ISSN 2229-5518

Handoff Implementation with Finite Bounce for Load Balance in Clustered Mobile Networks

Rahul Shome, Saksham Manchanda , Parag Kumar Guha Thakurta

Abstract—Previous work on load balance has focused on the single handoff from a congested base station to a neighboring base station, within the range of possible single handoff. However, with the view of congestion in mobile networks which is liable to make the neighboring base stations congested as well, conventional methods fail to balance the load optimally and hence results in call blocking and degradation of quality of service. The proposed technique looks to implement intelligent heuristics in successive handoffs post clustering of the differently congested regions of multiple base stations. The new method promises to greatly reduce the probability of call blocking and failure to balance the load in congested mobile networks by managing to implement handoffs beyond the immediate possibly congested vicinity of the congested base station. The selection of this path of successive handoffs has been proposed in a computationally inexpensive manner.

Index Terms—Load Balance, Handoff, Mobile Networks, Call Blocking, Congestion , Traffic Based Clustering, Path Discovery

1 INTRODUCTION

A given geographical area consists of hexagonal cells each served by a base station. A group of cells are in turn served by a mobile switching center (MSC). Traffic can move within the base station or between base stations. Blocking describes the situation when a user attempts to make a call and is not able to reach a dialed subscriber due to lack of resources. Currently a call is transferred from a heavily loaded Base Station to a lighter loaded neighbor whenever possible. We can expect entire regions to suffer from congestion at certain times like office hours. This leaves a cluster of Base Stations heavily loaded and unable to handoff calls to its neighbors. Thus heavily loaded regions experience higher call blocking while the lighter loaded base stations resources are idle. Different techniques have been proposed to deal with the scenario.

One of the methods proposed is a self-organizing load balancing framework[1]. It provides self-optimizing load balancing methodologies to improve the Fixed Relay Station (FRS) based cellular networks. A Self-organizing Cooperative Partner Cluster (SCPC) concept to dynamically select optimal partners of each BS and RS. This clustering is made to improve upon methods that do not implement clustering. There are five types of handover in Fixed Relay System networks: Intra-cell RS-RS, Intra-cell BS-RS, Inter-cell BS-BS, Inter-cell RS-RS and Inter-cell BS-RS handover.

Alternate path routing (APR) provides a load balancing and route failure protection by distributing traffic among a set of diverse paths.[2] Destination selection and optimal path identification provides a distribution of traffic from congested regions. Route discovery and route maintenance schemes in single-path and multi-path routing algorithms[3] deal with the actual evaluation of possible routes for the selection of an optimal one.

The performance of the evaluated route is generally estimated through factors like delivery ratio, latency, delay, jitter and loss. These depend on network parameters such as TTL, buffer size, time and load.[4] Factors impacting network measurements include cross traffic, architectures of intermediate nodes, complex interaction of hardware resources and protocols at various levels, as well as implementations that often involve competing and conflicting requirements.

In our approach, we take the handoff model one step further by 'bouncing' calls multiple times. So we can handoff calls further than the immediate neighbor of a congested Base Station which helps balance the load among more number of cells thus reducing call blocking.

2 PROPOSED MODEL

The distribution of load over cells in a mobile network does not provide equitable use of resources over extended regions due to differential traffic load. Real time load data is infeasible in terms of computational complexity, hence necessary statistical approximations are made and the temporal snapshots of the load state of the system.

$S_i = \{ s_j \mid j \in N \}$
where N is number of base stations
s_j is the state of a base station marked j
S_i is the collective state of the entire system at epoch sampling instant i

s_j can store various extent of details about the system, most importantly traffic load data.

The statistical model is intended to give an approximation of the current state of the system from pre-recorded data about the system. The intervals between consideration of the state of the system depends upon the variability of the system state.

$\Delta time = S_i.time – S_{i+1}.time$
where S_i.time is the time at which a snapshot is taken

The intervals between S_i and S_{i+1} ¬, where i and (i+1) are consecutive recorded samplings of the state of the system, need not be same for all I, ie. Δtime varies with depending upon the variability of the state of the system during the interval.

