. For homogeneous networks, we show that the proposed Min-Load-1 algorithm, which allows at most one hard handoff in order to accommodate each incoming call request, delivers performance close to the optimal algorithm. We also show that allowing a few base stations to connect to two RNCs (a 10 percent increase in the number of links in our network) results in resiliency to RNC failures that is comparable to the resiliency of RANs with full-mesh connectivity. Finally, for heterogeneous networks, we show that the Min-Load-k algorithm (with at most k hard handoffs per call) is effective in handling load imbalances. These results provide strong motivation for deploying IP-based RAN, as they suggest that enhancing current point-to-point RAN with few additional links and allowing a few hard handoffs to accommodate incoming calls can result in significant gains in performance and resiliency.

Index Terms—CDMA IP-based RAN, connectivity, resiliency, RNC selection algorithms.

1 Introduction

Currently, third-generation wide-area wireless networks based on the CDMA2000 [1] and UMTS [2] are being deployed throughout the world. These networks provide both voice and high-speed data services to the mobile subscriber. According to the CDMA development group [3], as of June 2004, there were more than 110 million CDMA2000 subscribers. As the cost of these services are being reduced to attract more subscribers, it becomes important for the network operators to reduce their capital and operating expenses.

In wireless access networks today, the base stations and the radio network controllers are connected by point-to-point T1/E1 links as shown in Fig. 1a. These back-haul links are expensive and add to operating costs. Additionally, in this point-to-point architecture, the Radio Network Controllers (RNCs) are only shared by a small set of base stations (BSs) and can contribute to significant blocking during hot-spot and peak hours; thus, the network operator needs to appropriately scale-up the RNC capacity, thereby increasing capital costs. Furthermore, in this architecture, RNC is typically a single point of failure and is thus made highly redundant—this again increases the cost of each RNC.

One effective way to reduce these costs is to replace the point-to-point links with an IP-based Radio Access Network [4] (IP-based RAN). An architecture based on IP RANs is shown in Fig. 1b.

An IP-based RAN has a number of benefits, including:

- *Scalability:* RNC capacity can be shared with a larger set of base stations. By load balancing calls across the different RNCs, call blocking and dropping can be lowered.
- *Reliability:* When base stations are connected to multiple RNCs, failure of RNCs can be accommodated by transferring the calls from one RNC to another, thereby increasing reliability.
- *Cost:* Point-to-point links are expensive and cannot be shared. An IP-based RAN will benefit from statistical multiplexing gains and could also be shared with other applications (such as operator's wired network traffic) as long as appropriate QoS can be ensured (for example, using MPLS tunnels).

Note that today's standards [1], [2] do not specify how calls from one base station can be load balanced across multiple RNCs. This is primarily because of the existing point-to-point RAN architecture (Fig. 1a), where one base station is connected to one RNC. An IP-based RAN allows for the new possibility of one base station directly connecting to multiple RNCs. We hope that the aforementioned benefits of scalability and reliability will motivate the standards bodies to specify the mechanisms for load balancing calls from one base station into multiple RNCs.

IP is expected to be the access network for next generation UMTS networks. While IP RAN has to typically meet stringent delay and loss constraints, several researchers have proposed solutions for addressing quality of service (QoS) issues in IP-based RANs [4], [5], [6]. As shown in Fig. 1, use of IP-based MPLS tunnels between base

- *T. Bu is with Bell Labs, Lucent Technologies, 101 Crawfords Corner Road, Holmdel, NJ 07733. E-mail: tbu@research.bell-labs.com.*
- *M.C. Chan is with the School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Republic of Singapore. E-mail: chanmc@comp.nus.edu.sg.*
- *R. Ramjee is with Bell Labs, Lucent Technologies, 600-700 Mountain Avenue, Murray Hill, NJ 07974. E-mail: ramjee@research.bell-labs.com.*

Manuscript received 14 Sept. 2004; revised 30 Apr. 2005; accepted 4 July 2005; published online 15 June 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0265-0904.

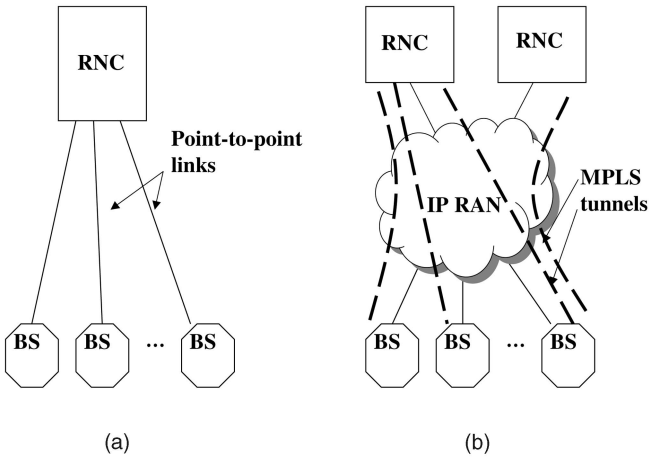


Fig. 1. Wireless access network architectures. (a) Current RAN. (b) IP-based RAN.

stations and radio network controllers is another viable approach for providing QoS in the access network.

While these studies have shown the feasibility of an IP-based RAN to support quality of service requirements in wireless access networks, the question of connectivity, i.e., how best to connect base stations to the radio network controllers in the IP-based RAN, has not been addressed by any research literature to our knowledge. While IP networking provides full mesh, best effort connectivity, enabling full mesh connectivity with QoS constraints between base stations and RNCs is very expensive. In addition, enabling full mesh connectivity may not be necessary and may have little incremental impact on performance beyond a certain level of connectivity. Thus, understanding the problem of connectivity and analyzing the impact of connectivity on the performance of the RAN is essential for the success of transitioning current point-to-point RANs to IP-based RANs. Note that analyzing connectivity is a hard problem since, even for a simple network with 100 base stations and 10 RNCs, the number of possible connection configurations between the base stations and the RNCs is enormous ($\approx 2^{1,000}$).

Furthermore, given a connection configuration, we also need an algorithm to select an RNC for an incoming call (since IP-based RANs enable base stations to connect to more than one RNC). Note that the incoming call can be a new call or a handoff call. The RNC selection algorithm needs to ensure that both the dropping of handoff calls and the blocking of new calls are minimized with priority given to handoff calls. While there exists some similarity between our problem of load balancing calls across RNCs and traditional distributed systems load balancing problems, there are two important differences. First, load balancing mechanisms in distributed systems [7], [8], [9] were designed so that idle machines in a network of workstations could be transparently used. Thus, the choice of a processor on which to execute a process was primarily based on the load conditions in the processors. However, in our case, the choice of the RNC on which to assign a call is determined both by the load conditions as well as the current location of the mobile user and the connectivity of base stations to RNCs (that is, in turn, determined by proximity of the base station to the respective RNC, given the QoS constraints). The second difference between the two problems is the impact of moving a call. Since traditional process migration techniques [10] which implement load balancing in distributed systems are general purpose mechanisms, they result in a considerable overhead in migration. In our case, moving a call from one RNC to another is already a well-defined and efficient feature of the RNC, called the *hard handoff*. The only drawback in moving a call is that the user might hear a "click" during the conversation and, thus, it is desirable to minimize the number of hard handoffs per call.

In this paper, we make three main contributions: First, we systematically evaluate different ways of connecting base stations to RNCs and provide insights into the minimum connectivity that is necessary to obtain maximum performance gain. Second, we evaluate the performance under different failure scenarios (such as RNC failure, base station failure, link failures, etc.) and propose resilient IP-RAN topologies that suffer minimum degradation in performance during failures, while requiring few additional links. Finally, we propose a load balancing algorithm called *Min-Load-k* that can achieve the maximum performance gain with the minimum set of connectivities and is particularly effective in handling load imbalances in heterogeneous networks. The Min-Load-k algorithm assigns calls to RNCs such that the RNC load is balanced. It uses hard handoff to redistribute the load dynamically while placing a bound on the number of hard handoffs (k) required to fulfill the assignment. While this algorithm incurs a cost in terms of an increased number of hard handoffs, we show that a simple reservation-based scheme that reserves a few call resources of the RNC can significantly reduce this cost.

We compare the performance through extensive simulations of the Min-Load-k algorithm with an online optimal algorithm that has no hard handoff constraints. *We find that, by allowing at most one hard handoff in order to accommodate each new request,* Min-Load-1 *achieves performance that is very close to the optimal algorithm. We also find that using the Min-Load-1 algorithm and allowing the base stations to connect to two RNCs results in a resiliency to RNC failures that is similar to having full-mesh connectivity between base stations and RNCs. Finally, we find that the* Min-Load-k *algorithm is effective in handling load imbalances and uses larger values of k to accommodate greater load imbalances.*

The rest of the paper is structured as follows: In Section 2, we review related work. In Section 3, we present an overview of the problem. In Section 4, we present our approach to make the connectivity problem between base stations and RNCs tractable by systematically evaluating different connection topologies. In Section 5, we present several algorithms for RNC selection and an analytical model for the optimal algorithm. In Section 6, we evaluate the impact of connectivity between base stations and RNCs on the overall performance and the resiliency of the network. In Section 7, we evaluate the impact of heavy-tailed holding/sojourn time, soft handoff, and finite capacity at the base station on the observations we made in previous sections. In Section 8, we study the impact of reservations to reduce the number of hard handoffs. In

Section 9, we discuss issues with modeling heterogeneous networks and show that the *Min-Load-k* algorithm is effective in handling reasonable load imbalances. Finally, in Section 10, we present our conclusions.

2 Related Work

All-IP wireless networks are getting increasing attention recently both by the industry [11], [12] and by researchers [13]. While all-IP wireless networks encompass both local-area and wide-area networks, we focus on work related to wide-area all-IP networks, and in particular, IP RANs.

IP RANs provide significant benefits including higher scalability, higher reliability, and lower cost. Thus, IP RAN is provided as an option in the UMTS 3G Standards [14]. However, IP-based wireless networks also pose significant challenges such as mobility management and Quality of Service (QoS) support. The authors in [15] provide an overview of the various issues faced in introducing IP in the UMTS radio access network.

Several researchers have examined QoS issues in IP-based wireless networks [4], [5], [16]. The authors in [4] and [16] propose reserving resources in order to support QoS because inaccurate resource estimation due to dynamic load patterns and/or mobility can lead to inefficiencies. Instead of reserving resources, the authors in [5] propose congestion control policies that reduce the impact of congestion in a best-effort IP RAN, thereby allowing adequate QoS support to be maintained.

