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

Tian Bu Mun Choon Chan Ram Ramjee
Bell Laboratories, Lucent Technologies
{tbu,munchoon,ramjee}@bell-labs.com

***Abstract*— IP-based Radio Access Networks (RAN) are expected to be the next generation access networks in UMTS and CDMA networks. There are several benefits of IP-based RAN including lower costs, flexibility of merging wired and wireless networks, and network scalability and reliability. While Quality of Service issues in IP-based RAN have been addressed by a number of researchers, 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 any research literature. Furthermore, given a connection configuration, an RNC selection algorithm that assigns an incoming call to an RNC is also necessary. This paper examines Radio Access Network (RAN) connectivity and its impact on the performance and resiliency of the wireless network using different RNC selection algorithms. The proposed Min-Load-1 algorithm, which allows at most one hard handoff in order to accommodate each incoming request, delivers performance close to the standard Min-Load algorithm using a RAN of much higher connectivity and is close to the optimal algorithm using the same RAN. We also find that using Min-Load-1 algorithm and allowing the base stations to connect to two RNCs result in resiliency to RNC failures that is similar to having full-mesh connectivity between base stations and RNCs.**

I. 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. 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. 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 [3] (IP-based RAN). The current wireless access network architecture and an architecture based on IP RANs is shown in Figure 1.

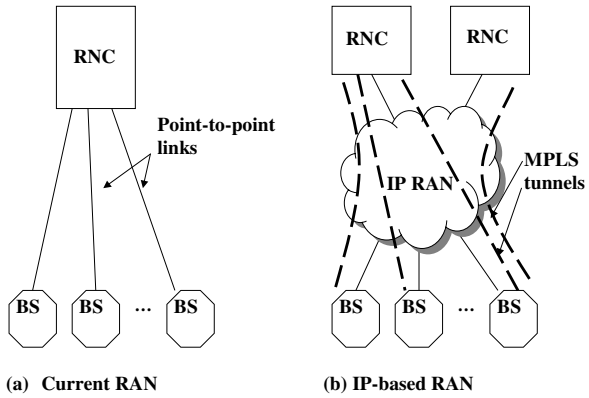


Fig. 1. Wireless access network architectures

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

- *Scalability:* RNC capacity could 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.
- *Flexibility:* 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).

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], [3]. As shown in Figure 1, use of IP-based MPLS tunnels between base 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. Given the QoS constraints, enabling full mesh connectivity between base stations and RNCs can be expensive. On the other hand, enabling full mesh connectivity may not be necessary and may have little incremental impact on performance. 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 ($\approx 2^{1000}$).

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 dropping of handoff calls and 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 [6], [7], [8] 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 [9] which implement load balancing in distributed systems are general purpose mechanisms, they result in 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 connectivity. The Min-Load-k algorithm assigns calls to RNCs such that 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.

We compare the performance through extensive simulations of Min-Load-k algorithm with an on-line 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 Min-Load-1 algorithm and allowing the base stations to connect to two RNCs result in resiliency to RNC failures that is similar to having full-mesh connectivity between base stations and RNCs.*

The rest of the paper is structured as follows. In Section II, we present an overview of the problem. In Section III, we present our approach to making the connectivity problem between base stations and RNCs tractable by systematically evaluating different connection topologies. In Section IV, we present several algorithms for RNC selection and an analytical model for the optimal algorithm. In Section V, we evaluate the impact of connectivity between base stations and RNCs on the overall performance and the resiliency of the network. In Section VI, we discuss issues with modeling heterogeneous networks. Finally in Section VII, we present our conclusions.

II. Problem Setting

As shown in Figure 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 [4], 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, 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: a) current cellular networks are predominantly used for voice transmission; and b) voice has tighter 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 cannot occur at the same time. These are common assumptions and are used in [10], [11]. For call event, we are interested in *call blocking rate*, the average rate of blocking a new call. For 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. Therefore, 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 rate assumed is 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.

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 should the RAN 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 for a given RAN. These two issues are inter-related as the choice of algorithms is a function of 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 need 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 III and the RNC selection algorithms are presented in IV.

III. 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 connectivities 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. This presentation here follows [12]. 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 focus on the homogeneous network case in the remainder of this paper. The issues in modeling heterogeneous network are discussed in more detail in Section VI.

