The 11th International Symposium on Wireless Personal Multimedia Communications (WPMC'08)

ENERGY-AWARE UMTS CORE NETWORK DESIGN

Luca Chiaraviglio
Politecnico di Torino
Torino, Italy

Marco Mellia
Politecnico di Torino
Torino, Italy

Fabio Neri
Politecnico di Torino
Torino, Italy

ABSTRACT

According to several studies, telecom operators account for a not negligible power consumption, and several initiatives are being put into place to reduce the power consumption of the ICT sector in general. To this goal, we propose a novel approach to switch off some portions of the UMTS core network while still guaranteeing full connectivity and maximum link utilization. After showing that the problem falls in the class of capacitated multi-commodity flow problems, and therefore it is NP-complete, we propose some heuristic algorithms to solve it. Results show that it is possible to reduce the number of links and nodes currently used by up to 30% and 50% respectively during off-peak hours while offering the same service quality.

I INTRODUCTION

Power consumption in general, and of ITC technologies in particular, has become a key issue during the last few years. The ratio of power demand versus power resources is constantly growing, and energy costs are increasing at a constant rate. The electricity cost jumped up of about 35% in Italy during the period 2004-2007 [4]. Moreover, Green House Gases (GHG) emissions have a negative impact on the world climate change [8], and people are becoming more conscious about problems that will arise in the near future due to this.

According to a number of studies, ICT alone is responsible for a percentage which varies from 2% to 10% of the world power consumption [7], due to the ever increasing diffusion of electronic devices. In this scenario, the power consumption of telecommunication networks is not negligible. For example, Telecom Italia consumes more than 2 TWh of energy, representing the second national user [6] after the national railway operator.

More recently, telecom operators have begun to adopt equipments that are able to be enter into standby mode [1].

In this paper, we consider the packet switching domain of a typical UMTS architecture. Given the network topology and a traffic demand from the Radio Access Network, we evaluate the possibility to turn off some elements (nodes and links) under connectivity and Quality of Service (QoS) constraints. The goal is to minimize the total power consumption of the core network, in which usually resource over-provisioning is large. We investigate some simple optimization algorithms. In particular, we selectively power off nodes and links of the topology following different strategies. Results show that it is possible to reduce the percentage of nodes and links actually powered on up to 30% and 50% respectively, while guaranteeing that the resource utilization is still below a given threshold. e.g., 50%.

The paper is organized as follows: Sect. II presents the scenario and the problem formulation; Sect. III describes the implemented heuristics, and results are presented in Sect. IV. Finally, conclusions are drawn in Sect. V.

II PROBLEM FORMULATION

The UMTS architecture [5] can be divided in three main domains: user equipments (UE), radio access network (UTRAN) and core network (CN).

The radio access network is composed by Node-B's and RNC's. Typically, a Node-B is the entity which controls a set of cells and the radio interface. RNC provides instead control for radio resources, and brings connectivity from Node-B to the fixed network.

The fixed network is divided into packet switching (PS) and circuit switching (CS) domains. The PS domain comprises 3G-SGSN and 3G-GGSN nodes, which implement all the functionalities of a typical IP router, i.e. routing, QoS, security. Both 3G-SGSN and 3G-GGSN nodes are connected to a traditional IP backbone.

An informal description of the design problem studied in this paper is the following:

Given

i) a topology for the UMTS fixed network, comprising nodes (RNC, 3G-SGSN, IP-routers) and links, in which links have a known capacity, ii) the knowledge of the average amount of traffic exchanged by any source/destination RNC pair, iii) the maximum link utilization that can be supported, iv) the power consumption of each link and node,

Find

the set of nodes and links that must be powered on so that the total power consumption of the network is minimized,

Subject to

flow conservation and maximum link utilization constraints.

We provide an Integer Linear Programming (ILP) formulation of the problem to precisely define it. Let us represent the UMTS fixed network infrastructure as a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. Vertices represent network nodes, while edges represent network links, being $N =| V |$ and $L =| E |$ the total number of nodes and links respectively. Let c_{ij} be the capacity of link from node i to node j and let $\alpha \in \{0, 1\}$ be the maximum link utilization that can be tolerated. Let t^{sd} be the average amount of traffic going from node $s = 1, \ldots, N$ to node $d = 1, \ldots, N$, i.e., $\{t^{sd}\}$ represents the traffic demand.