ISSN 2229-5518

Δtraffic = Si.t – Si+1.t
where Si.t is the traffic load at the time Si.time

$$\frac{\Delta \text{traffic}}{\Delta \text{time}} = \Gamma$$

To maintain a manageable value of Γ that remains constant between consecutive intervals, Δtime has to be changed ie. the times of sampling have to be changed. When there is high variability of traffic, samplings should be made more frequently and when the changes are not drastic, samplings can be made after more extended intervals.

The applicability of this in real life network usage data reveals that the traffic load changes drastically during very concentrated periods of time on particular days. For example, the changes are very pronounced during the office hours on weekdays, in commercial regions populated by such consumers, who are more likely of increase network usage during these hours. Thereby, the sampling during this period has to be very frequent and depending upon the computational overhead associated with the updation of the load data of the system, the updation can be done at intervals as small as 5 to 15 minutes.

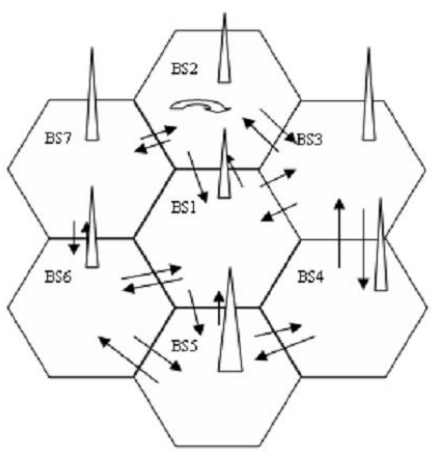

Fig 1 : General Mobile Architecture

2.1 Pre-processing Data

The proposed model uses information about the base stations to implement efficient load balance in the system. Knowledge of the system as a whole, enables the approach to take global considerations into account while performing the required optimizations to prevent call blocking.

2.1.1 Real world coordinates

The coordinates of the base stations in the given region : latitudes lx and longitudes ly

Call handoff is possible between a pair of base stations only if the distance between the base stations is less than the threshold distance. To reduce complexity of computation only the latitude and longitude is considered and not elevation.

2.1.2 Network traffic

The network traffic at the base stations : t

The most important dynamic condition affecting call handoff is the prevailing traffic load at the base stations. The traffic load is taken from the state information of the system, taken from the statistical data. The data about the individual base stations provides a feasible approximation to base any real time algorithm within permissible degree of error.

2.1.3 Load sharing capacity

The load sharing[6,7] of base stations : c

The extent to which the base stations can implement call handoff is dependent upon load sharing capacity of the base stations. The effect of the load sharing capacity determines the call handoffs at different levels of traffic load on the base stations.

2.1.4 Handoff delay

Handoff[8] overhead or delay for base stations : h

The time delay associated executing a handoff is a measure of the loss of quality of service.

For an existing call, if h>H , where H is the maximum permissible delay in a continuous voice call, there is a non-permissible loss of quality of service.

The overhead associated with handoff relates to the channels being used and the computation that goes on at the base stations regarding the handoff.

2.1.5 Threshold distance

The threshold distance up to which a single handoff is possible without unacceptable degradation of quality of service : x

Here x is not the mere Euclidian consideration. The distance with respect to the entire coordinate vector is taken into account. The value of x determines the distance up to which a single handoff is feasible, taking into account all the different factors affecting the single call handoff and its associated effect on the degradation of the quality of service for the ongoing call, or the delay caused to the waiting call.

2.1.6 Coordinate Vector

D< w1.lx ,w2.ly ,w3.t ,w4.c ,w5.h >

Where w1, w2, w3, w4, w5 are appropriate weights assigned to the respective dimensions

The coordinate vector takes into account all the factors affecting call handoff and presents a five-dimensional coordinate system for the proper representation of the state of the mobile network.

The weights are assigned to represent the differential contribution of the different factors to the handoff considerations. Changes to the relative weights are expected to give changes in the nature of the algorithmic solution since all subsequent distance measurements are done with respect to the proposed coordinate system.

2.1.7 Static Records

The static records present at every base station are used in preprocessing. The static records are computed beforehand, or at intervals when the contents of the record changes.