Mobility management issues in IP-based wireless networks have also been investigated by several researchers [16], [17], [18]. In [17], authors propose a fast Serving RNC relocation scheme in UMTS RANs that do not require packet duplication in order to reduce packet losses during handoff. The authors in [16] propose the use of integrated resource reservation and handoff to reduce service disruption to mobile users.

While these studies have shown the feasibility of an IP-based RAN to support quality of service and mobility management requirements in wireless access networks, in this paper, we address the question of connectivity, i.e., how best to connect base stations to the radio network controllers in the IP-based RAN.

3 Problem Setting

As shown in Fig. 1, the wireless access network consists of a set of base stations (BS) that are managed by a Radio Network Controller (RNC). A Radio Access Network (RAN) connects the BSs to the RNCs. The RNC performs a number of functions [5], including soft handoffs, reverse outer loop power control, and termination of the Radio Link Protocol (RLP) for data users.

The abstract network architecture analyzed in this paper has the following components: a set of RNCs, R, a set of base stations, B, and a set of communication links, L, that connect the base stations to the RNCs and a set of users, U. Note that, in practice, the logical communication links may translate either to a T1 leased line, an ATM connection, or an MPLS path, and many logical links may traverse the same physical link. This logical connection provides Quality of Service necessary to ensure that CDMA soft handoff functions correctly. A user in the network can be either *active* or *idle*. A user, whether *active* or *idle*, is associated with a base station. An active user needs radio resource from a base station and processing resource from an RNC.

Two types of user events are modeled: voice call events and mobility events. We focus on the voice application for two reasons: 1) current cellular networks are predominantly used for voice transmission and 2) voice has higher QoS and hard handoff requirements than data (where retransmission is an option). Call events can be either an arrival or a departure event. Call arrivals for a user is Poisson distributed with mean λ and call duration is modeled as exponentially distributed with mean $1/\mu$. A successful call arrival event changes a user's state from idle to active. A mobility event occurs when a user roams from one base station to another. After the movement, the user stays in the new base station for a period of time that is exponentially distributed with mean $1/\gamma$ before moving again. It is assumed that mobility and call events are independent and may not occur at the same time. These are common assumptions and are used in [19], [20]. For a call event, we are interested in *call blocking rate*, the average rate of blocking a new call. For a mobility event, we are interested in *call dropping rate*, the average rate of dropping an existing call.

As the focus of this paper is in the study of RAN connectivity and RNC utilization, we do not place capacity constraints on base stations and communication links for now. Consequently, blocking or dropping a call can only occur due to insufficient RNC capacity. Note that we are considering the aggregate arrival of calls from many BSs to RNCs, and the blocking and dropping rates assumed are low. As a result, even though call blocking and dropping due to insufficient radio capacity on the base stations may be common in practice, the relative results obtained for call blocking and dropping rates at the RNCs are still valid, though the actual rates might be lower. We evaluate the impact of finite capacity at base station and communication links in Section 7.3.

As mentioned earlier, we are interested in exploring two important and related aspects of RAN performance in this paper. First, we are interested in understanding how connectivity impacts the performance of the network. In other words, we would like to answer the question of how the RAN should be connected with few additional links while obtaining the maximum gains in performance and resiliency. Second, we would like to answer the question of what algorithm should be used to select the RNC for a call so that call blocking and dropping are minimized. These two issues are interrelated as the choice of algorithms depends on the RAN connectivity and vice versa. In particular, when hard handoff is used as a call reassignment mechanism in the RNC selection algorithm, the connectivity needs to be designed such that the reassignment capability can be exploited to the fullest.

The issue of designing the connectivity of the RAN is presented next in Section 4 and the RNC selection algorithms are presented in Section 5.

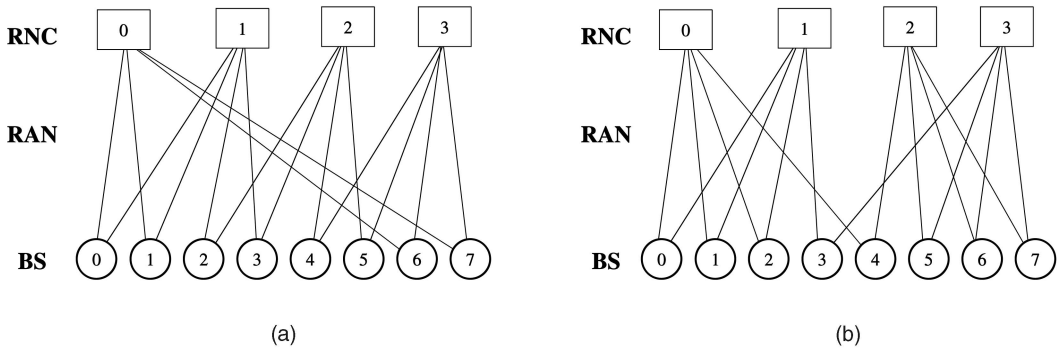


Fig. 2. Two RAN topologies with arc connectivity of two.

4 Designing the RAN Topology

The number of possible configurations in a RAN graph with M BS and N RNC is 2^{NM}. Even though some of these configurations are not interesting, for example, the set of configurations where one or more nodes (RNC or BS) are isolated, the remaining set of possible configurations is still enormous. In order to make this problem tractable, in this section, we systematically study a much smaller set of graphs with well defined and desirable properties. These graphs are representative of the range of connectivity from a mesh connectivity between the BSs and RNCs to a single-connected graph where each BS is connected to exactly one RNC.

Before we proceed further, we need to define the concept of graph connectivity. The presentation here follows [21]. A graph is *connected* if there is at least one path between every pair of nodes. The *arc connectivity* of a connected graph is the minimum number of arcs whose removal from the graph disconnects it into two or more components. For example, with N RNCs and M BSs ($M > N$), a mesh connectivity has $M \times N$ links and is of arc connectivity N.

Our approach is to focus on a set of *balanced graphs* with properties that are desirable in a homogeneous network where RNCs have the same capacity and the BSs have the same average load. Each element in this set of balanced graphs has a different number of links L and we can enumerate members of this set by varying the number of links L from M to NM. By focusing on this set of balanced graphs, we have reduced the connectivity problem from the original state space of 2^{NM} to NM. Given that there is very little known in the literature even about the impact of connectivity on homogeneous networks, we now focus on the homogeneous network case. The issues in modeling a heterogeneous network are discussed in more detail in Section 9.

The balanced graphs are first defined using the following conditions:

1. The number of BS connected to any RNC cannot differ by more than 1.
2. The number of RNC connected to any BS cannot differ by more than 1.

This set of graphs also has the following properties. First, their arc connectivities vary from 0 to N. The arc connectivity of a graph with L links is $k = \lfloor \frac{L}{M} \rfloor$. The set of graphs with the minimum number of links to maintain an arc connectivity of $k = 1 \quad to \quad N$ (which has kM links) is part of this set and we will refer to a member in this set of graphs as the *minimum connected balanced graphs* with arc connectivity k.

The two conditions defined are insufficient to construct a set of useful balanced graphs. Fig. 2a and Fig. 2b show two ways of constructing a minimum connected balanced graph with four RNCs, eight BSs, 16 links, and an arc connectivity of 2. In order to differentiate among the different minimum connected balanced graphs, we introduce the concept of an RNC accessibility tree for a BS i. The RNC accessibility tree for BS i is constructed as a spanning tree rooted at BS i that connects all RNCs using a breadth-first search. The weight of each arc in the spanning tree is defined to be the number of base stations connecting two RNCs which are at two ends of the arc. Thus, except for the root, all the vertices in this graph represent the different RNCs in the network.

Using Fig. 2a and Fig. 2b as examples, the corresponding RNC accessibility graphs are shown in Fig. 3a and Fig. 3b, respectively. In Fig. 3a, the RNC accessibility graph for BS 0 is shown. Due to the regular structure of the network in Fig. 2a, all BSs have similar RNC accessibility graphs. In Fig. 3a, there is one path from BS 0 to RNCs 0 and 1. From RNC 0, there are two paths to RNC 3 (through BS 6 and 7) and, from RNC 1, again there are two paths to RNC 2 (through BS 2 and 3).

In Fig. 3b, the RNC accessibility graph for BS 0 has one path to all RNCs and the graph for BS 3 has one path each to RNC 1 and 3 and three paths each to RNC 0 and 2. Obviously, the graph in Fig. 3a is more balanced. In fact, due to its regular structure, it is the most balanced RNC accessibility graph possible.

The concept of an RNC accessibility graph is very useful in predicting the impact of connectivity on performance since it captures the impact of dynamic load balancing using call reassignment (hard handoffs)—the more RNCs that are accessible from a given BS, the greater the impact of reassignment; the larger the arc weights, the more possibilities (paths) where calls can be reassigned from one RNC to another. A balanced arc weight across all paths where the smallest arc weight is maximized is the most preferable graph (Fig. 3a) since we are focusing on homogeneous networks. Furthermore, the depth of the RNC accessibility graph indicates the maximum number of hard handoffs that may be necessary in order to free up capacity to accept a new call (the k in the Min-Load-k algorithm). Thus, an RNC accessibility graph with a small depth is preferred since a

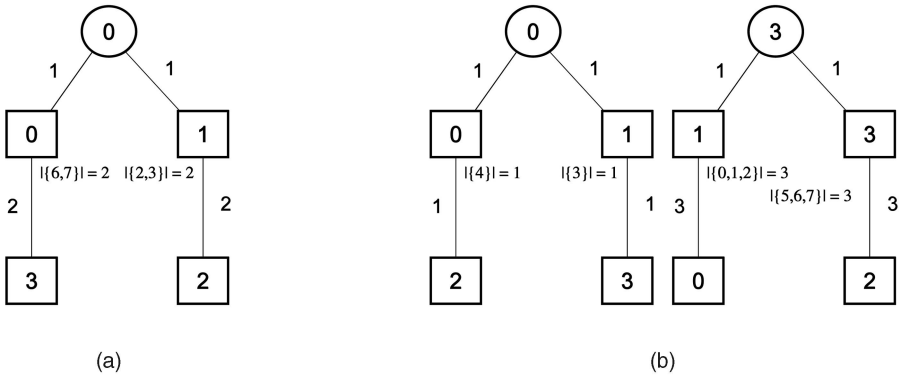


Fig. 3. RNC accessibility graph of BS 0 and 3. (a) Graph for Fig. 2a. Graph for Fig. 2b.

smaller number of hard handoffs are sufficient to attain maximum performance.

