

Extending the OSS in LTE-Advance Network to Support Dynamic Resource Allocation

Ronit Nossenson

Service Performance, Web Experience BU

Akamai Technologies

Cambridge, MA, USA

Email: rnossens@akamai.com

Abstract— In this study, we present a novel architecture and procedures that extends the Long Term Evolution (LTE)-Advance Operations Support Systems (OSS) to support dynamic resource allocation. For each network entity and link, the extended system uses real traffic to evaluate the demand at several time scales from immediate to short term and long term. Using proper Application Program Interfaces (APIs), the anticipated demand can be used to control dynamic configuration, to determine ongoing network upgrades, etc. Currently, cellular network OSS does not support dynamic resource allocation. As a result, it is not suitable as an optimal solution for two key requirements from the advance standard: (i) the requirement for Self Organized Networks (SON) and (ii) the requirement to support on-line resource sharing between different operators.

Keywords- LTE-Advance, OSS/BSS, QoS provisioning, Dynamic Configuration, Network Dimensioning.

I. INTRODUCTION

According to Cisco Global Mobile Data Traffic Forecast Update, the overall mobile data traffic is expected to surpass 24.3 exabytes by 2019, a tenfold increase over 2014 [1]. Data delivered to smart phones is growing at a significantly faster rate than the revenues it generates. The process of upgrading a cellular network to overcome bottlenecks and to support Mobile Broadband is long and expensive. Planning and optimizing a network with mobile users is a big operating challenge and cellular operators are facing the huge challenge of a multi-year, multi-billion dollar investment in highly competitive market. LTE-A is the next major step in mobile radio communications, and was introduced in 3rd Generation Partnership Project (3GPP). LTE brings the concept of Self-Organizing Networks (SON) providing new challenges and opportunities for Operations Support Systems (OSS) vendors. SONs will require new policy-based OSS software capabilities to realize their full potential — and these advanced OSS features are required.

Advanced cellular network deployment in rural areas, in particular, in 3rd world countries, is typically not economic for a single cellular operator. Furthermore, most of the governments in these countries are too poor to fund a large scale deployment project and, as a result, wide geographical areas are disconnected from the Internet and from other modern communication mediums. Flexibility in regulation can be the key to solve this problematic situation. If the

regulator will allow resource sharing between multiple private cellular operators, joint rural deployment can be much more cost effective. However, resource sharing between different operators should be adaptive to the traffic patterns per operator and should be dynamic to maximize the cost effectiveness of each participating operator, yet, such important capability does not exist in today's OSSs [7].

The Dynamic Resource Block (DRB) sub-system described here extends the OSS in modeling and evaluating network traffic potential. The system processes information from several databases such as users, billing, and radius to construct an accurate stochastic network model. The model captures several timeframes – immediate, short and long. It is based on classifying users, devices, applications, services and on-location estimations (in addition to the conventional traffic parameters). Using this extended operations system, the operator can share resources with other operators, and can improve network readiness for new demanding services, evaluate and optimize the current resource allocations and the end-to-end network performance.

Additional requirements from the future OSS are described in [7], such as holistic integration between the business processes and the network operations processes. Several network planning tools are currently available. Usually, these commercial tools are stand-alone systems, used exclusively for the initial stage of the network set-up. Sophisticated planning tools provide a network set-up plan based on human population maps together with three-dimensional (3D) geographic models. However, these tools do not provide operators with any useful information regarding immediate dynamic resource allocation, network updates required in the short term for the coming weeks or months or a true understanding of long term trends (coming years). As a result, operators are forced to operate on-going network updates and long-term upgrade plans using internal guesswork. Furthermore, the existing planning tools and internal procedures cannot provide information regarding the impact of new services, new devices or new user data packages. By contrast, the Dynamic Resource Block sub-system is embedded in the network; it can answer these requirements in a simple manner and provide a true integration between the operator's OSS and its network.

Both the strengths and the weaknesses of the DRB system are direct outcomes of its multiple interfaces and the fact that it must process multiple information streams online. Fortunately, IBM has developed a platform – the IBM

InfoSphere Streams platform [2] which is fully capable of supporting all DRB implementation requirements.

This paper is organized as follows. In Section II, a full description of the DRB internal structure is given. Section III presents selected internal methods of the DRB. Finally, the paper is concluded and some future research directions are listed in Section IV.