Li <base station id, D, b>
For all base stations I within a distance x of the respective base station where Li is stored
Where base station id is an identification for each base station
D is the coordinate vector of the base station
b is the binary check which is 1 if the base station lies on the boundary of the group of base stations falling in the region x, 0 if it is an internal base station

The base station id is required to uniquely identify the base stations for all further computations. The coordinate vector, as already mentioned is used to determine the variable effects of the different factors on call handoff and quality of service. The binary check, b is needed to minimize the number of handoffs in case of repeated handoffs. The consideration of the boundary base stations obviates the necessity to make multiple handoffs when one handoff will suffice to reach the same distance. Thereby, the prevailing factor here is the overhead associated with the repeated handoffs which cumulates over the handoffs, and not the loss in quality of service for the selection of a farther base station with respect to a nearer one, which will involve a greater number of handoff requirements.

2.2 Pre-Processing

Pre-processing is done on the available data at sustainable intervals of time so as to enable real time successive handoffs with the results of preprocessing that would provide acceptable approximations for the parameters for handoff path selection, without compromising on either correctness of the handoff or the time complexity.

2.2.1 Clustering

All base stations can be classified into three types based on the load on the station : Lightly loaded, Medium loaded and Heavily loaded. All these types have a well-defined range for classification.

Load Based SCAN with respect to D in five dimensions[5] limiting cluster size to 1 based on the condition of light load, so that all outliers are also unique clusters. This gives us clusters formed on parameter D. These clusters thereby include the appropriate effect of the traffic and load at the corresponding base stations and depending upon the weights, can cluster on a combination of real world distance parameters and load parameters. This method is capable of forming irregular clusters depending upon the parameter. This reduces the number of clusters significantly, which in turn should reduce computation cost.

It is important to consider only those outliers that are lightly loaded to be defined as singleton cluster. Thereby, load balancing opportunities will not be overlooked post-clustering. We get clusters classified as heavily loaded, medium loaded and lightly loaded(possibly more levels of classification depending upon need). Also the clustering algorithm is computationally efficient and runs with the average time complexity of $O(n * \log n)$.

Eps-neighborhood of a point: The Eps-neighborhood of a point p, denoted by NEps(p), is defined by
$NEps(p) = \{q \in D \mid dist(p,q) \le Eps\}$.

Minimum Number of points: The minimum number of points required in the ε-neighborhood of a point to make it part of a cluster in defined as minPTs.

Procedure:
The algorithm requires two parameters: ε (eps) and the minimum number of points required to form a cluster (minPts). It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.
If a point is found to be part of a cluster, its ε-neighborhood is also part of that cluster. Hence, all points that are found within the ε-neighborhood are added, as is their own ε-neighborhood. This process continues until the cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.
Pseudocode:

```
LBSCAN(D, eps, MinPts)
  C = 0
  for each unvisited point P in dataset D
    mark P as visited
    N = regionQuery(P, eps)
    if sizeof(N) < MinPts
      mark P as NOISE
    else
```

```
      C = next cluster
      expandCluster(P, N, C, eps, MinPts)
   end for
   for all points P marked as NOISE
      if P is lightly loaded
         Make P a cluster
   End for

expandCluster(P, N, C, eps, MinPts)
  add P to cluster C
  for each point P' in N
    if P' is not visited
      mark P' as visited
      N' = regionQuery(P', eps)
      if sizeof(N') >= MinPts
        N = N joined with N'
    if P' is not yet member of any cluster
      add P' to cluster C
```

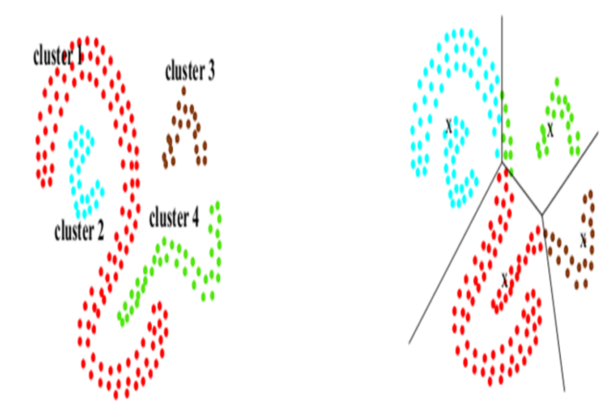

Fig 2 a) LBSCAN b) CLARANS

2.3 Real Time Route Discovery for Handoff

In case a base station has a traffic load beyond the threshold we will need to balance the load and thus be in a position to accept new calls. If possible a single handoff will be implemented. However, if that is not permissible, multiple handoff is performed.