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 graph. Figures 2 (a) and 2 (b) show two ways of constructing a minimum connected balanced graph with 4 RNCs, 8 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 a 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 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 Figures 2(a) and 2(b) as examples, the corresponding RNC accessibility graphs are shown in Figures 3(a) and 3(b) respectively. In Figure 3(a), the RNC accessibility graph for BS 0 is shown. Due to the regular structure of the network in Figure 2(a), all BSs have similar RNC accessibility graphs. In Figure 3(a), there is 1 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 2 paths to RNC 2 (through BS 2 and 3).

In Figure 3(b), the RNC accessibility graph for BS 0 has 1 path to all RNCs and the graph for BS 3 has 1 path each to RNC 1 and 3 and 3 paths each to RNC 0 and 2. Obviously,

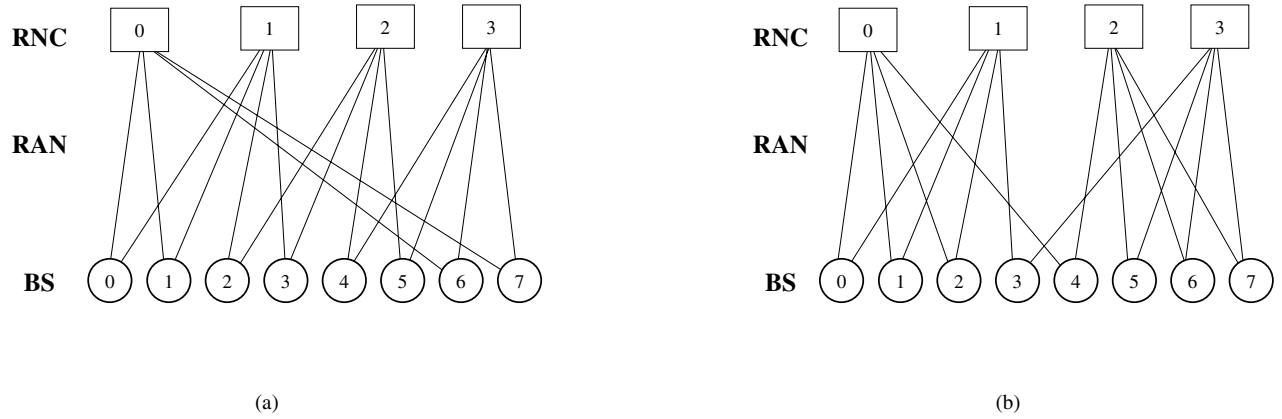
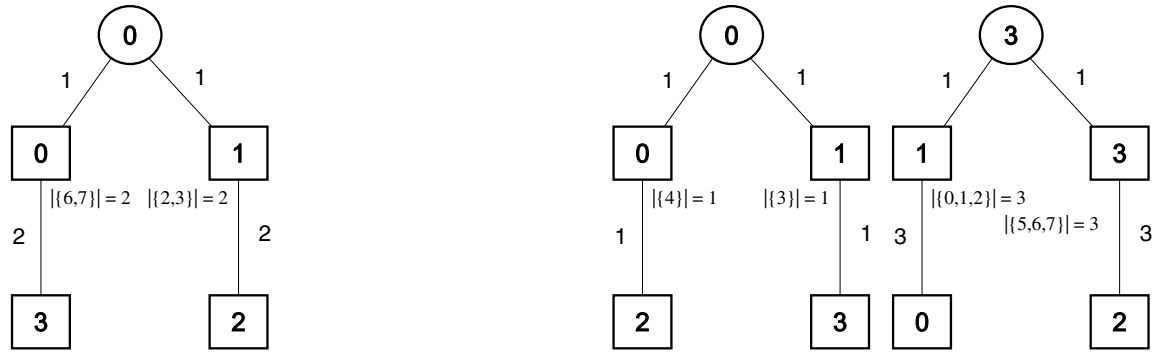


Fig. 2. RAN with arc connectivity 2



(a) Graph for Figure 2(a)

(b) Graph for Figure 2(b)

Fig. 3. RNC accessibility graph of BS 0 and 3

the graph in Figure 3(a) 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 (Figure 3(a)) 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 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 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 graph, the behavior of different instantiations of these balanced graph with the same L is the same.

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

TABLE I

NOTATIONS FOR ALGORITHMS

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.

IV. 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 a 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 I.

A. 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 RNC 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 find a feasible solution with the new set of load configuration. The feasible solution is solved by formulating it as a maxflow problem as illustrated in Figure 4. A set of BS, each with b_i active users, and a set of RNCs, each with capacity r_j is shown on the left in Figure 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 Figure 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 reader can refer to [12]. Assuming the maximum flow value 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 $\sum_j r_j < \sum_i b_i$.