II. INTERNAL STRUCTURE OF THE DYNAMIC RESOURCE BLOCK

In this section, we provide a full description of the proposed DRB system. Figure 1 depicts a block diagram with the main elements of the DRB. The DRB comprises eleven different types of units: a network builder module, a subscriber's profiler module, a parser module, a traffic classifier module, a statistic characterization module, a mobility module, a database for traffic processing, a network graph, three demand prediction engines, a requirement enforcer module, and an upgrade network module.

At initiation, all resources involved in the DRB should be allocated, reset, and/or introduced to each other. The network builder module uses information from the operator geographic Information System or any other operator system to build the DRB internal network graph. The network graph consists of a set of nodes N and a set of links E . Each node $n \in N$ represents a network entity such as E-NodeB (eNB), Base Station (BTS), Aggregation entity (AGG), Radio Network Controller (RNC), Serving Gateway (S-GW), etc. The node structure includes information on the represented network entity. For example, information on a BTS can include its radio capabilities (2G, 3G, 3.5G, 4G), its neighboring BTSs, its cover area, the population density in its cover area, etc. Each link $e \in E$ represents a network link. The link structure includes information on the represented link. Information on a link can include its bandwidth (BW), its protocols (for example Asynchronous Transfer Mode (ATM), Internet Protocol (IP)), its monthly rental fee, its technology (for example, E1, T1, wireless), its cost, etc.

The subscribers' profiler module analyzes the information from the operator subscribers' information system and from the operator billing system and updates the DRB internal network graph. Information on a subscriber includes its mobile device, its "home cell" (the cell/BTS in which the subscriber is usually present in its cover area), its average data consumption, its tariff deal, information on its mobility pattern, etc.

The parser module parses messages and information from the traffic capture file provided by a probe or a monitoring tool and/or it can parse traffic statistics arriving from the operator network entities, such as backhaul switches, aggregation nodes, etc. It converts each message into a unified form and sends it to the traffic classifier module. The classifier module classifies each message according to its corresponding subscriber/link etc. Then, it stores the traffic record in the real traffic database.

The statistic characterization module statistically analyzes the real traffic in the real traffic database according to the link/entity association. For each network entity/link in the DRB internal network graph, the statistic characterization

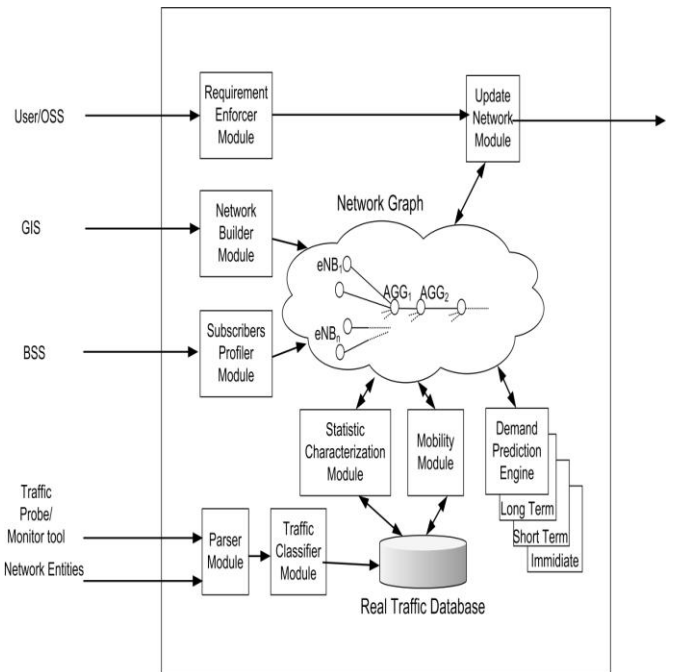


Figure 1. The internal structure of the Dynamic Resource Block and its interfaces

module calculates the packet arrival process and the packet service time. This information is stored in the DRB internal network graph and is used by the demand prediction engines. The mobility module calculates estimates of mobility parameters on the traffic records in the real traffic database. These parameters are stored in the DRB internal network graph and are used by the demand prediction engines.