Let $x_{ij} \in \{0, 1\}$, $i = 1, \ldots, N$, $j = 1, \ldots, N$ be a binary variable that takes the values of 1 if link from node i to node j, i.e., (i, j), is present and powered on. Similarly, let $y_i \in$

$\{0,1\},\ i=1,\dots,N$ be a binary variable that takes the value of 1 if node i is powered on. Let f_{ij}^{sd} denote the amount of flow from s to d that is routed through the arc from i to j. Similarly, let f_{ij} be the total amount of traffic flowing on the link from i to j.

Finally, let $\mathcal{PL}_{ij}$ and $\mathcal{PN}_i$ be the power consumptions of the link from i to j, and of node i, respectively.

Given the previous definitions, it is possible to formalize the problem as follow:

Minimize

$$\mathcal{P}_{tot}=\sum_{i=1}^{N}\sum_{j=1}^{N}x_{ij}\mathcal{PL}_{ij}+\sum_{i=1}^{N}y_i\mathcal{PN}_i \tag{1}$$

Subject to:

$$\sum_{j=1}^{N}f_{ij}^{sd}-\sum_{j=1}^{N}f_{ji}^{sd}=\begin{cases}t^{sd}, & \forall s,d,i=s\\ -t^{sd}, & \forall s,d,i=d\\ 0, & \forall s,d,i\neq s,d\end{cases} \tag{2}$$

$$f_{ij}=\sum_{s=1}^{N}\sum_{d=1}^{N}f_{ij}^{sd} \quad \forall i,j \tag{3}$$

$$f_{ij}\leq \alpha c_{ij}x_{ij} \quad \forall i,j \tag{4}$$

$$\sum_{j=1}^{N}x_{ij}+\sum_{j=1}^{N}x_{ji}\leq My_i \quad \forall i \tag{5}$$

Eq. (2) states the classical flow conservation constraints, while Eq. (3) evaluates the total flow routed on each link. Constraint (4) forces the total link offered load to be smaller than α, while constraint (5) states that a node can be turned off only if all incoming and outgoing links are actually turned off. The big-M method is used to force this constraint, $M\geq 2N$.

The presented formulation falls in the class of capacitated multi-commodity minimum cost flow problems (CMCF) [3], i.e., the problem in which multiple commodities have to be routed over a graph with capacity constraints. CMCF problems are known to be NP-hard, so exact methods can only used to solve trivial cases. In this paper, we therefore propose some simple heuristics in order to solve the design problem also for large networks. Moreover, since $\mathcal{PL}$ and $\mathcal{PN}$ are difficult to know and vary widely depending on the considered technology, in the following we consider $\mathcal{PN}=1$ and $\mathcal{PL}<<\mathcal{PN}$ [2], so that the objective function can be pursued by trying to switch off the largest possible number of network elements.

III ALGORITHMS

The algorithms we propose consider a network in which all elements are powered on, so that $x_{ij}=1\ \forall i,j$ and $y_i=1\ \forall i$. Each algorithm then iteratively tries to switch off each element (either a node or a link). At each step, traffic is then rerouted on the shortest path for each (s,d) pair to verify Eq. (2), and the utilization constraint (4) is checked for all links. If no violation is present, then the selected element is powered off. Fig. 1 reports a schematic description of the algorithms.

```
//node optimization
sort_nodes(vect_nodes);
for (i=0; i<N; i++) {
   disable_node(vect_nodes[i]);
   compute_all_shortest_path();
   compute_all_link_flow();
   if (check_paths() == false) {
     enable_node(vect_nodes[i]);
      continue;
    }
    if (check_flows() == false) {
      enable_node(vect_nodes[i]);
      continue;
    }
}
//link optimization
sort_links(vect_links);
for (j=0; j<L; j++) {
    disable_link(vect_links[j]);
    compute_all_shortest_path();
    compute_all_link_flow();
    if (check_paths() == false) {
      enable_link(vect_links[j]);
      continue;
    }
    if (check_flows() == false) {
      enable_link(vect_links[j]);
      continue;
    }
}
```

Figure 1: The pseudo-code description of the proposed algorithms.