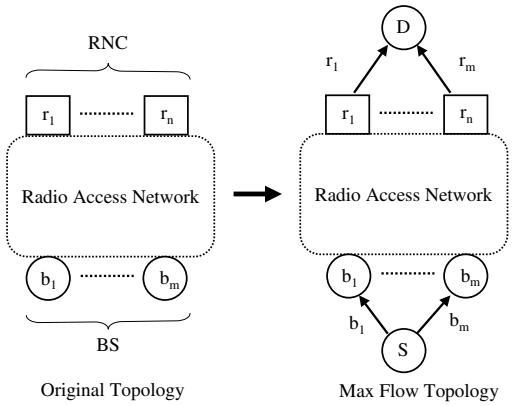


Fig. 4. Transforming optimal algorithm into a Max Flow problem.

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 hand-offs. 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 handoff such that a call request can be satisfied. An example of how a Min-Load-1 algorithm works is shown in Figure 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,

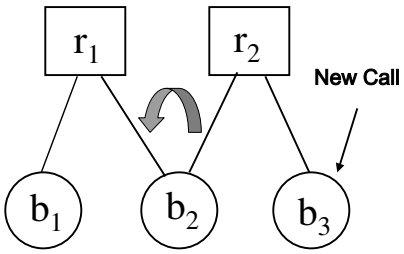


Fig. 5. Reassign an existing call from r_2 to r_1 to accept a new request.

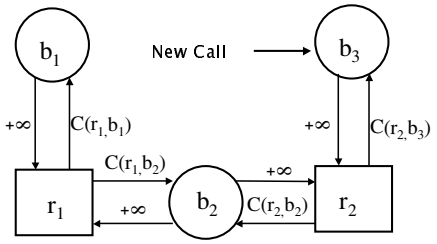


Fig. 7. Reassignment graph

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
  od
}
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
    fi
  fi
}

Fig. 6. Min-load-k Functions

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 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 pseudo code of Min-load-k algorithm is shown in Figure 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 Figure 5 can be converted into the reassignment graph in Figure 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 non zero available capacity through a non-blocked 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 non-zero available capacity at the same 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.

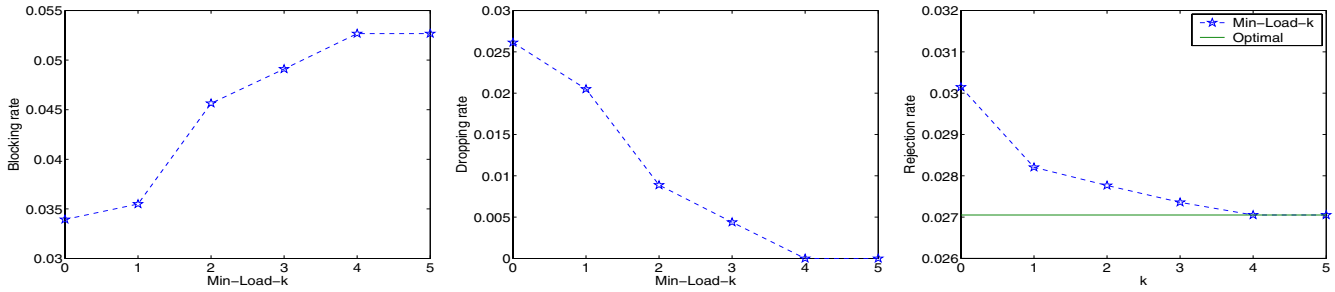
In practice, the Min-Load-k algorithm runs as a distributed algorithm that is initiated at the base stations with help from the RNCs. It is important to keep k as small as possible since a large k incurs more hard handoff and larger call setup time. We are interested in exploring how large k needs to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

B. Analytical Model

In this section we present an analytical model for the optimal algorithm. If we assume that 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 [13]. 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 $\{a_1, d_1, ..., a_M, d_M\}$. The feasibility of a set of $a_1, a_2, ..., a_M$ depends on not only the RNC capacities but also the RAN connectivity, which can be checked by using the max flow graph, e.g., Figure 4. In addition, $\sum a_i + \sum d_i = |U|$, the total number of users in the system. Let the feasible set of $\{a_1, d_1, ..., a_M, d_M\}$ 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



(a) Blocking probability (b) Dropping probability (c) Rejection probability

Fig. 8. Performance of different algorithms using a minimum connected graph with arc connectivity 2

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 $x = \{x_1, x_2, ..., x_{2M}\}$, 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 $\alpha_1, \alpha_2, ...\alpha_{2M}$ be the unique collection of positive numbers, summing to unity, that satisfy

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

For a given set of mobility rate, arrival rate and call holding times, the system is stable only if there are solutions to the set of simultaneous equations given in Equation 1. In the case where the RNC capacity is infinite, α_j is the equilibrium probability that a user is in state j.