The demand prediction engines run demand prediction algorithms to estimate the traffic demands on each entity/link in the DRB internal network graph. Each engine uses the information stored in the entity/link to evaluate the current demand and the potential demand at the next time point. Different demand prediction algorithms are used for different time scales such as the long term algorithm, the short term algorithm, and the immediate demand prediction algorithm. The long term demand prediction engine provides prediction of the traffic demands in the next year, next six months, etc. The output can be used to calculate the operator's required network upgrade plan. The short term demand prediction engine provides predictions of the traffic demands in the next week, next day, etc. Its output can be used to calculate an accurate order of rental backhaul lines, for example. The immediate demand prediction engine provides a prediction of the traffic demands in the next few minutes/seconds. This can be used to request/guarantee resources on a dynamic shared medium such as a backhaul link in a 4G network.

The requirement enforcer module translates the user's instructions and the operator network management system instructions into traffic rules in the desired network upgrade plan. Examples of traffic rules are maximum delay on a link/entity, minimum BW, etc. The upgrade network module combines the traffic demand prediction with the traffic rules and calculates the desired/recommended network upgrade

plan (in the long term case), the required link rental order (in the short term case) or the resource allocation requests (in the immediate case). In addition, it calculates a cost effective upgrade using information from the DRB internal network graph, such as link/entity upgrade costs.

Figure 2 depicts a block diagram with the main elements of the network builder module. A network builder module is comprised of three different types of sub-modules: the Geographic Information System (GIS) report interpreter, the node generator and the link generator. The GIS report interpreter is responsible for the analysis of the GIS file/report from the operator GIS. The node generator is responsible for the creation of new nodes in the network graph that correspond to the operator network entities. The link generator is responsible for the creation of new links in

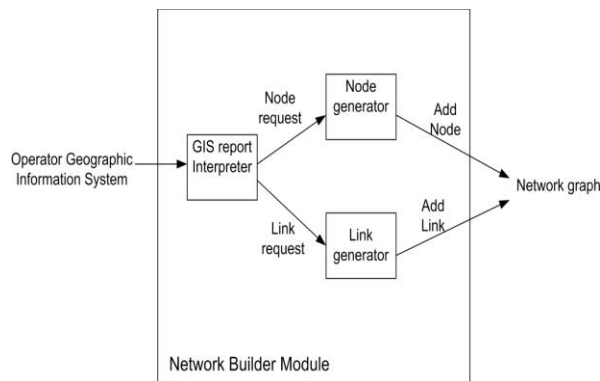


Figure 2. The network builder module

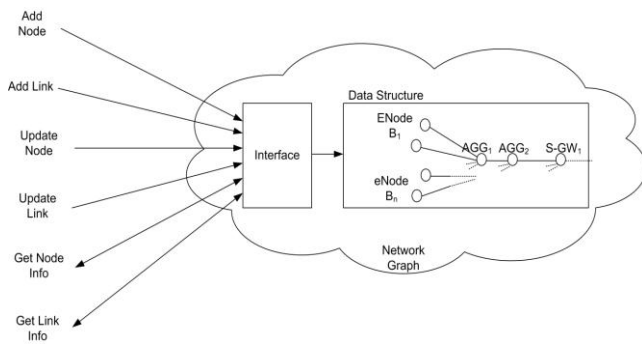


Figure 3. The network graph

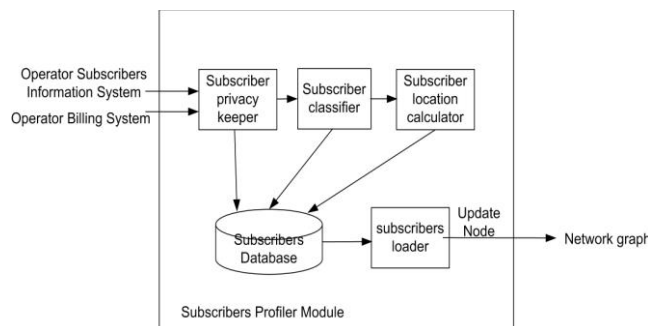


Figure 4. The subscribers' profiler module

the network graph that correspond to the operator's network links. Figure 3 presents a simplified block diagram with the key elements for the network graph. A network graph is comprised of an interface and a data structure. The network graph interface handles the API to network graph data structure. A possible API list includes Add Node, Add Link, Update Node, Update Link, Get Node Info, and Get Link Info.