We implemented two different kinds of algorithms: node-oriented and link-oriented heuristics. We expect that it is more difficult to turn off a node than a single link, but the energy saving introduced in the former case is much larger, as reported in [2]. The two heuristic approaches are therefore combined so that the nodes are checked first, and then links are possibly powered off at a second stage.

Several policies can be adopted to iterate through the node set. We implemented the following ones:

- random (R)
- least-link (LL)
- least-flow (LF)

According to each heuristic, the node set is first sorted considering a given rule before iterating through all the nodes. In particular, the least-link heuristic sorts the nodes according to the number of links that are sourced and sinked at each node, so that nodes with a smaller number of links are checked first, i.e., V is sorted in increasing value of

$$X_i=\sum_{j=1}^{N}x_{ij}+\sum_{j=1}^{N}x_{ji}. \tag{6}$$

The least-flow heuristic takes instead into account first the nodes with the smallest amount of information flowing through them, i.e., V is sorted in increasing value of

$$F_i = \sum_{j=1}^{N} f_{ij} + \sum_{j=1}^{N} f_{ji}. \tag{7}$$

Finally, the random heuristic sorts nodes in random order.

Similarly, considering link heuristics, we implemented two algorithms:

- least-flow (LF)
- random (R)

which leverage on the same intuition as the corresponding node sorting heuristics: the least-flow policy sorts links in increasing order of carried flow, i.e., E is sorted in increasing value of f_{ij}, while the random policy sorts links in random order.

All possible node-link sorting heuristics have been studied. Besides these heuristics, we also tested the corresponding ones in which a decreasing order is adopted. Since they all perform consistently worse, we decide not to consider them in this paper.

IV PERFORMANCE COMPARISON

In order to assess the performance of the proposed heuristics, we consider a simple UMTS fixed network. The goal is to show that, for a given (static) traffic demand, it is possible to power off a number of network elements, and to still guarantee full connectivity between sources and destination, while enforcing that the link utilization remains smaller than a QoS threshold.

We suppose the network follows a hierarchical topology, which is typical of the packet switching domain of a UMTS network. All links are supposed to be bidirectional links, so that if link (i, j) exists, then link (j, i) exists as well. Three levels of nodes are considered: core, edge and aggregation nodes.

The network core is composed by few nodes (IP routers) that are highly interconnected by means of high-capacity links. Each link connects nodes which may be also geographically far away, e.g., optical links connecting different cities.

Edge nodes are instead used to interconnect aggregation nodes to the core nodes. Typical examples of edge devices are the 3G-SGSN and 3G-GGSN nodes. Links have middle-range capacity, i.e., smaller capacity than the one of links interconnecting core nodes. Each edge node is connected to some of the closest core nodes, and to other edge nodes.

The last level of nodes is composed by the aggregation nodes (RNC), to which the Node-B's are directly connected. Each RNC is dual-homed, i.e., it is connected to the closest pair of 3G-SGSN nodes (to guarantee alternate paths in case of failure). The links that connect aggregation nodes to edge nodes have low capacity, i.e, smaller capacity than the one of links interconnecting edge nodes.

Considering the link capacity assignment policy, three classes of links are defined: high, middle-range and low capacity links. Each class has a minimum capacity c_{ij}^{min} constraint, that was selected to be 15, 5, and 1 units of traffic respectively. Minimum link capacities are also used as link routing weights, so that the routing cost is inversely proportional to the link capacity. This is commonly adopted to force the traffic to be routed through the edge and the core nodes, rather than through aggregation nodes (which are connected by means of low capacity links). A simple minimum cost path is considered as routing algorithm, similarly to what is commonly adopted in a typical IP network. Furthermore, a QoS constraint is considered. It forces the total traffic flowing through a link to be smaller than an over provisioning factor β. We used $\beta = 0.5$ in the remaining of the paper. Therefore, after routing all the traffic, link capacities are finally assigned so that:

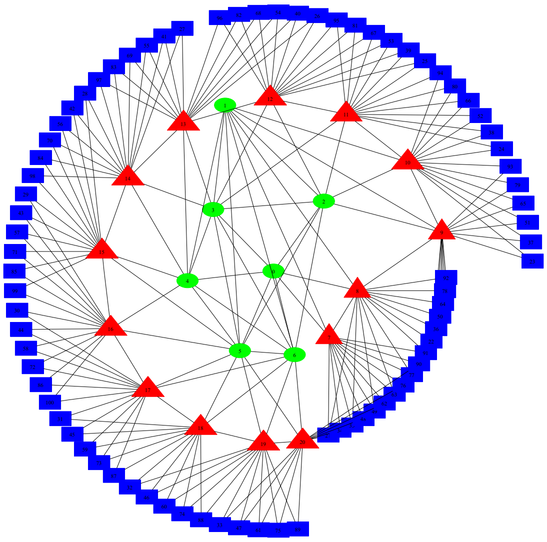

Figure 2: An example of random topology.

$$c_{ij} = \max(\lceil f_{ij}/\beta \rceil, c_{ij}^{min}) \tag{8}$$

Results presented in this paper have been obtained considering randomly generated hierarchical topologies in which 160 nodes are considered. In particular, 10 core nodes, 30 edge nodes, and 120 aggregation nodes are considered. Nodes are assumed to be placed on a plane. Core nodes are randomly connected to other core nodes with probability $p = 0.5$. Each edge node is then connected to the two closest core nodes and to another randomly selected close edge node. Finally, aggregation nodes are connected to the two closest edge nodes. An example of possibly topology is presented in Figure 2. Aggregation, edge and core nodes are represented by squares, triangles and circles respectively.

Only aggregation nodes are traffic sources and sinks. For the sake of simplicity, we consider a uniform traffic pattern, so that $t^{sd} = U[0.5, 1.5]$ [1]units of traffic if s, d are aggregation nodes; $t^{sd} = 0$ otherwise.

IV.A Simulation Results

For each considered heuristics, we collected the percentage of links and nodes that are turned off, η_L and η_N respectively. This test was repeated on 20 randomly generated topologies and traffic patterns. Fig. 3 and Fig. 4 show the comparison

---

[1] $U[a, b]$ is the function that returns a uniform random value in $[a, b]$.

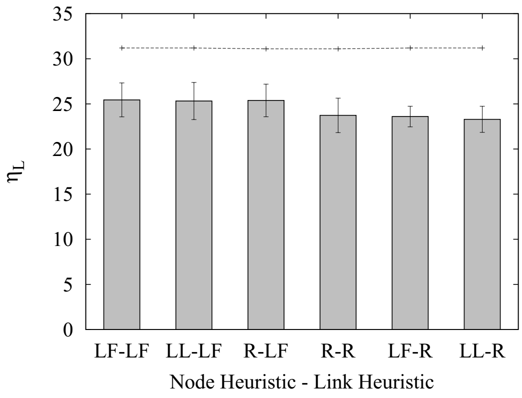

Figure 3: Comparison of the percentage of links switched off considering different algorithms.

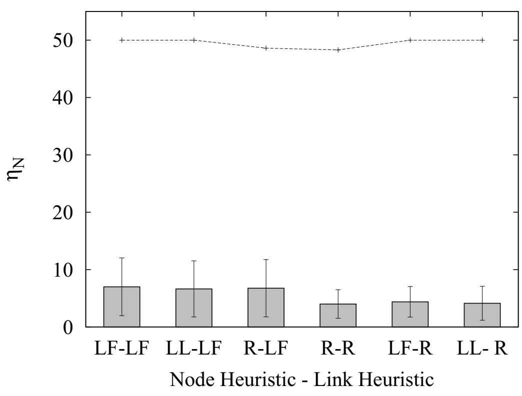

Figure 4: Comparison of the percentage of nodes switched off considering different algorithms.

of the different heuristics by reporting η_L and η_N respectively. Bars report mean values, while the error bars show the standard deviation. Labels on the x-axis report the node-link heuristic combination. A maximum link load factor $\alpha = 0.8$ was considered. We consider the same traffic demand, so that the network must guarantee to transport the same amount of traffic.

We report also an upper-bound obtained by relaxing constraint (4), so that only the flow conservation constraint is imposed. This is equivalent to find the minimum set of nodes and links that permit to route all the offered flows. This allows to better assess the impact of the QoS constraint, and the quality of the solutions generated by the proposed heuristics.

