A Service Oriented Framework for Multimedia Radio Networks

Asma Ben Letaifa, Sami Tabbane and Zièd Choukair
Ecole Supérieure des Communications de Tunis (SUPCOM)
Unité de recherche MEDIATRON
Route de Raoued, Km 3.5 –2083 Cité El Ghazala - Tunisia
{asma.benletaifa, sami.tabbane}@supcom.rnu.tn

***Abstract* - In order to enable fast deployment of new emerging services over multimedia radio networks, it is important to design an efficient service-based platform with necessary traffic management. Among the solutions, mobile network platforms must be able to answer efficiently and on time to the expectations of the users through dynamic adaptation of the architecture to answer the demands on time and with the minimum cost. it runs dynamically by modifying services features according to subscribers' requirements, network capacity and QoS parameters. In this paper, we propose a new service oriented framework for wireless radio networks, based on a hybrid algorithm in which we adopt an adaptive approach toward optimal traffic distribution. It is inspired from methods from graph theory, operational research, network optimization and load balancing work.We use media streaming for "*Emergency Services*" as an example to illustrate how such a service can be provided. Performance evaluation shows that the platform can substantially improve the conventional approach and achieves similar performance than the one obtained by hybrid scheme. We present in this paper some results which show the congestion status of servers and links obtained from the simulation platform.**

***Index Terms* - reconfiguration, wireless networks, multimedia services, graph theory, network optimization, load balancing, compensating paths, replication**

I. INTRODUCTION

With the growth of wireless subscribers' demands, service providers and telecom mobile operators face the challenge of maximizing their network capabilities with their existing infrastructure. This paper develops our work to find an automatic method to optimize the network use to answer subscribers' demands and to reconfigure services deployment on the network when necessary to face the challenge of maximizing the resources use and guarantying subscribers QoS when their traffic and mobility change.

We propose a method to rapidly answer user's demands, optimize link's use by maximizing the bandwidth offered and reducing delay on routers and nodes. This can be done by shifting the traffic on the network via other links. This paper describes a method inspired from load balancing, operational research, graph theory and network optimization. When no new link is available, we develop a new reconfiguration method in presence of congestion using "*Compensation Paths*". This work is based upon compensation path and server shadows which duplicate the information on visited web pages and conserve a copy in other processors.

We present here a new result in the area of reconfiguration of interactive services in presence of congestion or non available path. More precisely, we consider a set of servers which have the same features, i.e., able to perform the same tasks and provide the same set of services to their clients. In case some of them become out of service (congested) or if there is no path to send information correctly, we reconfigure the system so that the clients of congested servers are served by some other fault-free servers.

Compensation paths are an efficient way of shifting spare resources from where they are available to where they are needed. The proposed solution is focused towards "*Emergency Services*" and is based on both centralized and distributed approaches to execute real-time services. We have limited our work to apply this approach only to emergency services because such services are mandatory whereas, for other applications, it is not as useful to duplicate the information on neighbour servers to use it in case of congestion or luck of ordinary path. Let us remind that the approach we propose replicates a limited amount of information for normal platform functioning.

The rest of the paper is organised as follows. Section II presents the overview and related work. In section III, we develop a network model Based on this model, we propose a hybrid solution to maximize both rate and subscribers numbers. Section IV discusses the proposed solution. Section V verifies the model using experiments. Finally, section VI concludes the paper and highlights future work.

II. OVERVIEW AND RELATED WORK

Service deployment in wireless networks experiences several specific radio related problems such as link errors, jitter, delay variations and then heterogeneity of user equipments due to the lack of potential infrastructure support. Proper traffic engineering scheme can help to alleviate some of these problems, in particular using multipath transmissions. Traffic engineering has been used in the internet to achieve better resource use and to optimize network performance. The output of traffic engineering [1] is a set of paths and link loads which improve the performance according to available resources. The problem can be formulated using linear programming problem. By solving the optimization problem [2], each link can have a weight so that the routing protocol will select the data forwarding paths in order to deliver the traffic optimally. Authors of [3] solved the dual problem by a computational simplified solution of the linear programming. Reference [4] achieved a near-optimal solution using destination-based aggregation of traffic and approximating unequal split of traffic using heuristics for traffic splitting to deal with the problem of unequally splitting the traffic.