The network graph data structure can be implemented by any data structure that can represent node structure, link structure and describe their connectivity. Examples of such structure are a linked-list/array of node structures, together with a matrix for the links, where a matrix cell[i,j] represents the link between node i and node j. In case of multiple links between network elements, each matrix cell can be implemented by a linked-list of link structure.

Figure 4 plots a simplified block diagram with the main elements of a subscriber's profiler module. A subscribers' profiler module is comprised of five different types of sub-modules: the subscriber privacy keeper, the subscriber classifier, the subscriber location calculator, the subscriber database and the subscriber loader.

The subscriber privacy keeper is responsible for hiding the subscriber's identity and creating a new identity to be used by the DRB. The new identity can be a one-way hash on the real subscriber's identity (such as full name, Mobile identification number (IMSI), etc.). Once a new identity is created, the subscriber privacy keeper can add the new subscriber's record to the subscriber database (which can be implemented as an internal or external database). The subscriber classifier identifies specific information such as the device category, registered services, payment deal, voice/data average/min/max usage, etc. According to the subscriber's information, the subscriber classifier can assign the subscriber to a specific potential load class. The subscriber information is stored in the subscriber record at the database.

The subscriber location calculator calculates the subscriber location probability function from the subscriber's call records. Each call record includes the cell ID of the cell in which the call was originally generated. This histogram can be stored in the subscriber's record in the database. The subscriber loader collects the information on the potential load generated by each subscriber and locates the right indicator in the cells according to each subscriber's location probability function.

III. SELECTED INTERNAL PROCEDURES FOR THE DYNAMIC RESOURCE BLOCK

The LongTermTrafficPrediction method below illustrates the main steps for a long term traffic demand prediction procedure executed by the DRB. This method can be executed, per each planned upgrade phase scheduled at the operator network, for example.

```

LongTermTrafficPrediction()
Begin
1: System Initialization
2: Insert and analyze network
  information and build the network
  graph
3: Insert and analyze subscriber's
  info and update the network graph
4: Insert and analyze Real traffic
  and update the network graph
5: Run Long Term demand prediction
  algorithm
6: Calculate network upgrade
End
    
```

In step 2, the DRB receives information on the operator network, including entities and links from the operator information system, such as GIS. The information is processed by the network builder module and used to build the DRB internal network graph (method BuildNetworkGraph below). In step 3, the DRB receives the operator information on the subscribers from the operator subscriber information system and the subscriber profiler module can analyze it and update the DRB internal network graph (method SubscribersAnalysis below).

Next, the DRB receives files provided by a probe or a monitoring tool, capturing real traffic. In step 4, the parser module parses messages and information from these files and converts each message into a unified form and sends it to the traffic classifier module. The classifier module classifies each message according to its corresponding subscriber/link. Then, it stores the traffic record in the real traffic database. Afterwards both the statistic characterization module and the mobility module can estimate the real traffic and update the DRB internal network graph accordingly.

In step 5, the demand prediction engine runs the long term demand prediction algorithm to estimate the traffic demand on each entity/ link in the DRB internal network graph using the information stored in the entity/ link to evaluate the current demand and the potential demand in the next time point (for example, next year). The output of the prediction engine can be used together with pre-defined traffic rules to calculate the operator required network upgrade plan by the upgrade network module in step 6.

Clearly, using the short term demand prediction algorithm, a similar method can be applied for short term traffic prediction. Such a method can be executed, each month/week to calculate the rental backhaul link order or on-going specific upgrades for example.

Figure 5 depicts a wireless backhaul network using the DRB. Each cell site can use the DRB to calculate the next resource requests for bandwidth allocation in the next time slot on the wireless backhaul link. The requests are sent to the backhaul aggregation site which dynamically assigns the bandwidth to the cell sites. The DRB can be implemented as an integrated module within the cell site system or it can be implemented as a separate system with the right interfaces to the cell site system.