Procedure:

First the destination needs to be identified. If the source is in a lightly or medium loaded cluster, the handoff will occur in the same cluster. However, if the base station is in a heavily loaded cluster, ie. t>tmax, nearest lightly loaded cluster is considered. Base stations in that cluster are selected at random and if the distance from the base station is greater than the minimum, all the base stations in that neighbourhood are removed and this is repeated till all the base stations are checked. The handoff with successive bounce is implemented with multiple bounce with this selected destination. The handoff starts occurring from the destination base station and proceeds towards the congested base station. If the number of handoffs exceeds the maximum permissible number of handoffs that would not compromise the quality of service, the handoff process is stopped, else the successive handoffs will reach the source base station and the new call can be accommodated.

```
Base_Station_Call(id,call)
        If Si.next_updation >= current_Time()
                Update_State()
                Update_Vectors()
        End
        If Lid.D(t) < tmax
                Allocate_Call(id,call)
                call_Allocated=True
        Else
                Handoff(id,call)
                call_Allocated=False
        End
End

Handoff(id,call)
        Bsource=id
        call_Allocated=False
        For all Sort_By_D(L Bsource (id), Bsource)
                If LBsource.D(t) < tmax
                Base_Station_Call(LBsource (id),call)
                        call_Allocated=True
                        break
                End
        End
        If call_Allocated==False
                Finite_Bounce(Bsource,call)
        End
End
Finite_Bounce(Bsource,call)

        C = Nearest_Cluster_Copy();
        N = Count_Base_Stations(C);
        min_BS_Id = Cmedoid
        While(N!=1)
                rand_Sel=random(N)
                BS_Id = Crand_Sel
                If distance(BS_Id, Bsource) < distance(Bsource, min_BS_Id)
                min_BS_Id = Crand_Sel
                Else Remove_From_Cluster(C, Lrand_Sel, min_BS_Id)
                End
        End
        Bdest=min_BS_Id;
        no_Handoffs=0;
        from_BS=Bdest
        While(no_Handoffs<threshold_QOS)
                Lcurrent=LFrom_BS
```

```
                near_Id=Find_Nearest_BS(from_BS, Bsource, Lcurrent)
        call_Allocated=Handoff_Source_Dest(id, call, from_BS, near_Id)
                If(call_Allocated==True)
                        from_BS=near_Id
                        no_Handoffs++
                        Continue
                Else
                        Lcurrent=Lcurrent-BNear_Id
                End
        End
        End
```

3 SIMULATION

A mobile network is simulated in Matlab R2009a. 2500 base stations are considered in the network and the traffic is simulated with a peaks function.

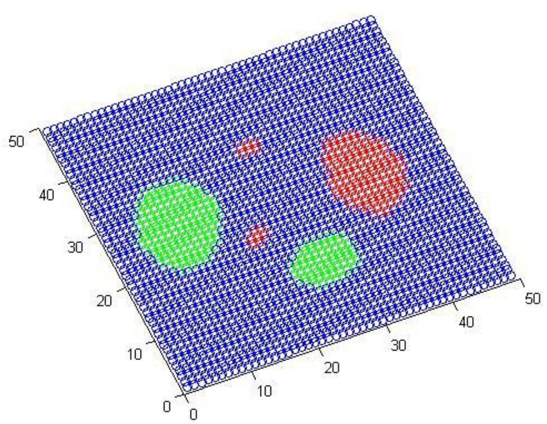

Fig 3: Initial network. Green(lightly loaded), Blue(medium loaded), Red(highly loaded)

100000 calls are simulated. The arrival of calls occurs in a weighted fashion, ie. More loaded the cell, more likely it is to get another call request in that time interval, due to pattern of network usage.

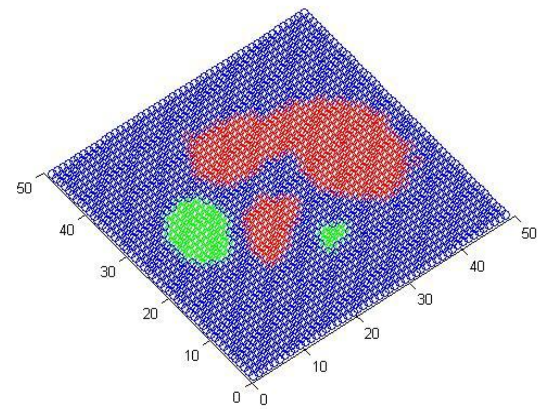

Fig 4: Network after simulated 100000 calls, without handoff

10504 calls are blocked when they get no free channels, without implementation of handoff scheme of load balance.