To manage traffic in the proposed framework, we will talk about hybrid algorithm that uses studies achieved about centralized and distributed techniques. Those two approaches can potentially be applied in node/path selection. In centralized one, a node computes the optimal data routing and distribute the result of the operation to the concerned nodes. This implies that this node has knowledge of the network topology. While this can be a possible solution when the network size is small, it becomes unpractical in large scale networks. Distributed approaches commonly seek for a solution locally.

Following what is done in frequency assignment problem, we will talk about distributed approach to assign the optimal path for each {Source, Destination}. [5] presents several algorithms for distributed problems: Asynchronous backtracking, Asynchronous weak commitment and distributed breakout algorithm (DBA). The first two algorithms are based on node priority; each node has an identifier to indicate its priority. The node with the lowest priority has to change selected path if it is selected before by a node with higher priority. The algorithm stops when a satisfactory solution is found or if no solution can be obtained. In DBA, the constraints are initially weighted according to some predetermined parameters. DBA presents the drawback pf requiring a global synchronization which is not suitable for radio networks. Where there is no any available path to lead data from the server to its clients, we use methods based on compensation path, but when its is possible to change the link, we perform an hybrid algorithm and consider concepts and methods taken from graph theory, network optimization and operational research, each of them is providing partial solutions to help us finding solutions to our reconfiguration problem.

A. Graph Theory

Many algorithms and methods are used from graph theory especially in transport network. In this respect, operators must also find the best route with minimum cost and maximum throughput in serving their applications. We consider that finding solution to a network can benefit from graph theory and moreover, it helps schematizing properly the network, with links and nodes. We can also operate on the graph to formulate the problem, consider the constraints and find the optimum serving way. For such purpose, many algorithms can be found in graph theory.

We can use the maximum flow algorithm to answer the requests with minimum cost and obtain the list of the links that lead to a destination but with minimum cost. Dijkstra algorithm indicates also the short possible path. Obtaining maximum flow is a fundamental problem in graph theory with a variety of important applications such as transportation and assignment problems. In this paper, the authors present the first distributed self-stabilizing algorithm for the maximum flow problem. Starting from an arbitrary state, the algorithm computes the maximum flow in acyclic network in finitely many steps. Many sequential algorithms exist for the maximum flow problem. Ahuja, Magnanti and Orlin [6] present a survey of some of those algorithms, Bazarra, and al. [7] propose the maximum flow algorithm for linear programming problem with a special structure such as transportation and assignment problems. To date, the fastest sequential maximum flow algorithm is due to Goldberg and Tarjan [8]. They work on the preflow-push method which works in localized manner implemented in distributed network and needs to maintain shortest path information. However, it does not tolerate dynamic changes in the network structure.

B. Network Optimization

Network optimization is a special type of linear programming model. It can be used to help the operators plan a network by localizing base stations, servers and others equipments. Engineering of mobile telecommunication networks consists of two major problems, the design of network and the frequency planning. For service furniture, we include a third major problem which consists in server location. It consists in positioning servers on potential sites in order to fulfil some objectives and constraints.

The server positioning problem deals with finding a set of sites for the equipments which consider many constraints like distance to potential zone, cost and the bandwidth offered in the zone to cover. Many search algorithms have been used to solve multi-objective combinatorial optimization problems [9] which do not generally have a unique solution like in single-objective linear programming but share one common feature: at each iteration, a single-objective linear programming problem is formulated to generate a candidate solution or a set of candidate solutions for a DM's (decision-maker's) examination. Exact algorithms such as branch and bound and dynamic programming have been used to solve small instances of bi-objective problems. Branch and bound method consists on finding two boundaries to limit the solution domains and after, doing the best to find the solution in this domain. For dynamic programming, it considers a sub-problem which solution is trivial; we try to find a global one by adding precedent steps one by one and calculating optimal solution for every step.

C. Operational Research

Operational research methods can help operators to find the best location of its nodes. To place a server, he can choose between two fundamental ways: determine the center of revolves (gravity center) in order to reduce the distance, or the second way with discretion method when time to access is more important then other parameters. It gives multiple location choice and distributed solution. Operational research methods give always a basic solution and after tries to optimize it thanks to dynamic programming or simplex algorithms or others. In the bibliography, we found method like Franck-Wolfe, linear programming (like simplex algorithm which found an optimal solution by testing vertex which one can give the best solution), branch and bound [10] method based on the principle of separation of the studied zone on others sub-zones to find a first solution and after generalized it for all the problem, or Djikstra algorithm applied to identify the shortest path. They perform all on the graphical solution, which, when found, can usually be optimized.