In the case of a full-mesh connected network, the RNC accessibility graph (identical for all base stations) is of depth 1 with the arc weight, for all arcs, equal to 1. Clearly, this is the best possible configuration for maximizing performance. However, full-mesh connectivity is expensive. Thus, we are interested in identifying a connectivity graph that adds the minimum number of links to a point-to-point RAN while providing close to the maximum performance obtainable in a full-mesh connected network.

A balanced graph whose corresponding RNC accessibility graph is also balanced can be constructed in the following way. Let there be L links and the BSs be labeled from 0 to $M-1$ and RNC from 0 to $N-1$. Initially, each BS i is connected to $k = \lfloor \frac{L}{M} \rfloor$ RNCs starting from RNC $\lfloor \frac{iN}{M} \rfloor$ using a total of kN links. If $L > kN$, excess links are added one per BS such that conditions 1 and 2 are satisfied.

The rationale for considering this set of balanced graphs should now be clear since such graphs maximize the performance for a homogeneous network where all the RNCs in the network have the same capacity and the average load on each of the BS is the same. Furthermore, due to the "balanced" nature of these graphs, the behavior of different instantiations of these balanced graphs with the same L is the same.

The concepts of balanced graph and RNC accessibility graph reduce the state space of connectivity configurations from 2^{NM} to NM, while retaining the important configurations that maximize performance. This makes the connectivity problem tractable and will help us select between different connectivities possible for the same number of available links in the RAN and identify a suitable connectivity graph that shows the greatest promise for sharing of RNC resources and, thereby, improving RAN performance. However, even given a connection topology for the RAN, we still need an RNC selection algorithm for assigning calls to RNCs that will fully exploit this connectivity. This topic is discussed in detail in the next section.

5 Algorithms and Analytical Model

When a new call arrives at a base station or an existing call roams to a base station, a *RNC selection algorithm* is necessary to select an RNC r to serve the call among all RNCs directly connected to the base station. In this section, we first introduce three RNC selection algorithms, the Min-Load algorithm, the optimal algorithm, and the Min-Load-k algorithm. We then present an analytical model for the optimal algorithm.

Before presenting the details of the algorithms, we first list some notations that we will use in the algorithm description. Let A be an $|R| \times |B|$ adjacency matrix where $A(r,b) = 1$ if RNC r and BS b is directly connected by the RAN. $R_b = \{r | r \in R, A(r,b) = 1\}$ is the set of RNC that base station b directly connects to. We denote the number of active calls associated with base station b and served by RNC r by $C(r,b)$. Let $D(r)$ be the load at RNC r. The load value used in this paper is the normalized load defined as the ratio of the number of active calls supported by the RNC over the total RNC capacity. We summarize the notations in Table 1.

5.1 RNC Selection Algorithms

Min-Load algorithm: When a call request (either a new or a handoff call) arrives at BS b, and at least one of the RNCs in R_b is not full, the Min-Load algorithm selects the RNC with the minimum load among the set of RNC R_b. Otherwise, the call is rejected. This is the simplest algorithm used and is the basis for performance comparison.

Optimal algorithm: When a call request arrives, the optimal algorithm attempts to admit the call as long as there is a feasible solution. In order to do so, the algorithm treats the new request as if it has been accepted and then tries to

TABLE 1
Notations for Algorithms

Notation	Explanation
R	RNCs in RAN
B	BSs in RAN
N	Number of RNC
M	Number of BS
A	Adjacency matrix of RNCs and BSs
R_b	RNCs directly connected to BS b
$D(r)$	The normalized load at RNC r
$C(b,r)$	Calls associated with BS b served by RNC r

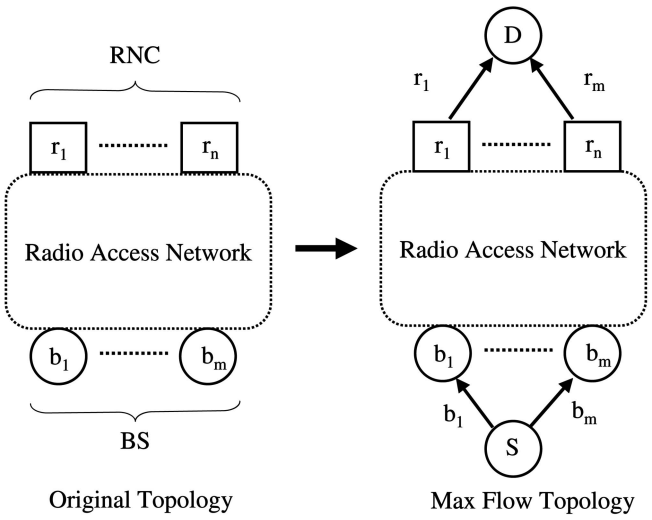


Fig. 4. Transforming optimal algorithm into a Max Flow problem.

find a feasible solution with the new set of load configurations. The feasible solution is solved by formulating it as a maxflow problem as illustrated in Fig. 4. A set of BSs, each with b_i active users, and a set of RNCs, each with capacity r_j, is shown on the left in Fig. 4. The graph is transformed by adding a source node (S) which is connected to all BSs and a destination node (D) which is connected to all RNCs. The link capacity between S and BS i is set to b_i and the link capacity between D and RNC j is set to r_j. As a result, by finding the maximum flow for the graph on the right in Fig. 4, we can decide if the new request can be accepted or not. The max-flow problem is a well-known problem and will not be described in more detail here. Interested readers can refer to [21]. Assuming the maximum flow value to be f, if $f = \sum_i b_i$, then the new request is admitted. Otherwise, it is rejected. Note that there might be multiple placements of active calls to RNC for a single value of max flow. It is obvious from the maxflow graph that the new request cannot be accepted if $\sum_j r_j < \sum_i b_i$.

Another way to view the optimal algorithm is that, in order to satisfy a new request, it is possible to move/reassign existing calls such that RNC resources can be freed up to accept the request. Such movement or reassignment can be interpreted in practice as performing hard handoffs. Hard handoff results in service degradation for the call being moved but may be an acceptable cost if it allows a new call to be accepted or a call is allowed to move into a BS without being dropped. While the optimal algorithm maximizes the chances of a call being accepted, it does not take into account the number of hard handoffs that may be necessary to accept a call request. This leads us to the third and last algorithm.

Min-Load-k algorithm: This algorithm extends the Min-Load algorithm by allowing up to k hard handoffs such that a call request can be satisfied. An example of how a Min-Load-1 algorithm works is shown in Fig. 5. When a new call arrives at BS 3, if RNC 2 is full, then the call will be blocked by Min-Load, which does not allow reassignment. However, with reassignment, an active user from BS 2 that is served by RNC 2 can be moved to RNC 1 through a hard handoff and the new call can be served by RNC 2. Note

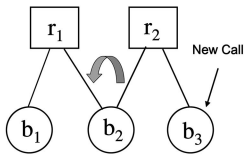


Fig. 5. Reassign an existing call from r_2 to r_1 to accept a new request.

that, if no call from BS 2 is served by RNC 2, or RNC 1 is full, then the call will still be blocked. The pseudocode of Min-Load-k algorithm is shown in Fig. 6. In the algorithm, b is the base station where a call arrives.

In order to have a better understanding of the Min-Load-k algorithm, we can convert the snapshot of a RAN to a directed reassignment graph when a call arrives. In the reassignment graph, each node is either a base station or an RNC, the capacity/bandwidth of a directed link from a base station to an RNC is $+\infty$, and the capacity/bandwidth of a directed link from an RNC to a base station is the number of calls associated with the base station and served by the RNC. For instance, the RAN in Fig. 5 can be converted into the reassignment graph in Fig. 7. Starting at the base station where the new call arrives, the Min-Load-k algorithm traverses the graph in a breadth-first manner until it either reaches an RNC with nonzero available capacity through a nonblocked path or the maximum depth is reached. A path is blocked if the capacity/bandwidth of any directed link on the path is zero. When there are multiple RNCs with nonzero available capacity at the same

Min-Load-k(b, k) {
  for $i = 1$ **to** k **do**
    l = try(b,k)
    if $(l < 1)$ admit the call
    else block the call
}
try(b,k) {
  //check for the RNC with the minimum load
  $x = \min(D(r)), \forall r, A(r, b) = 1$
  //if all RNCs are full, go one more level, otherwise return
  if (x < 1)
    return x
  else
    if (k=0) **return** 1
    else
      $x = \min(try(b', k-1)),$
        $\forall b', \forall r, A(b, r) = 1 \cap A(b', r) = 1 \cap C(b', r) \neq 0$
      return x
}

Fig. 6. Min-Load-k functions.

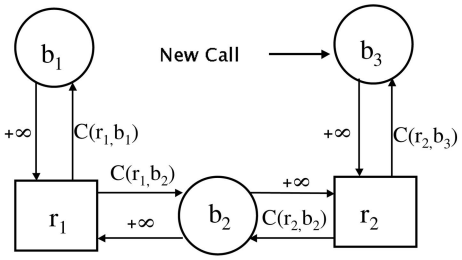


Fig. 7. Reassignment graph.

depth, the algorithm selects the one with the minimum load. The algorithm probes to a maximum depth of $2k-1$ for a Min-Load-k algorithm. If no RNC can be reached within maximum depth, the call is blocked. The blocking rate decreases as k increases until $2k$ reaches the diameter of the reassignment graph. When the search depth reaches the diameter of the graph, all RNCs have been visited and searching beyond will yield no additional resource.

In practice, the Min-Load-k algorithm runs as a distributed algorithm that is initiated at the base stations with help from the RNCs. It is important to keep k as small as possible since a large k incurs more hard handoff and larger call setup time. We are interested in exploring how large k needs to be (without reaching the graph diameter) in order to exploit the added flexibility of reassigning calls through hard handoff.

5.2 Analytical Model

In this section, we present an analytical model for the optimal algorithm. If we assume that the number of users in the system is constant, the system can be modeled as a closed migration process that is based on the approach described in [22]. In a closed migration process, users move randomly from one queue to another and the movement is governed by the transitional rate from one state to another.

For every base station i, we are interested in two state variables, the number of active users a_i and the number of idle users d_i. We model each base station with two queues, one for the active users and one for the idle users. The state of the system is thus completely defined by the vector $\{a_1, d_1, \ldots, a_M, d_M\}$. The feasibility of a set of $a_1, a_2, \ldots, a_M$ depends not only on the RNC capacities, but also the RAN connectivity, which can be checked by using the max flow graph, e.g., Fig. 4. In addition, $\sum a_i + \sum d_i = |U|$ is the total number of users in the system. Let the feasible set of $\{a_1, d_1, \ldots, a_M, d_M\}$ vectors be denoted by ζ.