Due to space limitations, we will omit the proof that the system has a product form solution except to mention that the system satisfies Theorem 1.7 and Corollary 1.10 in [13]. Using theorem 2.3 in [13], the equilibrium distribution for the system is given as

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

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

Using equation 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 Equation 2 due to state space explosion. Instead, the Monte Carlo approach [14] can be used.

Since there is no analytical model for Min-Load and Min-Load-k algorithms, we compare the algorithms with simulations.

V. 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 V-B, the performance of the various algorithms are compared. In Section V-C, we perform detailed evaluation of the impact of connectivity on the various algorithms. In Section V-D, the resiliency of the various connectivity graphs in the presence of a single link, BS and RNC failure are evaluated. Finally, in Section V-E, the cost of the various algorithms, measured in number of hard handoffs performed, is presented.

A. Simulation setup

The Radio Access Network simulated has 10 RNCs and 100 based stations. 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 base stations which are neighbors to its currently associated base station.

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 [10], [11]. Thus, an algorithm may reduce dropping probability and increase 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 [10], [11] can help control the relative preference between blocking and dropping probabilities, but this issue is outside the scope of this paper.

B. 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 2 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. Figure 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 optimal algorithm is also plotted in the Figure 8(c) 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 anymore as explained earlier.

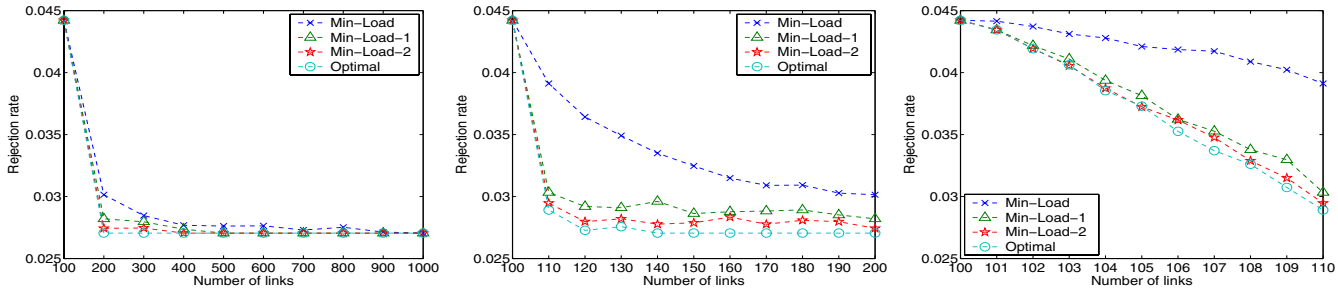
C. Connectivity

In this section we evaluate how the connectivity of RAN impacts 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 Figure 9(a). 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 in Figure 9(b). 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 1. This is shown in Figure 9(c). The RNC selection algorithms, Optimal, Min-Load, Min-Load-1, and Min-Load-2 are evaluated for all the connectivities considered.

From Figure 9(a), 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 algorithm perform the same on the single-connected graph because each base station only connects to one RNC and there is no alternative RNC to select. The large performance improvement from single-connected graph to graph with arc connectivity two motivates the next graph which zooms into the set of graphs with connectivities between the single-connected and arc connectivity 2 cases.