D. Load Balancing

In wireless network, some MSC can be heavily loaded while others are lightly loaded due to *Ad Hoc* distribution of subscribers, thus, resulting in a poor overall network performance. By transferring loads from heavily loaded feeders to lightly loaded ones, network reconfiguration can balance feeder loads and alleviate overload conditions of a network. Aoki et al. [11] formulated the load balancing and service restoration problem by considering the capacity constraints as a mixed integer non-linear optimization problem and converted the problem into a series of continuous quadratic programming sub-problems. Baran and Wu [12] formulated the problem of loss minimization and load balancing as an integer programming problem. They used a heuristic solution algorithm based on the method of branch exchange, where different radial configurations are generated, improving the objective function and originating a sequence of manoeuvres to be performed on the network. In order to guide the search for configurations, they do load flow calculations, in iterative and direct simplified ways.

Other main types of the load balancing problem have been studied, we talk firstly about Schaar et al. [13] who apply heuristics to solve the problem; in order to minimize the mean response time of a task, every new task is scheduled to be executed either locally or at a remote host, depending upon the current load distribution. A copy of scheduler runs on each workstation, this copy decides, where locally created, tasks are to be executed. Alanyali et al. [14] incorporates finite capacity constraints in resources in their algorithm to minimizing the cost and the average of blocking call, this can be done by approximating the system with traffic flow problem. Kremien et al. [15] consider information dissemination and allocation decision making. However, terminal or subscribers mobility are not considered in these studies.

E. Flow Deviation Method

It is based Frank-Wolfe (flow deviation) method and used for solving general nonlinear programming problems with convex constraints sets in order to reduce the value of cost function to its minimum.

In the following section, we describe briefly problems that cause congestion and the cost-function that we use to minimize it and how we proceed to make reconfiguration.

III. NETWORK MODELLING

We modelled the network as a directed graph G (V, E), where V is the set of nodes and E is the set of links. We consider a set of node pairs (S, C), 1, 2, …, Q where Q =N(N-1) is the total quantity of node pairs. Dq is commodity for each Source-Destination pair (S, C), Dq may be routed via the Kq different acyclic paths binding node S to C. These paths are indexed a1q, a2q , …,aKqq where any path can be represented as a vector akq,i =1 if path a1q uses link I, 0 otherwise.

A management system knows the load of each server and each link for each group of nodes. It knows also the status of all links around the area and can then balance the traffic when it seems necessary, i.e. when blocking probability is increasing or when delay is felt. In the network model, all physical equipments are represented by nodes: routers, switchers, servers and the group of subscribers located in the same zone.

A. General Framework

To help users access services in a wireless environment without or with limited infrastructure support, we assume that some of the nodes are able to serve as application-layer routers. They become the interconnection between different networks and form a hybrid one which connects on one side laptops, PDAs, smart phone and on the other side connected networks and also satellite networks. We use a measurement parameter we call the *"Distance Vector"*, which indicates the distance from node I to node J that optimises function cost [16]. Each node maintains a routing table entry based on *"Function_Cost"* for every known destination. Each node broadcasts periodically a route advertisement packet containing a new measurement. When a node receives another node's route advertisement, it updates its own route entries if of course; the routing advertisement contains more updated route information. Given the above information, we can make local optimization, done in a subgroup of the total platform, made by searching the best path to route data or by updating the used path by advertisement messages. The objective of the optimization is to minimize the generated traffic and balance the use of logical links, while satisfying as much traffic demand as possible. For node "I", a linear programming formulation can be obtained thanks to [16]. The proposed platform is designed to ease service deployment and enhance traffic delivery performance in wireless networks. However, using the physical topology to perform traffic delivery suffers serious drawbacks because of the unpredictable evolving characteristics of wireless links. Therefore, building an overlay on top of wireless network becomes a necessary yet challenging problem.

B. Traffic Management

In this paper, we consider that the servers have same functionalities. In other words, servers are able to perform the same tasks and provide the same services to the clients. If the system involves different servers organised into subsets of servers with same functionalities, then the model considered in this paper can still be used; each subset must be considered separately. According to [17], this problem is very similar to the problem of providing fault-tolerant file systems, which has been extensively studied. Hence, we considered relevant techniques and models. As for file systems, interaction between the server end the clients generates modification in the path which need to be maintained, updated and recovered, in case of congestion. As detailed later, a failure in communication between primary and backup server is modelled for our needs as connection failure and then terminates this connection. Failure of communication between a server and a client is modelled as failure of the server. For our emergency services, if the server can not find a path to answer the client, it shifts the application to a neighbour available server. According to our proposed reconfiguration solution, based on compensating path, spare resources are "virtually" shifted from where they are available to where they are needed.