Considering η_L (Fig. 3), we can see that is is possible to actually turn off about 25% of links in the considered scenario. The node selection heuristics show very similar results, while a larger impact of the link heuristics is shown. Indeed, random link selection heuristics (R-R, LF-R, LL-R) show consistently worse results compared to the least flow selection policy (LF-LF, LL-LF, R-LF). Notice also that the best performing algorithm is only 7 percentage points below the upper bound, which shows that little improvement is possible.

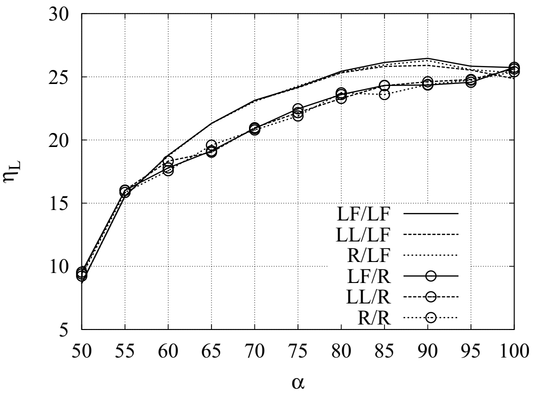

Figure 5: Percentage of links switched off versus α.

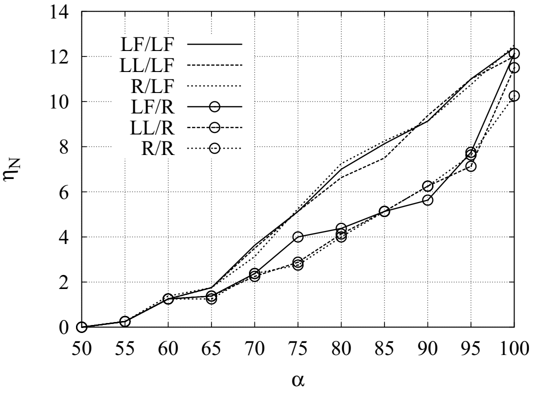

Figure 6: Percentage of nodes switched off versus α

Fig. 4 reports instead the average number of nodes that different heuristics are able to switch off: in this case, only 5-10% of nodes can actually be turned off, since Eq. (5) must be verified. Also in this case there is little impact on the node selection policy, while the LF link selection policy generally performs better. Notice that the upper bound is much higher than any admissible solution, suggesting that the QoS constraint (4) cannot be relaxed.

IV.B Parameter Impact

We performed a study on the impact of the α parameter, in order to observe the possible range of network elements that can be successfully switched off while guaranteeing a maximum offered load on links. For sake of simplicity, only mean values are reported for each heuristic combination.

Fig. 5 reports the number of links switched off for $\alpha \in [0.5, 1]$ in the considered scenario. All algorithms show large improvements for α up to 0.8; after that, little improvement is noticeable, and a final minor decrease in the average percentage of links that can be turned off is observed for values of $\alpha > 0.8$. This is due to the fact that when α is higher, a larger number of nodes can actually be switched off (see Fig. 6). This reduces the freedom of turning off other links, since not many

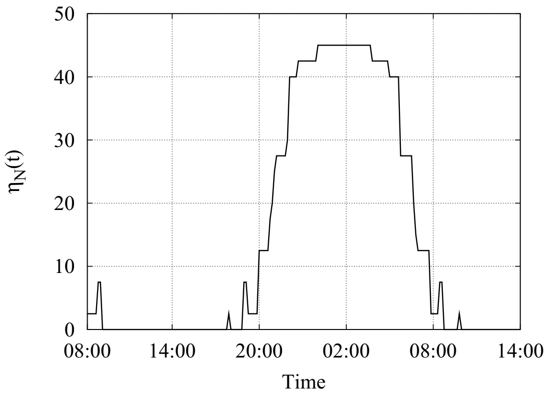

Figure 7: Percentage variation of nodes switched off versus time. The $LF - LF$ algorithm is considered.