Note that moving from a_i to a_j and d_i to d_j indicates moving from one base station to another. Moving from a_i to d_i and d_i to a_i indicates a user in base station i going from active to idle and vice versa. Therefore, λ_{jk} represents either the movement rate for a single user among base stations, the call arrival rate, or the call departure rate. In order to simplify the notation, we rewrite the state vector as $x = \{x_1, x_2, \ldots, x_{2M}\}$, where $x_1 = a_1, x_2 = d_1$, and so on. $\lambda_{jk}x_j$ is the transition rate going from state j to state k.

Let $\alpha_1, \alpha_2, \ldots, \alpha_{2M}$ be the unique collection of positive numbers, summing to unity, that satisfy

$$\alpha_j \sum_k \lambda_{jk} = \sum_k \alpha_k \lambda_{kj}, j = 1, 2, \ldots, 2M. \quad (1)$$

For a given set of mobility rates, arrival rates, and call holding times, the system is stable only if there are solutions to the set of simultaneous equations given in (1). α_j is the equilibrium probability that a user is in state j.

The system satisfies Theorem 1.7 and Corollary 1.10 in [22], thus has a product form solution. Using Theorem 2.3 in [22], the equilibrium distribution for the system is given as

$$\pi(x) = B_x \prod_{j=1}^{2M} \frac{\alpha_j^{x_j}}{\prod_{r=1}^{x_j} r}, x \in \zeta, \quad (2)$$

where B_x is the normalization constant, chosen such that the distribution sums to 1.

Using (2), the blocking rate and dropping rate is computed in the following way: First, enumerate all blocking and dropping states. A state is a blocking state if a new call arrival can result in a call being blocked. However, unlike a single queue system, not all call arrivals result in a call being blocked in our system. The blocking probability in our system is obtained by multiplying the equilibrium distribution by the ratio of the sum of transitional rates at which calls can be blocked over the sum of all transitional rates. Similar computation is used for computing the dropping rate. We use the analytical model to verify the simulation results for optimal algorithm and find that the blocking probabilities obtained using these two different approaches are close. Unfortunately, we cannot compute the blocking probability for larger RAN (with larger N and M) by enumerating all states and applying (2) due to state space explosion. Instead, the Monte Carlo approach [23] can be used.

So far, we have only modeled the optimal algorithm. Our state variables are the number of active and idle users at each base station. Using the optimal algorithm, the next state is determined solely by the current state and the event (mobility, new call, etc.)—this is because the optimal algorithm can potentially reassign all calls. However, for a general assignment algorithm with a limit on the number of reassignments, the optimal move depends on both the current assignment and the network topology. This results in an explosion in the state space and makes modeling the Min-Load and Min-Load-k algorithms intractable for reasonably sized topologies. For example, analytical models for reassignment algorithms are only available for simple topologies [24]. Thus, in this paper, we evaluate the Min-Load and Min-Load-K algorithms using simulations in Section 6.

6 Evaluation

In this section, we present a detailed simulation-based evaluation of the performance of the wireless access network. We first describe our simulation setup and the performance measures of interest. In Section 6.2, the performance of the various algorithms are compared. In Section 6.3, we perform detailed evaluation of the impact of connectivity on the various algorithms. In Section 6.4, the resiliency of the various connectivity graphs in the presence of a single link, BS, or RNC failure are evaluated. In

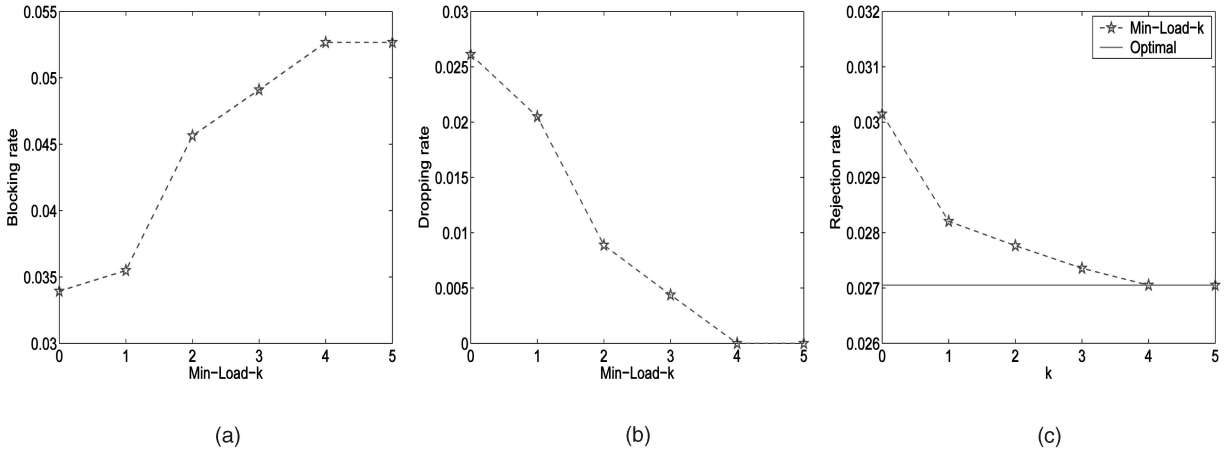


Fig. 8. Performance of different algorithms using a minimum connected graph with arc connectivity 2. (a) Blocking probability. (b) Dropping probability. (c) Rejection probability.

Section 6.5, the cost of the various algorithms, measured in number of hard handoffs performed, is presented.

6.1 Simulation Setup

The Radio Access Network simulated has 10 RNCs and 100 BSs. Each RNC can process up to 500 calls simultaneously. The calls arrive at each base station according to a Poisson process with rate $\lambda = 0.003$. The call holding time is exponentially distributed with mean $1/\mu = 1$. There are a total of 2,250,000 users in the system. A user roams among base stations at rate $\gamma = 1$. We lay out all base stations on a two dimensional plane, where each base station has four neighboring base stations. When a user roams, it has the same probability to roam to any one of the four BSs that are neighbors of the one its currently associated.

The performance metrics measured are call dropping and call blocking probabilities. These two measures, while different, are not independent. For instance, assuming a network of fixed capacity, by blocking more calls, one necessarily decreases call dropping since more resources are available for handoff calls. This is the idea behind the use of guard channels for reducing call dropping [19], [20]. Thus, an algorithm may reduce the dropping probability and increase the blocking probability or vice versa. Cellular operators are typically interested in minimizing a weighted sum of these measures, with higher weight allocated to call dropping. However, the choice of appropriate weighting is not clear. Instead of using a weighted sum of these probabilities, we define a single performance metric called the *rejection probability*, which is computed as the ratio of all call requests (new call and handoff) that are rejected to the total number of call requests (new call and handoff). This is an excellent measure of the algorithms in this paper since a lower rejection probability automatically implies better utilization of RNC resources and, hence, a better algorithm. Complementing these algorithms with guard channels [19], [20] can help control the relative preference between blocking and dropping probabilities, but this issue is outside the scope of this paper.

In our simulations, we terminate the runs when the rejection rates converges, i.e., the rejection rate varies less than 0.1 percent for a period of time. In practice, we double the simulation duration each time until the rejection rates vary less than 0.1 percent.

6.2 Algorithms

In this section, we evaluate the different RNC selection algorithms, i.e., Optimal, Min-Load, and Min-Load-k using a minimum connected balanced graph with arc connectivity 2. This particular connectivity is used because it is the graph with the smallest L such that all BSs are connected to at least two RNCS. For graphs with lower connectivity, some base stations are connected to only one RNC and the selection algorithms have no choice in RNC selection.

Fig. 8 plots the blocking, dropping, and rejection probabilities for Min-Load-k algorithms as k increases. The Min-Load algorithm is indicated as Min-Load-0. The rejection probability of the optimal algorithm is also plotted in Fig. 8c as a solid line for comparison (the blocking probability is 0.053 and the dropping probability is zero for the optimal algorithm). From the figure, we observe that the rejection probability of Min-Load-k approaches that of optimal as k increases. At $k = 4$, the rejection probability achieved by Min-Load-k is the almost the same as the optimal algorithm. The biggest improvement comes from going from Min-Load-0 to Min-Load-1 showing that the even a small amount of flexibility to reassign calls provides a significant performance improvement. Note that we only plot k up to 5, which is the diameter of the graph. Increasing k to more than the diameter of the graph does not reduce the rejection probability any more, as explained earlier.

6.3 Connectivity

In this section, we evaluate how the connectivity of RAN impacts the network rejection probability when different RNC selection algorithms are used. The connectivity of the graphs are varied in the following way: First, we vary the graphs from a single-connected graph to a complete graph by looking only at minimum connected balanced graphs (with arc connectivity 2 to N). The number of links L is therefore incremented in units of 100 (M). This is shown in Fig. 9a. Next, we evaluate graphs between single-connected and a minimum connected balance graph of arc connectivity 2 by increasing L in increments of 10 (N). This is shown

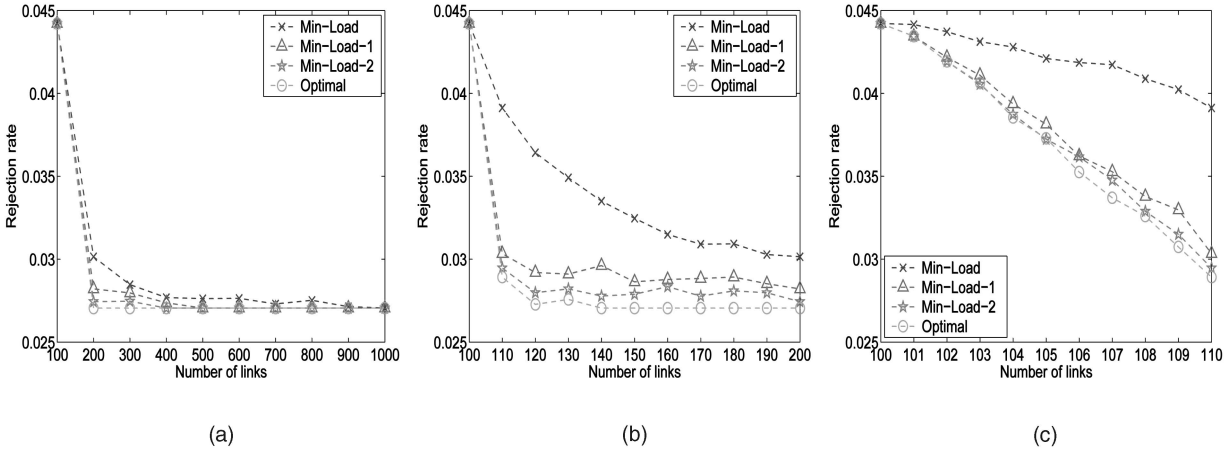


Fig. 9. Rejection probabilities for various connectivities. (a) Single connected to fully connected RAN. (b) Single connected to arc connectivity 1. (c) Single connected with a few additional links.