C. Assumption and Abjectives

On proposed platform, mobile telephony operator offers mobile multimedia applications classified in two classes: emergency ones that are critical and can't suffer delay or non available resources, and normal ones which allow retransmission or delay. According to this classification, our platform makes possible data replication in neighbour servers only for the first class. Based upon [18], we show that the main drawback of the process is the excessive amount of duplication they require. In order to guarantee recovery from two or more consecutive congestion problem, the system has to provide spare resources that are valued more than twice the resources. Proposed model considers n servers, S_1, S_2, …, S_n. Each of them provides services to a subset of clients. We make the following hypothesis. All application servers may have same functionality, serving multimedia data to clients.

- A state s_i is associated to each server: up or down.
- Any neighbour server can render any service to any client, provided that the server is informed about the current state si.
- Real-time establishment of state si to a server Sj
- If any server fails, its client should continue to be served without being aware of the failure.
- In order to preserve service quality, a server Sj should not serve more than Cj clients simultaneously.
- The number of replications done on neighbour servers is limited to avoid congestion on available resources, this phenomenon is called shadowing.
- The shadow process done on a neighbour server should have the full functionality of the main one and should periodically be updated by the current state of others services rendered to the clients.
- If a shadow server is relieving a congested one, it doesn't take all of its clients but only those which are asked for emergency services.
- If a shadow server Sj that is replacing Si fails as well, some other shadows must take over Si' clients.
- To ovoid considerable overhead on redundant servers [19], the number of neighbour replications will be limited.
- The server Si is capable of serving one or many clients at the same time.
- Server Sj is serving its clients and the ones it inherits from fault-server Si. Some of the Si' neighbours are declared to duplicate the Si state and update the information.

The example to illustrate data duplication on emergency service is given by an ambulance call service, where in case of road accident, the injured person or other drivers send a message to a server treating this application. The mobile telephony operator connects a server to emergency center where ambulances, located more or less all over the covered area, are waiting for call. When a mobile send a message, the server decides which ambulance are mobile' nearest and send an order to reach it. The ambulance has to answer the mobile before arriving. "Shadowing" works the same way than "Replication" in web consulting. We adopt the shadow notion to our approach: that is why not all information about all services will be duplicated in neighbour nodes, but only one concerning emergency services. In our solution, only clients of the out of service emergency services have their requests shifted to the node server that substitutes the congested one. Those clients of fault emergency services will be shifted on a selected compensation path. The advantage of our solution, compared to duplication done in the internet is that compensation paths shift not all the clients; but only information relative to emergency services.

IV. Proposed Solution

For emergency services, we have to find any time a server to render appropriate clients. If the state s_i of S_i is out of work due to link congestion or server overcharge, one of the neighbours declared servers must be able to render the service. The neighbour is selected through a process satisfying the cost function defined before and also available compensation path. Compensation path is defined as link connecting fault- server and the one which will relay it. One or more available path can be found according to a new neighbour server. To identify a compensation path, messages are sent from fault server to declared neighbour that saved the shadow information on client's applications. We define two types of messages:

- The first class of messages that fault server will send to find free path from its declared neighbours.
- The second class of messages that neighbours send back about the presence of a free link.

Reconfiguration with compensation paths can be more effective because, instead of dedicating a set of processes for the duplication, if the state and the support of just one set of processes, we can move effectively extra capacity through substitutions from where it is available to where it is needed. In the covered area of our model, we choose to group the nodes (servers) in groups. In computing science, two approaches are used: centralized one, where the presence of a management entity is required to supervise the network and to identify links and paths to lead services rendered to clients. The second approach is called distributed one, which is done on a group of nodes, each one is working like a network and there is a management system to supervise all the groups.

The advantages of each approach are resumed after:

- In centralized process, a manager has a complete knowledge of the platform topology, i.e., the corresponding logical graph of the system, the idle servers and out of service node of the network, the neighbouring declared relationship among them, as well as the specific load on links and accepted thresholds. However, we cannot always assume the existence of such a centralized process, at least at a reasonable cost. We present so the second approach.

- In distributed process, there is no central repository of information about the system topology. Each group of nodes with correspond servers are working together, and a group doesn't have any information or the state of the other group. We can make information on servers in the edge of indicated area; if we suppose that neighbour nodes can exchange messages.