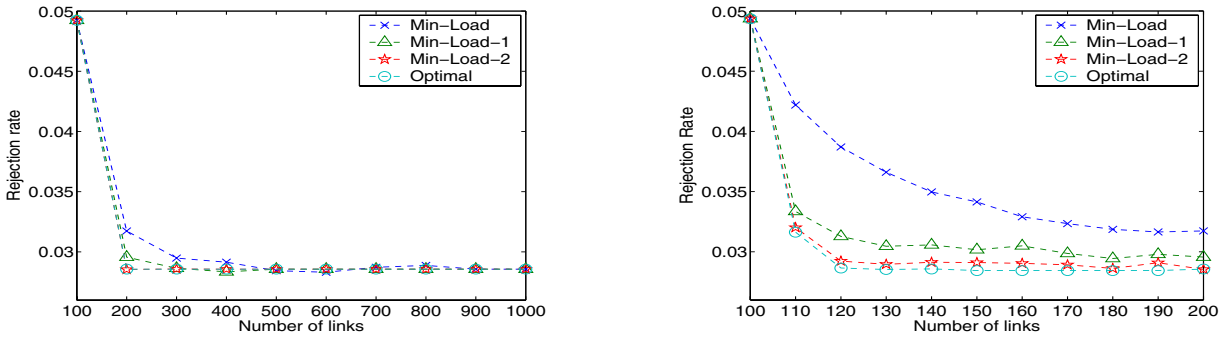
Figure 9(b) plots the rejection probability of RANs as we add links in increments of 10 to a single-connected graph. The x-axis is the number of links in RAN. In constructing the balanced graph using the methodology outlined in Section III, each time we add 10 links, we select base stations with the lowest connectivity and each link is connected to a different RNC. Figure 9(b) 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 performance gain (rejection probability reduction) occurs during the addition of the first ten links to the single-connected graph. This can be explained by recalling in the reassignment graph (Figure 7) that we constructed in Section IV. Reassignment can be visualized as visiting the directed graph in a breadth-first manner until an RNC with non-zero 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 non-zero 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 2 points in Figure 9(b), we now look further to see how the rejection probability changes as we add one link at a time to a single-connected graph. Figure 9(c) plots the rejection probability as we add up to 10 links. Observe that Figure 9(c) is different from Figure 9(a) and 9(b) in that there is no dramatic decrease in rejection. The decrease in rejection probability is almost linear showing that the performance gain



(a) Single connected to fully connected RAN (b) Single connected to arc connectivity 1 (c) Single connected with a few additional links

Fig. 9. Rejection probabilities for various connectivities



(a) Single connected RAN to fully connected RAN (b) Arc connectivity one to two

Fig. 10. Rejection probabilities as one base station fails

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 saving of 45% 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 base stations to connect to 2 RNCS (10% increase in number of links in our network) can provide significant decrease in rejection probabilities (33% decrease in our simulations).*

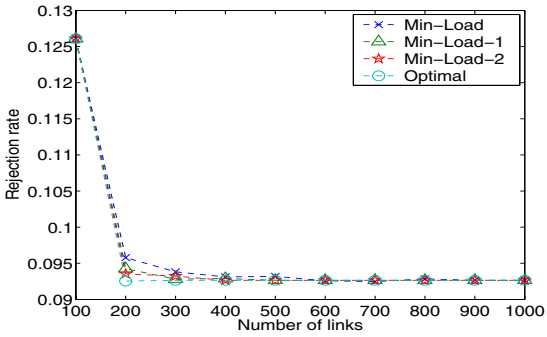
D. Resilience

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 base station 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.

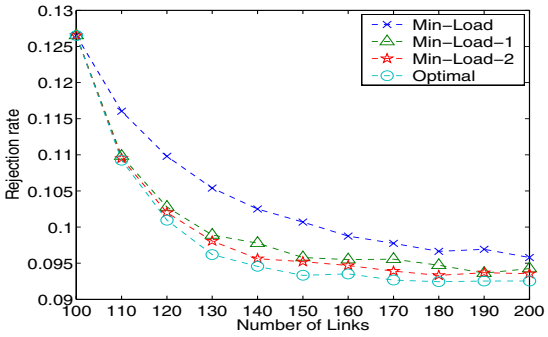
In general, there are three possible types of failure: link failure, base station failure and RNC failure. However, since the failure of a single link is in the worst case as serious as one base station failure when the base station it connects to is single connected, we will not present the evaluation 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. Figure 10(a) plots the rejection probability for RAN from single-connected to arc connectivity of ten after one base station fails. We observe that rejection probability drops dramatically when RAN changes from single-connected to 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 Figure 10(b). 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



(a) Single connected RAN to fully connected RAN



(b) Arc connectivity one to two

Fig. 11. Rejection probabilities as one RNC fails

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 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 result from Section V-C, 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 1 BS failure. Thus, *a single-connected graph with 20 additional links and the Min-Load-1 algorithm provide a good balance between cost and resiliency due to base station failures.* We next examine RNC failures.