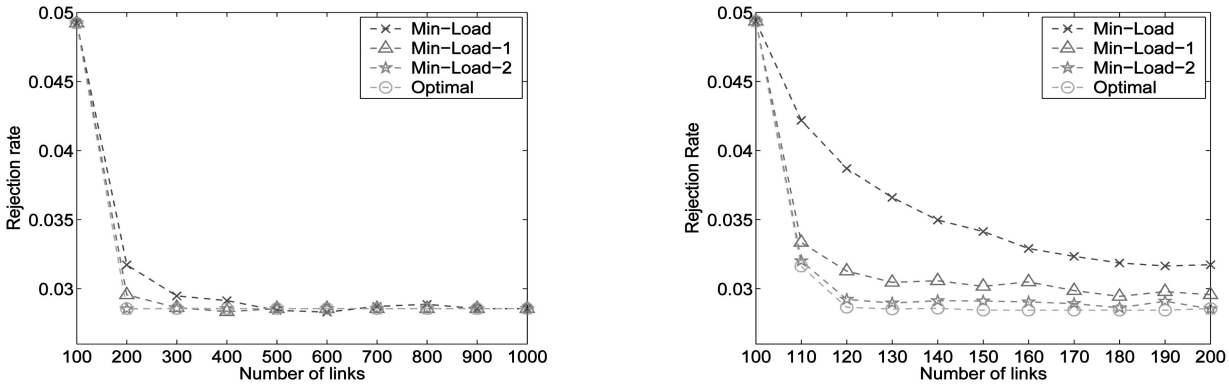
in Fig. 9b. Finally, we evaluate all the connectivity graphs between the single-connected case and single-connected case with N extra links by examining them in increments of one. This is shown in Fig. 9c. The RNC selection algorithms, Optimal, Min-Load, Min-Load-1, and Min-Load-2, are evaluated for all of the connectivities considered.

From Fig. 9a, we observe that the rejection probability drops significantly from the single-connected (100 links) graph to the RAN with arc connectivity of two (200 links). However, adding more links to a RAN of arc connectivity 2 does not reduce the rejection probability significantly. This is true for all four RNC selection algorithms shown, including the Min-Load algorithm. In addition, we see that the Min-Load-1 algorithm performs much better than the Min-Load algorithm and the difference between Min-Load-1 and Min-Load-2/Optimal is small. These differences become even smaller as the RAN becomes more connected. Note that all four selection algorithms perform the same on the single-connected graph because each BS only connects to one RNC and there is no alternative RNC to select. The large performance improvement from a single-connected graph to a graph with arc connectivity 2 motivates the next graph, which zooms into the set of graphs with connectivities between the single-connected and arc connectivity 2 cases.

Fig. 9b plots the rejection probability of RANs as we add links in increments of 10 to a single-connected graph. The x-axis is the number of links in RAN. In constructing the balanced graph using the methodology outlined in Section 4, each time we add 10 links, we select BSs with the lowest connectivity and each link is connected to a different RNC. Fig. 9b shows that the rejection probabilities decrease dramatically for the Min-Load-1/Min-Load-2 and Optimal algorithms after we add just one more link to each RNC. As more links are added, the rejection probability decreases at a much slower rate. This suggests that most of the performance gain (rejection probability reduction) occurs during the addition of the first 10 links to the single-connected graph. This can be explained by recalling in the reassignment graph (Fig. 7) that we constructed in Section 5. Reassignment can be visualized as visiting the directed graph in a breadth-first manner until an RNC with nonzero available capacity is reached. In a single-connected graph, the directed graph is disconnected and no reassignment can be performed. By adding 10 links in the way we have described, the directed graph becomes a connected graph with diameter 5. In the connected directed graph, the probability of reassignment or finding a path to a RNC with nonzero available capacity is greatly enhanced. The dramatic decrease in network rejection probability is not observed for Min-Load, which has a more gradual decrease. This is because reassignment is not performed and the gain from statistical multiplexing increases more gradually with the additional links.

Again, since the most performance improvement occurs between the first two points in Fig. 9b, we now look further to see how the rejection probability changes as we add one link at a time to a single-connected graph. Fig. 9c plots the rejection probability as we add up to 10 links. Observe that Fig. 9c is different from Fig. 9a and 9b in that there is no dramatic decrease in rejection. The decrease in rejection probability is almost linear, showing that the performance gain is directly proportional to the number of new links added. We have also repeated the simulations with lower and higher rejection probability ranges for the network. The observation is similar.

Summarizing our observations, we find that Min-Load-1 performs significantly better than Min-Load and its performance is very close to that of more complicated schemes such as Min-Load-2 and Optimal. In terms of connectivity, when Min-Load-1 is used, a balanced graph constructed with a single-connected graph with N extra links achieved a rejection probability of 0.03 (from 0.045), the same rejection probability achieved by a Min-Load algorithm using a graph with arc connectivity 2. This is a savings of 45 percent in terms of link cost for the same performance. Bringing the rejection probability down further (to 0.027) requires many more links to be added and/or more complicated algorithms and is not cost effective. In conclusion, we find that *allowing at most one hard handoff for each incoming request (Min-Load-1) and allowing some BSs to connect to 2 RNCS (10 percent increase in number of links in our network) can provide significant decrease in rejection probabilities (33 percent decrease in our simulations).*



(a) (b)

Fig. 10. Rejection probabilities as one base station fails. (a) Single connected RAN to fully connected RAN. (b) Arc connectivity one to two.

6.4 Resilience

We have demonstrated how the connectivity and RNC selection algorithm impacts the performance of RAN. In this section, we evaluate the impact of connectivity and RNC selection algorithm on the resilience of RAN. This is done by simulating both BS and RNC failures and computing the worst case network rejection probability after the failure event. We assume a single point failure model, i.e., there is at most one failure at a time.

In general, there are three possible types of failure: link failure, base station failure and RNC failure. However, since the failure of a single link is in the worst case as serious as one base station failure when the base station it connects to is single connected, we will not present the evaluation of single link failure in this section.

First, we investigate the case for a single BS failure. Since the minimum connected balanced graph is uniform in its connectivity, we can simply randomly pick any BS to fail. Fig. 10a plots the rejection probability for RAN from single-connected to an arc connectivity of 10 after one base station fails. We observe that the rejection probability drops dramatically when RAN changes from single-connected to an arc connectivity of 2. The RAN of arc connectivity 2 is almost as resilient as the RAN where each base station connects to all RNCs (mesh-connectivity). We also observe that Min-Load-1 is superior to Min-Load and slightly worse than Min-Load-2 and optimal. The difference between Min-Load-2 and Optimal is minimal.

Next, we investigate the impact of one base station failure to the connectivity between single-connected and arc connectivity 2 graphs in Fig. 10b. In picking the BS to fail, we select the BS with the highest connectivity so that the resulting rejection probability is the worst case rejection probability. Therefore, after the failure, the RAN may be partitioned. From the figure, we see that the rejection probability is reduced significantly as we add one link per RNC to a RAN of arc connectivity 1. Adding another link per RNC reduces the probability further, but not as significant as adding the first link per RNC. Adding links further does not help to reduce the rejection probability any more. In case of one base station failure, a RAN of arc connectivity 1 with two additional links per RNC appears to be as resilient as more connected RANs. In fact, using the result from Section 6.3, we can justify this observation. Recall that, for the Min-Load-1 algorithm, the minimum connectivity required to achieve good performance is a single-connected graph plus 10 links added in a balanced way. With a BS failure, a connectivity of a single-connected graph plus 20 links can always obtain this minimum configuration after one BS failure. Thus, *a single-connected graph with 20 additional links and the Min-Load-1 algorithm provides a good balance between cost and resiliency due to base station failures.* We next examine RNC failures.

Fig. 11a plots the rejection probability for RAN of arc connectivity from 1 to 10 as one RNC fails. Since the graph is uniform, a random RNC is chosen to fail. We observe from the figure that rejection probability drops dramatically from single-connected graph to arc connectivity 2. The rejection probability of a more connected RAN is similar to the RAN of arc connectivity 2. Therefore, RAN of arc connectivity 2 is much more resilient than the RAN of arc connectivity 1. On the other hand, adding more links to RAN of arc connectivity 2 does not improve the resilience significantly.

Again, in Fig. 11b, we look at the connectivities between a single-connected graph and a graph with arc connectivity 2. The x-axis is the number of links in RAN. Since the graph is uniform, a random RNC is chosen to fail. The result shows that there is a significant difference in terms of resilience between this range of connectivities. The rejection probability decreases rapidly when the first links are added, but the improvement tapers off after that. In this simulation, adding five links per RNC appears to be the turning point where the curve flattens in Fig. 11b. We have also evaluated the same connectivities at both higher load and lower load and have found that the turning points change with load. We found that the turning point moves toward the arc connectivity 2 when the load decreases. One can argue that a minimum connected balanced graph of arc connectivity 2 is the minimum connectivity required to maintain low rejection after an RNC failure since any graph with a lower connectivity will be partitioned (one or more base stations are not connected to any RNC) after an RNC failure. As a result, *in order to make RAN resilient to RNC failures at any load, arc connectivity 2 is required.*

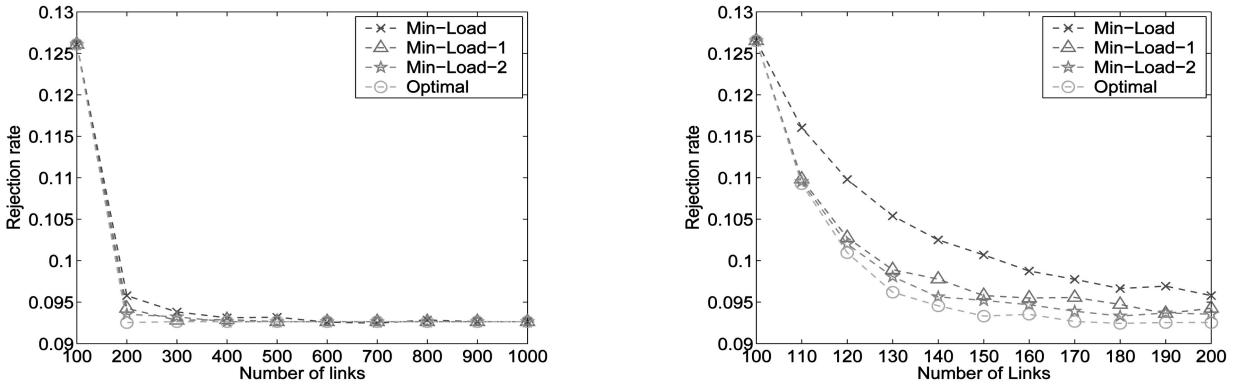


Fig. 11. Rejection probabilities as one RNC fails. (a) Single connected RAN to fully connected RAN. (b) Arc connectivity one to two.