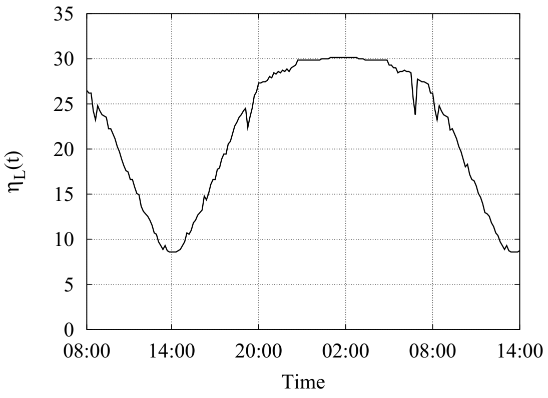

Figure 8: Percentage variation of links switched off versus time. The $LF - LF$ algorithm is considered.

alternate paths remain available. Fig. 6 shows the average percentage of nodes that are switched off for different algorithms. Similar considerations hold also in this case.

The last set of experiments investigate the effect of load traffic variation. In particular, capacities are assigned according to the peak-traffic scenario. Actual traffic is known to change according to a day-night trend. Here we assume that traffic load changes according to a simple sinusoidal function, with daily periodicity, i.e:

$$t^{sd}(t) = t^{sd}\left[\frac{1-\gamma}{2}\left(1 + sin\left(f_o t\right)\right) + \gamma\right] \qquad (9)$$

where $f_o = 24 hours$. We assume that during night the mean amount of traffic is equal to the 20% of the peak traffic, so that $\gamma = 0.2$.

Traffic is supposed to be uniformly distributed among aggregation nodes only, and routing weights and capacity assignment are performed as previously described. The Least Flow - Least Flow algorithm is adopted to observe the percentage of nodes and links that can be switched off. In this case, we select $\alpha = 0.5$ because we want to keep the same QoS constraint enforced during the design.

Fig. 7 shows the variation of the percentage of nodes switched off versus time. Interestingly, during night the percentage of nodes switched off is very close to the upper-bound (50%). On the contrary, during the day the percentage of switched off nodes is equal to zero, since the hole network capacity is required to satisfy the traffic demand.

Considering the percentage of links off (Fig. 8), we can notice that even during day it is possible to turn off some links, e.g. one out of two links between the aggregation and edge nodes that are installed for protection purpose and therefore do not carry traffic in normal conditions.

V CONCLUSIONS AND FUTURE WORK

In this paper we faced a network design problem for UMTS fixed network. We deviated from the traditional formulations of the problem, in which the objective function is to minimize cost or maximize performance, by considering the minimization of the total power consumed by the network as objective function, while connectivity and maximum link utilization are taken as constraints.

We provided an integer linear programming formulation of the problem, which shows that it is a NP-complete problem. Simple heuristics have been proposed, and their performance assessed considering some simple yet realistic traffic and network scenarios. Results (although dependent in absolute values from the chosen scenario) show that it is possible to switch off both full nodes and links, so that the total network power consumption can be reduced.

As future work, we plan to evaluate the power saving that can be achieved in a real network scenario. We aim also to study the possibility to switch off some elements of the Circuit Switching domain.

REFERENCES

[1] 3G.co.uk. Vodafone using ericsson's new power-saving base station feature, http://www.3g.co.uk/pr/dec2007/5524.htm, 2007.

[2] K. Christensen, C. Gunaratne, and B. Nordman. Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed. *International Journal of Network Management*, 2005.

[3] T. G. Crainic, M. Gendreau, and I. Ghamlouche. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Technical report CRT-2001-01, Centre de recherche sur les transports, Universite de Montreal*, 2001.

[4] Eurostat web page, http://epp.eurostat.ec.europa.eu/.

[5] H. Kaaranen, S. Naghian, L. Laitinen, A. Ahtiainen, and V. Niemi. *UMTS Networks: Architecture, Mobility and Services*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[6] S. Pileri. Energy and communication: engine of the human progress, http://www.ega.it/intelec2007/img/stefano_pileri.pdf. *Intelec Opening Keynote*, 2007.

[7] Global Action Plan. An inefficient truth, http://www.globalactionplan.org.uk/. *Global Action Plan Report*, 2007.

[8] K. Szomolnyi. Greenhouse gas effect of information and communication technologies, http://www.etno.be/. *ETNO Project Study*, 2005.