In our work, we are borrowing those two methods and apply each other to reconfigure the network. We use the notion of Compensation paths, which are selected thanks to neighbour relationship declared and maintained by the manager or a centralized process. They satisfy at first a Function-cost on the corresponding link and done from neighbour to another to satisfy a general function cost calculated on the complete path. A centralized process knows all the shadows defined on the servers and can compute the new function-cost value on each done step.

Thanks to the both computing process (centralized and distributed), we associate two reconfiguration processes and identify the manner that we choose to extract the compensating paths. We can describe at first centralized process to compute reconfiguration on our platform. We talk in [20] about the existence of a new unit called Manager which supervise the platform with all it components. The manager knows the platform state in each configuration, so it has information about each server was working, which links are used and what them states are. The manager gives also the order to duplicate some information and control the adaptive shadow process of our platform. In case of faulty transmission, it represents the only unit of the platform which can manage the traffic and re-reroute flows over new selected links from a new server. It orders to establish a new path between concerned clients and relative servers based on the notion of compensation paths. The manager works in centralized manner and supervises the platform and its reconfiguration process.

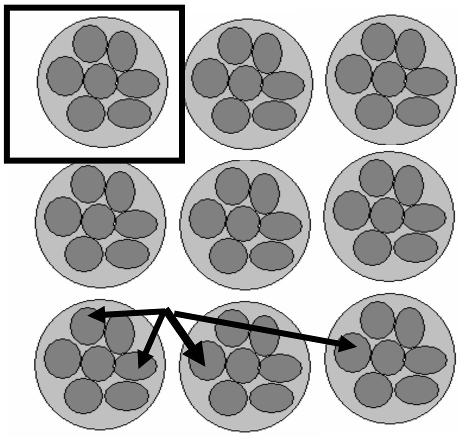

Figure 1. The platform subgroups.

This reconfiguration method, described above can be optimal, but necessitates knowledge of the whole system topology, including all nodes and neighbour relationships among them, as well us what servers are out of service and idle, and what clients correspond to what servers. According to this approach, there are some disadvantages associated with its centralized nature. It is often expensive to maintain such knowledge on the whole platform, and reconfiguration of the system depending on this process allows only one failure at a moment. In the other hand, a distributed approach doesn't give us a better solution, because we can find a solution in each group of nodes and have the idea to use such node from another group, but we haven't any information if failure has done on this selected nodes or it is still working. The fully distributed algorithm is an adaptation of what it is called the flooding algorithm used in the reconfiguration of ATM Network [20]. The main process in the fully distributed algorithm knows only their immediate neighbouring topology and exchange messages only with their immediate neighbours. The fully distributed process computes the set of compensation path in the most of the times when such a set is available. This is mainly due to the blocking problem that arises when an out of service server that search for compensation path, reserves a large quantity of the resources (nodes, links, etc.) around its neighbourhood. This reservation causes blocking which introduces very large delays in the reconfiguration process.

To overcome this impossibility to reserve a new compensation path, we are presenting a compromise that generalizes the both approaches: centralized and distributed processes. We describe here the hybrid reconfiguration algorithm, its advantages and some of each application on emergency services. We consider a platform and its nodes are divided in groups. Each group is treated as a subsystem where local reconfiguration mechanism id done. In each group, one node is playing the role of local manager and maintaining information about the group topology and all the servers' and links' states. The groups are sharing some nodes localized in the border. In the case of failure in the group gi, the local algorithm acts to find a valid compensation path using