6.5 Cost of Algorithms

As we have seen earlier, the Min-Load-k algorithms reduce the rejection rate as k increases. This is because larger values of k allow us to move existing calls further away from the base station toward an RNC with more available resources, such that the new call can be accommodated. However, the smaller rejection rate comes at the cost of a higher reassignment rate per new or handoff call. The reassignments may cause a disruption similar to that caused by a hard handoff (a slight "click" may be heard during the conversation). We define the number of reassignments per call (new or handoff) as the metric for the cost of the algorithm. Fig. 12 shows that this cost varies almost linearly from 0 for Min-Load to 0.43 for Min-Load-5. The optimal algorithm which can perform any number of reassignments will result in an even higher cost.

While the number of reassignments is still less than one per call, reassignments are undesirable from a user quality perspective. Note that we have not tried to optimize this value so far. We designed the algorithms to minimize the rejection rate with an upper bound on the number of reassignments per call. In the next section, we examine a simple reservation-based scheme that can reduce the reassignment cost of the Min-Load-k algorithms without causing any significant impact on rejection rates.

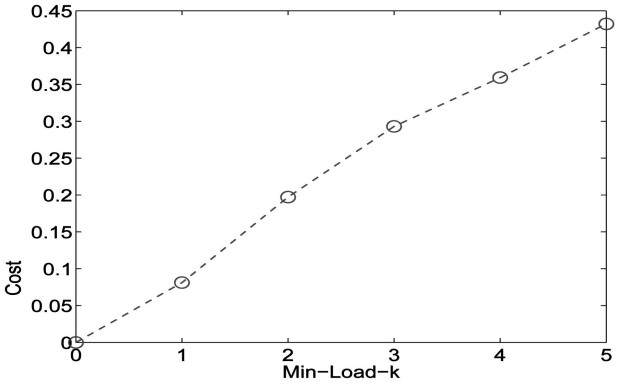


Fig. 12. Cost of algorithms.

7 Impact of Changes in Traffic and Network Model

In this section, we evaluate how changes to some of our assumptions may impact the observations made in Section 6.

7.1 The Impact of Heavy-Tailed Call Holding Time and Sojourn Time

So far, we have assumed that the call holding time as well as the mobile sojourn time at a base station are exponentially distributed. This makes the analytical model feasible. In this section, we consider a more general distribution for call holding and mobile sojourn times. Specifically, we replace the exponential distribution with a heavy-tailed distribution and repeat our simulations to see whether our observations still hold.

We use a Pareto-distributed call holding time and mobile sojourn time with a shape of 1.5 (corresponds to an infinite variance), but keep the same mean as in the exponential cases. We perform the same set of simulations and present our results in Fig. 13. We observe that the impact of heavy-tail distribution is hardly visible in the figures. From the values, we observe that the use of heavy-tailed distribution increases the reject ratio slightly for all assignment algorithms and all connectivities. However, the difference caused by the changed distribution is not significant enough to impact the observations that was made in Section 6.

7.2 The Impact of Soft Handoffs

So far, we have assumed that a user only communicates with one base station at a time. In this section, we investigate the impact of soft handoffs on both the connectivity and the assignment algorithms. In the soft handoff mode, a mobile has an active set of base stations that it communicates with. A base station that is in the active set of a mobile is also referred to as the mobile's leg. In the uplink direction, for each radio frame transmitted from the mobile, the RNC would pick the frame with the best quality among all the copies received. In the downlink direction, all base stations in the active set transmit the same frame synchronously, which is then soft-combined by the mobile.

Soft handoff not only improves the service quality, but also handles mobility better since the active set is constantly

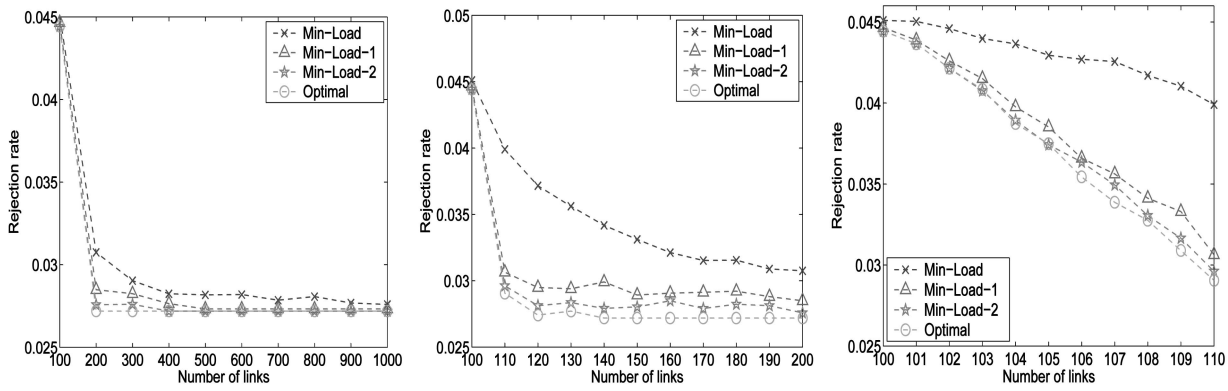


Fig. 13. Rejection probabilities for various connectivities: heavy-tailed call holding time. (a) Single connected to fully connected RAN. (b) Single connected to arc connectivity 1. (c) Single connected with a few additional links.

updated as a mobile user moves and the probability of an empty active set is very small. Among all base stations in the active set of a mobile, one base station is picked as the primary leg for managing the active set. The RNC capacity can be approximated by the total number of active calls (or primary legs) that can be supported.

Although there are obvious advantages for introducing soft handoffs, it creates some new problems to the assignment algorithms because all base stations in a mobile's active set have to be connected to the same RNC. Otherwise, the RNC would not be able to combine radio frames from the active set, leading to deterioration in voice quality. Therefore, when moving an existing call from one RNC to another, we would like every base station in the mobile's active set to be connected to the destination RNC. This would reduce the likelihood that an existing call can be moved for load balancing purposes—thus, soft handoff may reduce the advantage of the Min-Load-K algorithms and may require more reassignment in order to accept a new call.

For the purpose of evaluating the impact of soft handoffs, we extend our model in Section 3 to allow some of the mobile users to be in soft handoff mode. Recall that we modeled two types of user events, voice call events and mobility events. We need to make the following changes to our model in order to account for soft handoff:

- When a mobile is in the dormant mode, it does not communicate with any base station and, thus, its active set is empty. In order to model the mobility during dormancy, we assume the mobile is associated with a base station that would be the mobile's primary leg if it would have been active.
- When a successful call arrival event changes a user's state from dormant to active, we assume that the mobile only communicates with its primary leg at the beginning. The mobile is assigned to an RNC connected to its primary leg. The mobile then expands its active set by adding each neighboring base station of its primary leg that connects to the same RNC by a configurable probability. The probability is chosen such that the average size of the active set matches that observed in practice (1.5 average legs [5]).
- As a mobile moves, its primary leg may change to a new one depending on its new location. Again, the mobile is first assigned to an RNC connected to its primary leg. It then updates its active set by removing legs that are either not a neighbor of the new primary leg or not connected to the new RNC. The neighboring base stations that connect to the same RNC may be added to the active set such that the average size of active set is maintained.
- As a Min-Load-k algorithm moves an existing call from one RNC to another in order to accept a new call, it has to make sure that the new RNC is connected to all legs in the mobile's active set. This necessarily reduces the potential number of calls that can be moved.

In a 3G service provider's network, the base stations are usually connected to geographically close RNCs to save on the wiring cost and reduce delays. For instance, it is hard to imagine a base station in the east coast of the United States connecting to an RNC in the west coast of the United States. As a result, base stations in the same active set of a mobile tend to connect to a common set of RNCs. This increases the chance that the base stations in the active set of an existing call also share other RNCs than the one the call is currently assigned to. Thus, we can expect that the above locality property would overcome some disadvantages to the Min-Load-K algorithm introduced by soft handoff. We account for the locality property by dividing the two dimension base station layout plane evenly into zones and assign one RNC to each zone. A base station always connects to the RNC in the same zone and also connects to RNCs in the neighboring zone when additional connectivity is available.

We repeat our previous simulations but with the additional constraints imposed by mobiles in soft handoff mode. In Fig. 14, we plot the rejection rates with soft handoff. If we compare Fig. 14 to Fig. 9, we can see that soft handoff does decrease the improvement of Min-Load-K over Min-Load because it becomes more difficult to move existing calls to make room for new calls. Thus, more calls are dropped. However, as more connectivity becomes

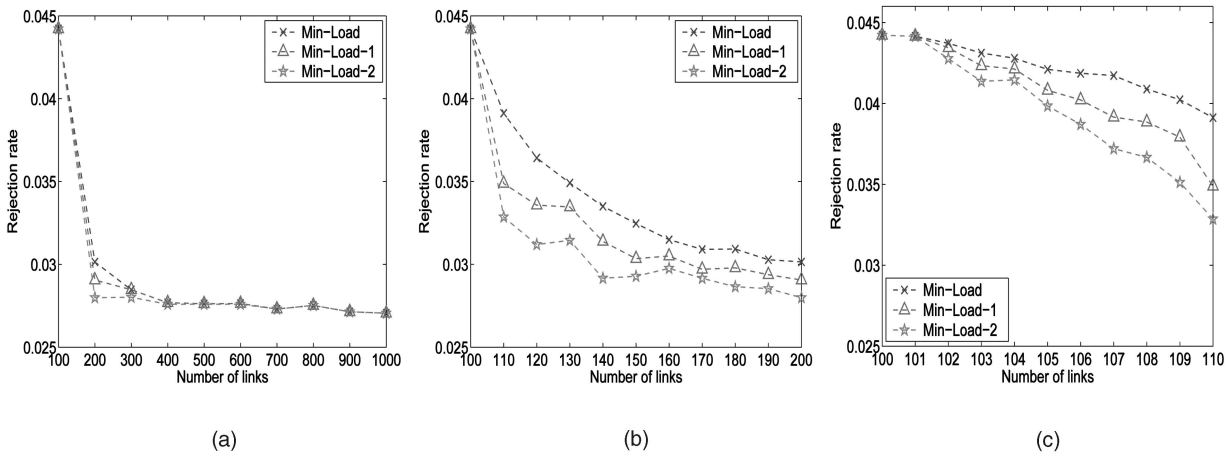


Fig. 14. Rejection probabilities for various connectivities: impact of soft handoff. (a) Single connected to fully connected RAN. (b) Single connected to arc connectivity 1. (c) Single connected with a few additional links.