Figure 11(a) plots the rejection probability for RAN of arc connectivity from one to ten 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 Figure 11(b), we look at the connectivities between 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 5 links per RNC appears to be the turning point where the curve flattens in Figure 11(b). 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 the turning point moves towards the arc connectivity 2 when

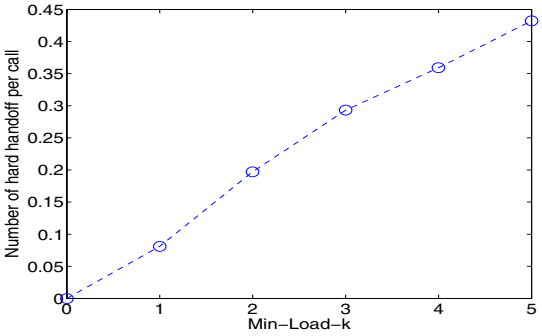


Fig. 12. Cost of algorithms

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.*

E. Cost of Algorithms

In this section, the cost of the various algorithms is evaluated. The cost is measured as the number of reassignments per new call arrival and handoff call. Figure 12 shows that the 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 have even higher cost. The number of reassignments is relatively high because the algorithms are designed to minimize rejection rate independent of reassignment cost. In high load conditions, there is a "ping-pong" effect where calls are moved back and forth from one RNC to another. One way to reduce the cost is to not reassign calls when the target RNCs are almost full by reserving a small amount of RNC resources for directly connected BSs. This is similar to the guard channel concept. From simulations, we observed that by reserving a capacity for 2 (out of 500) calls for directly connected BSs, the reassignment cost is reduced by 50% with almost no impact on rejection rate.

VI. Discussion

In the approach presented so far, we have assumed that the network is homogeneous. Therefore, all the BSs have the same average load, all the RNCs have the same capacities and all the link costs are assumed to be the same.

One approach to solve a heterogeneous BS and RNC problem is to map it to a constrained homogeneous network using the following strategy. The heterogeneous RNCs/BSs are split into homogeneous logical RNCs/BSs with capacities/loads equal to the highest common denominator of all the RNCs/BSs. In order to mimic the physical locality of the RNCs/BSs, 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. 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.

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. We are exploring these issues as part of future work.

VII. 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 each incoming request, delivers performance close to the optimal algorithm. We also found that allowing few base stations to connect to 2 RNCs (10% increase in number of links in our network) can provide significant decrease in rejection probabilities (33% decrease in our simulations). We further found that allowing base stations to connect to two RNCs result in similar resiliency to RNC failures as having full-mesh connectivity between base stations and RNCs. *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.*

Acknowledgment

The authors would like to thank Dr. Li Li for the helpful discussion on mapping the optimal RNC assignment problem into a maxflow problem.

References

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

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

[3] G. Heijenk, G. Karagiannis, V. Rexhepi, and L. Westberg, "Diffserv resource management in ip-based radio access networks," in *Proceedings of 4th International Symposium on Wireless Personal Multimedia Communications (WPMC'01)*, Aalborg, Denmark, September 2001.

[4] S. Kasera, R. Ramjee, S. Thuel, and X. Wang, "Congestion control policies for ip-based cdma radio access networks," in *Proceedings of Infocom*, San Francisco, CA, April 2003.

[5] H. el Allali and G. Heijenk, "Resource management in ip-based radio access networks," in *Proceedings CTIT Workshop on Mobile Communications*, February 2001.

[6] D. L. Eager, E.D. Lazowska, and J. Zahorjan, "Adaptive load sharing in homogeneous distributed systems," *IEEE Transactions on Software Engineering*, vol. 12, no. 5, 1986.

[7] V. Harinarayan and L. Kleinrock, "Load sharing in limited access distributed systems," in *Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, 1991, pp. 21–30.

[8] Y. Wang and R. Morris, "Load sharing in distributed systems," *IEEE Transactions on Computers*, vol. 34, no. 3, pp. 204–216, 1984.

[9] M. Litzkow and M. Solomon, "Supporting checkpointing and process migration outside the unix kernel," in *Usenix Winter Conference*, San Francisco, California, 1992.

[10] S.-H. Oh and D.-W. Tcha, "Prioritized channel assignment in a cellular radio network," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1259–1269, July 1992.

[11] C.H. Yoon and K. Un, "Performance of personal portable radio telephone systems with and without guard channels," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 6, pp. 911–917, August 1993.

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

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

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