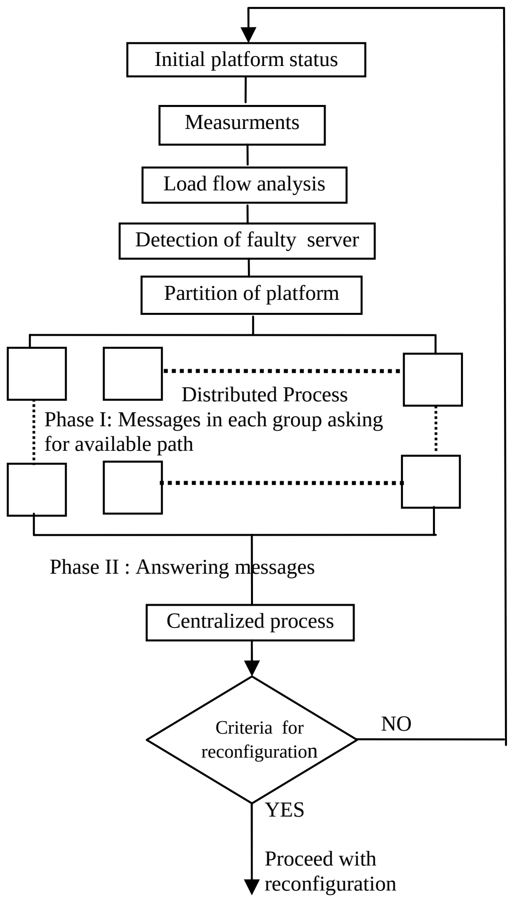

Fig.2. Proposed algorithm.

links and nodes of gi. If it needs to use nodes from the border of gi, it must ask for availability from the local manager of the neighbour gj, j≠i, because the node can be already used in the reconfiguration process. So, one local manager can't use any border node without asking the neighbour local manager or a centralized unit. (Fig.1)

Data duplication, what we called shadow, are done in two gi nodes types. Either border nodes can duplicate the information because they maintain in this way the continuity of the services in case of clients' mobility. If many fault process are detected, the local manager solves the maximum that it was possible in gi. But if the number of out of services application processes are increasing (be superior to a threshold fixed by the operator), the local algorithm must find a compensation path from neighbour groups, so it must include nodes from the border even if they don't satisfy Function_Cost. The local algorithm is encouraging in case of many fault serving services the selection of nodes that aren't belongs to previous reconfiguration process, which are still free or don't be present in selected compensation process.

Reconfiguration process or demands on free nodes, for establishing a new compensation path, are based on information exchanged between two or many groups. The local algorithm makes the node selection based on the messages that we present above. Those messages are returned to out of service nodes in order to indicate freedom or possible reservation of asked links. If one link is reserved by a local algorithm in gj, and doesn't be selected by other local algorithm in gi, j≠i, the centralized process must de-allocate the reservation. A centralized algorithm manages the reservation and de-allocation of compensation path all over the platform. Some nodes or links are reserve but not used. This happens when a node sends a message to indicate its free state to support the service; it can't stop until it receives an order from centralized process to indicate that other nodes are been used. Fig2. summarizes the algorithm process.

It is easy to see that our hybrid algorithm described above acts differently on groups (in selection process) and on the total platform on de-allocation of selected resources. Its main advantages over distributed approach is that reconfiguration messages are exchanged between the local managers of groups not all nodes can participate to path selection. Also, there is no node blocked for a long time in reservation process, because centralized algorithm acts here to avoid reservation. Over centralized approach, the hybrid algorithm gives the possibility to not generalize any search of compensation path to the entire platform. Only node which are in the same group that a fault is detected, are concerned. Our hybrid algorithm makes an efficient use of spare resources and we obtain better reconfiguration state without playing the entire nodes.

After modelling each zone by a graph with nodes representing sources and destinations, we use adjacency and incidence matrixes to find congestion on which links and consider the mobility and traffic of users. In one of our previous work, we have talked about the utility of those matrixes and how they can solve many problems. In other hand, graph theory algorithms can be applied to find the best solutions to optimize some criteria. We will expose here briefly a non-exhaustive list of algorithms and the criterion that they can optimize. Adjacency and incidence matrixes help to find congested parts of the network. Maximum flow algorithm (and its modified version of Goldberg and Tarjan) finds the best way to serve a zone according to weighting expressed on links (bandwidth, distance, transmitted throughput, etc.) Minimum cost algorithm gives a solution that optimizes the cost of the transmission. Aoki algorithm for balancing load in the network. Kunz algorithm for minimizing the mean response time. Alanyali et al. algorithm mixes a load balancing and a minimum cost objectives.