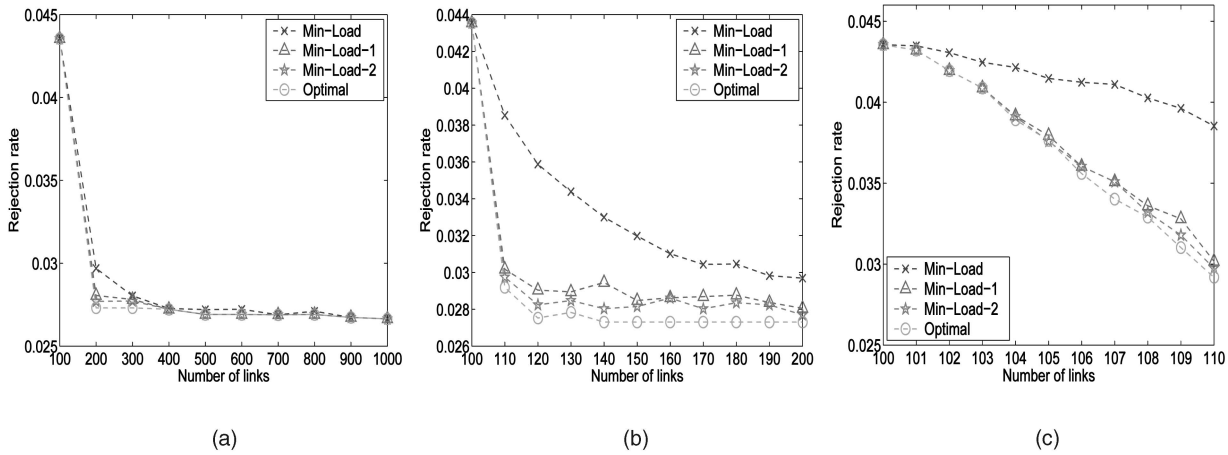


Fig. 15. Rejection probabilities for various connectivities: finite base station capacity. (a) Single connected to fully connected RAN. (b) Single connected to arc connectivity 1. (c) Single connected with a few additional links.

available, the impact of soft handoff diminishes for two reasons. First, the Min-Load algorithm can now do a better job. Second, the availability of movable existing calls also increases due to better connectivity.

7.3 Finite Base Station Capacity

So far, we have assumed that base stations have infinite capacities and, thus, never drop calls. However, in a realistic scenario, a base station may also drop calls when its radio resources are exhausted. For a given 3G network, the amount of radio resource available on the base station is limited by the wireless spectrum available. From the point of view of IP-RAN design, base station capacity is simply a given set of system configuration parameters. In this section, we assign a limited capacity to the base station such that the base station has to drop calls when the number of active calls associated with a base station exceeds its capacity. More specifically, we repeat the previous simulations but assign a capacity of 120 calls to each base station. This is about two times the average number of active calls on each base station station. We plot the rejection rates in Fig. 15. Note that the rejection rates only account for call drops due to RNC capacity since we are interested in connectivity issues and not base station planning. As we can see from the graph, the total rejection rates are lower for all algorithms and all connectivities. In addition, the performance difference between Min-Load and Min-Load-K narrows. This is because some calls are rejected by base stations and the total number of calls seen by the RAN is lower. For a lower load, the chance that the RAN can accept a call without moving an existing calls is higher. Thus, the difference between Min-Load and Min-Load-K is smaller. Nevertheless, the key observations made concerning the Min-Load-K algorithm remain the same. We find similar observations as we repeat simulations for RNC and base station failure with finite base station capacity.

8 Reducing Hard Handoffs through Reservation

In this section, we introduce an approach to reduce the cost of the algorithms through a simple reservation scheme. We then evaluate its impact on the cost and rejection rate metrics.

In Section 6, the number of hard handoffs was relatively high because the algorithms were designed to minimize rejection rate regardless of the reassignment cost. In high load conditions, the number of reassignments may be

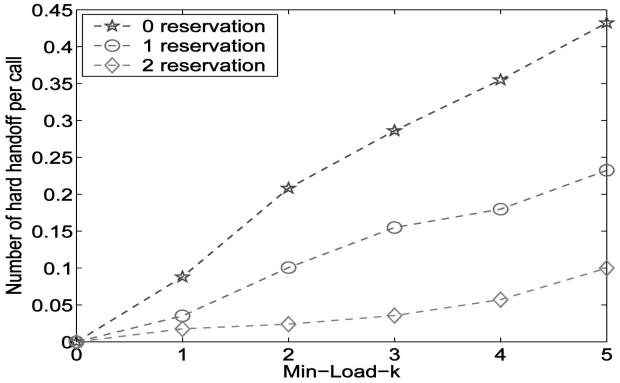


Fig. 16. Impact of reservation on cost.

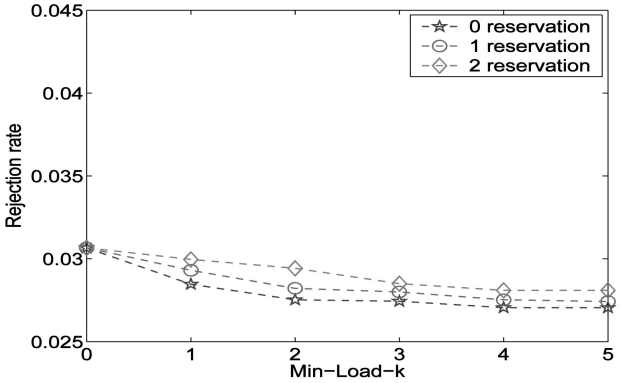


Fig. 17. Impact of reservation on rejection rate.

unnecessarily high due to a "ping-pong" effect, where calls are moved back and forth from one RNC to another. One way to reduce the number of hard handoffs is to reduce the possibility of ping-pongs by not reassigning calls when the target RNCs are almost full. This can be achieved by reserving a small amount of RNC resources for directly connected BSs only.

Note that this is similar to the guard channel concept [19], [20] used in cellular networks for reducing the call dropping probability. In the guard channel scheme, when the utilization is high, a few channels are reserved only for handoff calls. This results in a higher priority for handoff calls as compared to new calls, thereby reducing the call dropping probability. In our reservation scheme, we reserve a few RNC resources for directly connected BSs, thereby reducing the priority for calls that need to be reassigned.

Fig. 16 shows how reassignment cost decreases as more resources are reserved for Min-Load-k with different ks. The $0/1/2$ reservation in the figure refers to no reservation, reserving resource for one call, and reserving resources for two calls respectively. Fig. 17 plots the impact of reserving resources on the rejection rate. Fig. 16 shows that the reassignment cost can be decreased by 50 percent by just reserving resource for one call and by 80 percent when we reserve resources for two calls per RNC. The impact of the reservation on the rejection rates is insignificant for both cases, with less than 5 percent increase even when two calls worth of resources are reserved. This suggests that *a simple reservation scheme that reserves resources for a few calls per RNC reduces the cost of call reassignment dramatically while incurring an insignificant increase in the rejection rate*.

9 Heterogeneous Networks

In the approach presented so far, we have assumed that the network is homogeneous. However, in reality, heterogeneity can arise in a number of ways, including differences in load, BS/RNC capacity, and link costs. In this section, we study the effectiveness of the Min-Load-k algorithm in handling heterogeneous load and then briefly discuss the issues of heterogeneous BS/RNC capacity and link costs.

9.1 Heterogeneous Load

So far, we have demonstrated the advantage of the Min-Load-k algorithm when the load among different base states are the same. In this section, we investigate how the Min-Load-k algorithms perform when the load is unbalanced among the base stations.

We maintain the total load unchanged as compared to the previous section but vary the load distribution to introduce heterogeneity. The unbalanced load is constructed in the following way. We first order all base stations, say, by node identifier. We then assign an initial load d_0 to the first base station and assign Δ more load to each subsequent base station such that the highest load (for the last base station) is L times the lowest load d_0 and the total load is the same as the uniform case. Therefore, for a total load, D, and N base stations, we have

$$d_i + \Delta = d_{i+1}, i = 0, \ldots, N-1,$$

$$\sum_{i=0}^{N-1} d_i = D,$$

$$L \times d_0 = d_{N-1},$$

where d_i is the load at base station i.

The value of d_i can be solved from the above as

$$d_i = \frac{2D}{(L+1)N} + \frac{2(L-1)D}{(N-1)(L+1)N} i.$$

Since the load becomes more unbalanced when L increases, we refer to L as the unbalanced factor. We evaluate the Min-Load-k algorithm for different unbalanced factors and different RANs. We pick two representative topologies: the two connected RAN and the one connected RAN with 10 additional links. The first one represents a well connected RAN where each base station connects to two RNCs and the second one represents a poorly connected RAN, where only 10 out 100 base stations connect to two RNCs.

Fig. 18 plots the rejection rates when using Min-Load-k algorithms on a RAN of arc connectivity two. We simulate cases with balanced ($L = 0$) and unbalanced loads ($L = 2$ and $L = 4$). For each load, six Min-Load-k algorithms are shown where k varies from 0[1] to 5. From Fig. 18, it is clear that the rejection rate increases as the unbalanced factor increases. The rejection rate also decreases as k of Min-Load-k increases. Min-Load-1 achieves most of the improvement

1. Min-Load-0 is just the Min-Load.

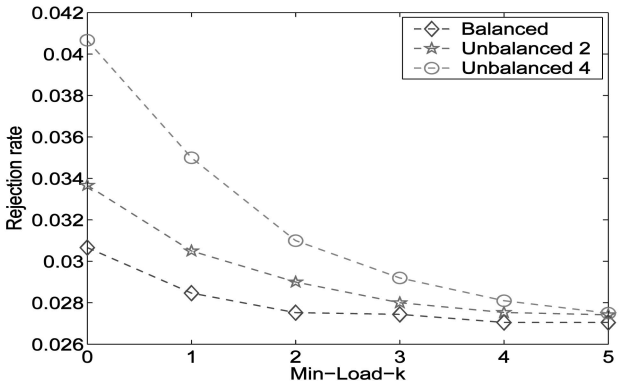


Fig. 18. Unbalanced load on two connected RAN.