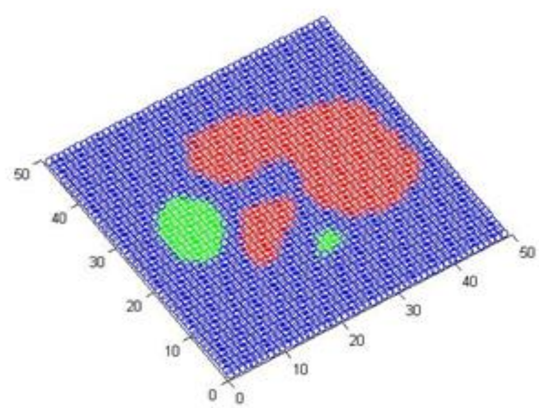

Fig 3.3 Network after simulated 10000 calls, with handoff

2812 calls are blocked after implementation of single handoff and multiple handoff scheme. The threshold for number of handoffs is 5, in this diagram.

There is a reduction in the number of blocked calls as well as a better distribution of traffic in the network.

4 CONCLUSION

The use of this algorithm is to apply both the methods of clustering and route selecting for implementing multiple handoffs. The main advantage of using LBSCAN is that is capable of forming irregular shaped clusters. This not only tries to accommodate the new call and hence as per the results, reduces call blocking, but also, can lead to better load balance and dis-

ISSN 2229-5518

tribution as the calls are being distributed away from highly loaded base stations. Further work will analyze the parameters taken into consideration and explore possibilities of increasing the number of possible handoffs.

REFERENCES

[1] L. Xu, Y. Chen, and Y. Gao, "Self-organizing Load Balancing for Relay Based Cellular Networks", in Proc. CIT, 2010, pp.791-796.

[2] Marc R. Pearlman , Zygmunt J. Haas , Peter Sholander , Zygmunt J. Haas Peter Shol , Siamak S. Tabrizi, On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks,2000

[3] *P.P. Pham and S. Perrau, "Performance Analysis of Reactive Shortest Path and Multi-Path Routing Mechanism with Load Balance,"Proc. IEEE INFOCOM Conf., pp. 251-259, Apr. 2003.*

[4] Adriano Galati and Chris Greenhalgh, "A New Metric for Network Load and Equation of State", *Fifth International Conference on Networking and Services* School of Computer Science, Nottingham, *UK, 2009*

[5] K. Mumtaz and K. Duraiswamy, "An Analysis on Density Based Clustering of Multi Dimensional Data," *Indian Journal of Computer Science and Engineering*, Vol. 1, No. 1, 2010, pp. 8-12.

[6] S.K. Das, S.K. Sen and R. Jayaram, A dynamic load balancing strategy for channel assignment using selective borrowing in cellular mobile environment, in: Proceedings of I EEE/ACM Conference on Mobile Computing and Networking(MOBICOM '96), Rye, NY (November 1996) pp. 73–84.

[7] S.K. Das, S.K. Sen and R. Jayaram, A structured channel borrow-ing scheme for dynamic load balancing in cellular networks, in:Proceedings of IEEE International Conference on Distributed Com-puting Systems(ICDCS), Baltimore, MD (May 1997) pp. 116–123.

[8] S. Tekinay and B. Jabbari, Handover and channel assignment in mobile cellular network, IEEE Communication Magazine (November 1991).

- *Rahul Shome is currently a final year undergraduate student pursuing a B.Tech in CSE at NIT Durgapur. E-mail: rahulshome.in@gmail.com*
- *Saksham Manchanda is currently a final year undergraduate student pursuing a B.Tech in CSE at NIT Durgapur. E-mail: saksham.manchanda@gmail.com*
- *Parag Kumar Guhathakurta is currently an associate professor in the department of CSE at NIT Durgapur. E-mail: parag.nitdgp@gmail.com*