We adopt a hybrid algorithm derived from the previous studied methods that optimizes the use of the network with respect of the operator priorities. Each algorithm from those exposed above or others is applied alone and used to extract a set of solutions ranked according to the criterion that it optimizes. Hence, applying all chosen algorithms, we obtain different solutions ranked differently according to each optimizing criterion. We map the operator priorities to weightings applied to the obtained solutions in order to extract the optimum solution for the reconfiguration of the network. Our algorithm is based on the maximum flow algorithm and considers the two main objectives: maximum flow and minimum cost without reaching congestion; it works iteratively and it is performed every time a network is detecting subscribers' location change. At each time, the parameters are changing, like cost function, mobiles positions, traffic demand expressed by "Consume" Vector, which give a combination of data, image, video and voice flow. When each parameter is changing, the algorithm will give every time a new list of the best way to answer mobiles demands. At each time and when the configuration changes, the first solution done in previous configuration must be eliminated and the algorithm gives new one basing its search on the list given the first time, but including new parameters. The algorithm is executed according to the following steps:

- A management system for each zone has information on links status and knows the traffic in each connection as well as the number of hops done by each flow from source to destination.
- For the configuration number I, we know subscribers location and what are their demands. The positions of servers are also known, we can performs maximum flow algorithm to determine the circuit chosen to lead data.
- We perform also min cost algorithm to consider constraints expressed by the cost function.
- We can perform periodically an algorithm to measure load in links and to be sure that we don't reach congestion
- When subscribers are move or their behaviors change we detect it, because we can know in each iteration, the status of the links thanks to the incidence matrix, it helps us know which links are more congested than others.
- When we detect such kind of problems, we must proceed by finding a solution, the first step consider balancing the load in other link, so we have to find a way to determine a second road. Branch and bound solution can help us; but we have to classify the road according to their performance, so we try to measure if the second road can optimize the traffic.
- We loop to test the same algorithms in configuration I and configuration II to know if links doesn't reach yet congestion and if all constraints are considered.

V. Performance Evaluation

We test the hybrid algorithm on the example of ambulance call service. Some nodes from the border (Edge) are duplicating the information of the server (messages sent to ambulance and answer to mobile who is calling for help). If the server is out of service, the local algorithm will search for free nodes and paths that lead the service in rapid manner. If a selected node is already working, the hybrid algorithm doesn't work like a distributed algorithm, but an order from centralized process is done to stop previous application by shifting it and act an emergency application services in this selected node. We conducted a series of experiments based on the implementation of our hybrid algorithm on a grid topology. We have simulated two urgent services in case of fire detection and ambulance search. We consider various graph size, and percentage of faulty nodes. We have tested our algorithm on a grid composed by 10, 20 or 30 groups. Each group contains respectively 15, 25 or 35 nodes, respectively 5, 10 or 15 of them are servers declared and can duplicate the information. A family of different hybrid algorithms is produced for each size. As the percentage of out of service nodes increases, it is reasonable to expect that all algorithms will become less and less successful in finding available nodes for the faulty ones.

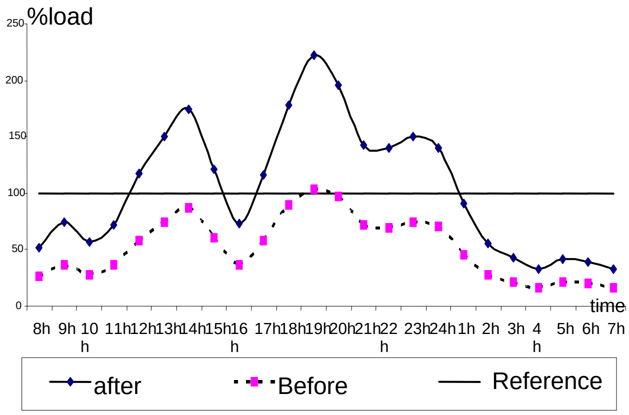

Figure 3. .Load on the server before and after reconfiguration

We can't make a decision on the number of selected neighbour node because we have to fix many parameters like throughput, link bandwidth, parameters of installed servers, capacity of neighbours, etc. In other words; there is trade-off between the amount of topological information used by the algorithm and the reconfiguration success achieved. The important question is how much the size of group can be reduced without reducing the reconfiguration probability to below an acceptable level.

But to test and validate the proposed hybrid algorithm, we developed a platform that helps to model clients' mobility and traffic models. We obtain the status of link and the load of servers across the time while the subscribers' profile is changing. We develop some scenarios but we give here curve showing that reconfiguration can avoid the users' reject rate and increase the number of admitted Video calls like it is shown at Fig.3.

VI. Conclusion and Future Work

In this paper, we propose a new architecture which can facilitate the deployment of new services in wireless networks. We investigate traffic management problem, in particular, how to route urgent traffic and reconfigure the platform in case of faulty nodes. Our reconfiguration method presents considerable advantages over the traditional reconfiguration models mentioned in Section II. Reconfiguration with compensation path in both centralised and distributed cases makes efficient use of the available resources, because it obtains higher reconfiguration probability using the same amount of extra capacity and the same number of relationship (between nodes and links). There are several unique advantages in our architecture:

- It can be done in both centralized and distributed platform architecture.
- Compensation path are generated periodically and change with every failure occurrence.
- The updates of shadow and replication information make the compensation path every time updated.