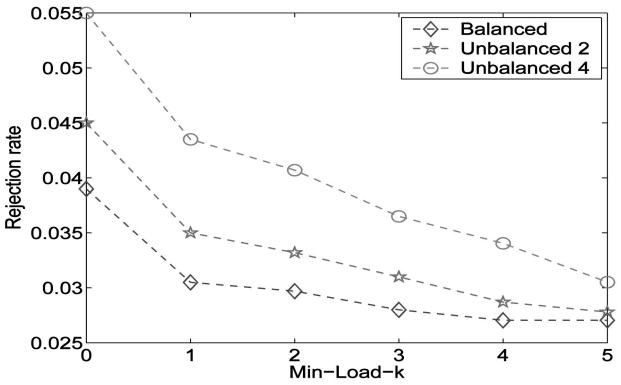


Fig. 19. Unbalanced load on one connected RAN.

over Min-Load as compared to these higher value of ks. Since the diameter of the reassignment graph is 5, Min-Load-5 always achieves closest to the minimum rejection rate under different types of loads and is the most robust with respect to unbalanced load. In general, as the load becomes more unbalanced, more reassignments through the use of a large k in the Min-Load-k algorithm are necessary to achieve similar rejection rates as the balanced load case.

We also evaluate a RAN of arc connectivity one with 10 additional links. Fig. 19 plots the rejection rates when using Min-Load-k algorithms, with balanced and unbalanced load ($L = 2, 4$). The relative performance of the Min-Load-k algorithms are similar to the previous case (RAN topology with 200 links). Rejection rates increase as the load becomes more unbalanced. However, the absolute values of the rejection rates of unbalanced RAN are higher as compared to the previous case since a RAN with lower connectivity restricts the reassignment ability of the algorithms. In addition, even Min-Load-5 cannot achieve the minimum rejection rates since the diameter of the RAN is now larger than 5. Nevertheless, Min-Load-1 still achieves the largest incremental improvement as k increases.

In conclusion, while the unbalanced load makes reassignment harder and increases rejection rates in most case, the Min-Load-k algorithms still perform significantly better than Min-Load and Min-Load-1 has the largest incremental performance improvement.

9.2 Heterogeneous RNC Capacity

One approach to solve a network with heterogeneous RNC capacities is to map it to a constrained homogeneous network. Then, we can use a similar approach as presented in this paper to tackle the connectivity problem. One way to map a heterogeneous network into a homogeneous network is through the following strategy. The heterogeneous RNCs are split into homogeneous logical RNCs with capacities/loads equal to the highest common denominator of all the RNCs. In order to mimic the physical locality of the RNCs, whenever a logical BS is connected to a logical RNC in the connectivity model, additional links are added between all the corresponding logical BSs of the original heterogeneous BS to all the corresponding logical RNCs of the original heterogeneous RNCs. We thus have a logical homogeneous network. However, in the presence of these "irregularities" in the connectivity graph, enumeration of the balanced graphs is a much harder problem and it is not clear if the state space can be reduced significantly as in the case for homogeneous network. Furthermore, this transformation is just one possible way of analyzing connectivity in heterogeneous networks and more work is needed to explore ways of constructing and enumerating other forms of balanced graphs that are better suited for heterogeneous networks with different capacities at the RNC.

9.3 Heterogeneous Link Costs

Heterogeneous link costs add a new dimension to the problem. Besides having different communication cost, addition of some links may not be allowed because of QoS and/or geographical constraints (e.g., delay is too large). In addition, the cost function is no longer just call blocking and dropping rates but also includes total communication cost. This results in significant additional complexity to the model. We are exploring ways to tackle this cost as part of future work.

10 Conclusion

In this paper, we addressed the question of how best to connect base stations to the Radio Network Controllers (RNC) in an IP-based RAN. Furthermore, given a connection configuration, we also developed RNC selection algorithms that assign an incoming call to an RNC. We found that the Min-Load-1 algorithm, that allows at most one hard handoff in order to accommodate a new request, delivers performance close to the optimal algorithm. We also found that allowing a few base stations to connect to two RNCs (10 percent increase in number of links in our network) can provide significant decrease in rejection probabilities (33 percent decrease in our simulations). We further found that allowing base stations to connect to two RNCs results in similar resiliency to RNC failures as having full-mesh connectivity between base stations and RNCs. Finally, for heterogeneous networks, we show that the Min-Load-k algorithm (with at most k hard handoffs per call) is effective in handling load imbalances. *These results provide strong motivation for deploying IP-based RAN as they suggest that enhancing current point-to-point RAN with few additional links and allowing a few hard handoffs to accommodate incoming calls can result in significant gains in performance and resiliency.*

REFERENCES

[1] TIA/EIA/cdma2000, *Mobile Station—Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems,* Washington: Telecomm. Industry Assoc., 1999.

[2] 3G Partnership Project, "Release 99," 2004.

[3] CDMA Development Group, http://www.cdg.org, 2004.

[4] G. Heijenk, G. Karagiannis, V. Rexhepi, and L. Westberg, "Diffserv Resource Management in IP-Based Radio Access Networks," *Proc. Fourth Int'l Symp. Wireless Personal Multimedia Comm. (WPMC '01),* Sept. 2001.

[5] S. Kasera, R. Ramjee, S. Thuel, and X. Wang, "Congestion Control Policies for IP-Based CDMA Radio Access Networks," *Proc. Infocom,* Apr. 2003.

[6] H. el Allali and G. Heijenk, "Resource Management in IP-Based Radio Access Networks," *Proc. CTIT Workshop Mobile Comm.,* Feb. 2001.

[7] D.L. Eager, E.D. Lazowska, and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," *IEEE Trans. Software Eng.,* vol. 12, no. 5, 1986.

[8] V. Harinarayan and L. Kleinrock, "Load Sharing in Limited Access Distributed Systems," *Proc. ACM Sigmetrics Conf. Measurement and Modeling of Computer Systems,* pp. 21-30, 1991.

[9] Y. Wang and R. Morris, "Load Sharing in Distributed Systems," *IEEE Trans. Computers,* vol. 34, no. 3, pp. 204-216, Mar. 1984.

[10] M. Litzkow and M. Solomon, "Supporting Checkpointing and Process Migration Outside the Unix Kernel," *Proc. Usenix Winter Conf.,* 1992.

[11] "Cisco IP Radio Access Network Transport Solution," http://www.cisco.com/en/us/netsol/ns341/ns396/ns177/ns329/, 2002.

[12] "Nokia Launches Its IP Radio Access Network Concept IP-RAN," http://press.nokia.com/pr/200002/, Feb. 2002.

[13] P. Agrawal, T. Zhang, C.J. Sreenan, J.-Y. Chen, and R. Ramaswani, "Guest Editorial: All-IP Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 4, pp. 613-616, 2004.

[14] 3GPP TSG RAN, "IP Transport in Utran Work Task Technical Report," tr-25.933 v0.2.0, 2000.

[15] K. Venken, I.G. Vinagre, and J. De Vriendt, "Analysis of the Evolution to an IP-Based UMTS Terrestrial Radio Access Network," *IEEE Wireless Comm.,* vol. 10, no. 5, pp. 46-53, Oct. 2003.

[16] S.-C. Lo, G. Lee, W.-T. Chen, and J.-C. Liu, "Architecture for Mobility and QOS Support in All-IP Wireless Networks," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 4, pp. 691-705, 2004.

[17] A.-C. Pang, Y.-B. Lin, H.-M. Tsai, and P. Agrawal, "Serving Radio Network Controller Relocation for UMTS All-IP Network," *IEEE J. Selected Areas in Comm.,* vol. 22, no. 4, pp. 617-629, 2004.

[18] C. Tao and S. Hamalainen, "Handover in IP Ran," *Proc. Int'l Conf. Comm. Technology (ICCT '03),* vol. 2, no. 9, pp. 812-815, Apr. 2003.

[19] S.-H. Oh and D.-W. Tcha, "Prioritized Channel Assignment in a Cellular Radio Network," *IEEE Trans. Comm.,* vol. 40, no. 7, pp. 1259-1269, July 1992.

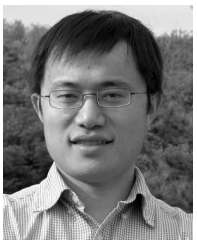
[20] C.H. Yoon and K. Un, "Performance of Personal Portable Radio Telephone Systems with and without Guard Channels," *IEEE J. Selected Areas in Comm.,* vol. 11, no. 6, pp. 911-917, Aug. 1993.

[21] R.A. Ahuja, T.L. Magnanti, and J.B. Orl, *Network Flows: Theory, Algorithms and Application.* Prentice Hall, 1993.

[22] F.P. Kelly, *Reversibility and Stochastic Networks,* chapter 2: Migration Processes, Wiley, 1979.

[23] S. Ross, *Simulation.* Harcourt/Academic Press, 1996.

[24] M. Alanyali and B. Hajek, "On Simple Algorithms for Dynamic Load Balancing," *Proc. INFOCOM,* pp. 230-238, 1995.



Tian Bu received the PhD degree in computer science from the University of Massachusetts, Amherst, in 2002. He has been at Bell Labs, Lucent Technologies, since 2002, where he is a member of the technical staff at the networking research lab. His current research includes network security, network modeling, and performance evaluation.

Mun Choon Chan received the BS degree from Purdue University, West Lafayette, IN, in 1990 and the MS and PhD degrees from Columbia University, NY, in 1993 and 1997, respectively, all in electrical engineering. From 1991 to 1997, he was a member of the COMET Research Group, working on ATM control and management. From 1997 to 2003, he was a member of the technical staff at the Networking Research Lab, Bell Laboratories, Lucent Technologies, Holmdel, NJ. Since January 2004, he has been an assistant professor in the Department of Computer Science, National University of Singapore. His current research includes wireless data and sensor networking.

Ram Ramjee received the BTech degree in computer science and engineering from the Indian Institute of Technology, Madras, and the MS and PhD in computer science from the University of Massachusetts, Amherst. He has been at Bell Labs, Lucent Technologies, since 1996, where he is currently leading the next generation networks research department with a group of researchers examining architecture, protocol, and performance issues in next generation wired and wireless networks. He is also an adjunct faculty in the Electrical Engineering Department at Columbia University, where he teaches graduate courses in wireless networks. He served as the program committee cochair of IEEE ICNP 2004 and will serve as the general cochair of WICON 2006. Dr. Ramjee serves as an area editor of the *ACM Mobile Computing and Communications Review*, an associate editor of *IEEE Transactions on Mobile Computing*, and a technical editor of the *IEEE Wireless Communications Magazine*. He has published more than 40 papers and holds 12 US patents.