We are currently carrying out experiments using the compensation paths. Another possible avenue for further study is to examine reconfiguration process based on both minimum cost and maximum flow compensation paths, and also better reconfiguration probability for the same amount of spare capacity.

REFERENCES

[1] D. Awduche et al., Overview and rinciples of internet traffic engineering, *IETF, RFC3272*, May 2002.

[2] Y. Wang, Z. Wang, L. Zhang, Internet traffic engineering without full mesh overlaying, in *Proc. IEEE INFOCOM*, 2001, pp 565-571.

[3] Y. Wang, Z. Wang, L. Zhang, "Internet traffic engineering without full mesh overlying", in *Proc. IEEE INFOCOM*, 2001, pp. 565-571.

[4] A. Sridharan, R. Guerin, C. Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS networks", in *Proc. IEEE INFOCOM*, 2003, pp. 1167-1177.

[5] M. Yokoo, Algorithms for distributed constraint satisfaction: A review, *autonomous agents and multi-agent systems 3*, pp. 185-207, June 2000.

[6] R. Ahuja, T. Magnanti, J. Orlin, Some recent advances in network flows. *SIAM Review*, 33:175-219, 1991.

[7] M.S. Bazara, J. Jarvis, H. Sherali, Linear programming and Network flow. *John Wiley and sons*, second edition, 1990

[8] A. V Goldberg, R. Tarjan, a new approach to the maximum flow problem. *Journal of the ACM*, 35:921-940, 1998.

[9] E.G. Talbi. Metaheuristics for multiobjective combinatorial optimization : state of the art. *Technical report, LIFL, Lille*, France 2000.

[10]] S. Sayin, S. Karabati. A a bicriteria approach to the two-machine flow shop scheduling problem. *European journal of operational research*. 113:435-449,1999.

[11] Aoki k., Kuwubara H., Satoh T., Kanezashi M., an Efficient Algorithm for load balancing of transformers and feeders, *IEEE Trans, 1998, PWRD-3*, pp 1865-1872.

[12] Baran, Wu, Network reconfiguration in distribution systems for loss reduction and load balancing, *IEEE Trans. 1989, PWRD*, pp. 1401-1407

[13] Schaar M., Efe K., Delcambre L., Koppolu S., Heuristic Algorithms for Adaptive Load Sharing in Local Networks, *Systems integration*, pp 762-770, 1990.

[14] Alanyali M., Hajek B., On Simple Algorithms for Dynamic Load Balancing, *Proceedings. IEEE INFOCOM*, Vol.1, pp 230-238, 1995.

[15] Kremien O., Kramer J., Methodical Analysis of Adaptive Load Sharing Algorithms, *IEEE Transactions on Parallel and Distributed systems*, Vol. 3, No. 6, pp 747-760, November 1992.

[16] A. Ben Letaifa, S. Tabbane, Z. Choukair, A Hybrid Algorithm for Reconfigurating Services Platforms in Radio Mobile Networks, *IEEE International Workshop on Performance and Management of Wireless and Mobile Networks*, 2005, Sydney

[17] B. Liskov, S. Ghemawat, R. Gruber, P. Johnson, L. Shrira, and M. Williams, Replication in the harp file system, *Proc. 13th SOSP*, pp. 226-238, Oct. 1991.

[18] Y. Huang and P. Jalote, Effect of fault tolerance on response time – Analysis of the Primary site approach, *IEEE Trans. Computers*, Vol41, pp. 420-428. Apr. 1992

[19] A. Ben Letaifa, internal report, Dynamic reconfiguration on multimedia mobiles services platforms, Master 2003, Sup'Com, Tunisia

[20] R. Kawamura, K. Sato, and I. Tokizawa, Self-Healing ATM Network Architectures utilizing virtual paths, *Proc. Networks'92*. Kobe, Japan, May 1992.

Asma BENLETAIFA, PhD Student, School of telecommunication of Tunisia : Sup'Com, working on reconfiguration of 3G/4G networks

Sami TABBANE, Professor, Sup'Com, working or Radio mobile networks

Zied CHOUKAIR, Professor, Sup'Com, working on Services for NGN networks.