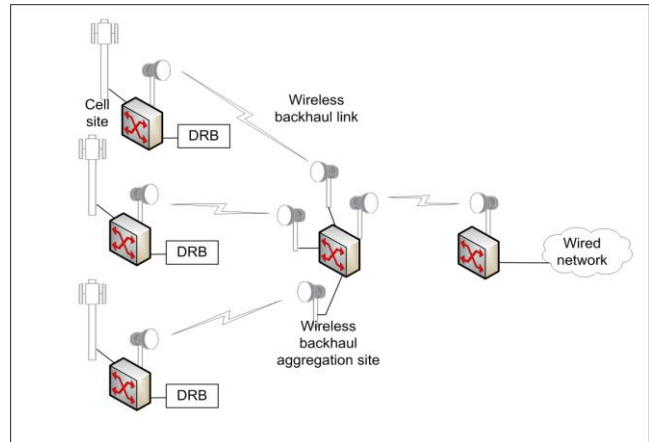


Figure 5. Wireless backhaul network using the immediate prediction capabilities of the DRB.

Possible bandwidth negotiation between a cell site and an aggregation site can be facilitated by employing the following control message-aided process (using a control messages defined in a Multipoint Control Protocol (MPCP) as describe in [9], for example). Each cell site requests its next transmission by piggybacking a REPORT message at the end of its current time slot. Instead of reporting the actually buffered data, the REPORT message includes a prediction of the data that arrived during the waiting time provided by the DRB, so as to reduce the delay over the upstream data transmission. The aggregation site can make a bandwidth allocation decision based on these reports. A GATE message can then be replied downstream to the cell site containing the information of time slot start time and time slot length. The destined cell site can update its local registers accordingly, and transmit data from the time slot start time in the time slot length.

The method ImmediateTrafficPrediction below illustrates the key steps for an immediate traffic demand prediction executed by the DRB. This method can be executed in a real time environment to calculate a bandwidth request on a dynamic link, for example. It can be implemented in a distributed manner in a backhaul system as illustrated in Figure 5.

```

ImmediateTrafficPrediction()
Begin
1: Parse Traffic REPORT information
2: Classify report to the network
  entity node
3: Calculate the traffic demand in
  the next period
4: Generate the REQUEST message
End
    
```

In step 1, a traffic report arrives from a network entity (for example, a cell site). The traffic report can include information on the entity's current queues state, number of byte/bits waiting for transmission, number of active users, etc. The report is received by the parser module which parses

the information. The traffic classifier module classifies the information and associates it with the corresponding network entity node in the real traffic database in step 2. Note that if the DRB is integrated within the cell site, then steps 1 and 2 are no longer required.

In step 3, the immediate demand prediction engine calculates the traffic demand in the next time slot and updates the corresponding network entity node in the network graph. It can use the fact that the network traffic is self-similar [3], which implies that the actual network traffic exhibits long-range dependence (LRD), and the burstiness of the traffic does not decrease with the time scale from which the traffic is observed or with the amount of multiplexing that occurs at a node. Owing to self-similarity, the correlation in network traffic does not decay rapidly, and traffic is correlated over time slots. Given its advantages of low computational complexity, fast convergence, and no prior knowledge of the traffic statistics, the Linear Predictor (LP) is a practical tool to conduct such real-time traffic prediction [4]-[6]. Finally, in step 5, the resource request message can be generated by the update network module.

The BuildNetworkGraph method below illustrates the steps for processing the received information on the operator network. This method can be executed by the network builder module (see Figure 2) to build the DRB internal network graph (see Figure 3). The GIS report interpreter receives the GIS file/report. It processes the information and identifies the network elements and links. For each identified network entity or link, the GIS file/report includes an introduction record with the appropriate information. For example, information on a base station can include its radio capabilities (2G, 3G, 3.5G, 4G), its neighboring base stations, its cover area, the population density in its cover area, etc. Information on a link can include its bandwidth (BW), its protocols (for example ATM, IP), its monthly rental fee, its technology (for example, E1, T1, fiber), its upgrade cost, etc.

```
BuildNetworkGraph()
Begin
1: Receive a GIS file
   While (record) {
2:   Analyze next record
3:   Case 'Node': create new node
4:   Case 'Link': create new Link
   }
5: Introduce network graph to other
   modules
End
```

The SubscribersAnalysis method below illustrates the steps for processing the received data and analysis of the operator subscribers' information from the operator Subscribers' Information System (SIS) and operator Billing Systems (BS). This method can be executed by the subscriber profiler module (see Figure 5) to analyze the subscribers' influence on the network demand.

```
SubscribersAnalysis()
Begin
1: Receive a SIS/BS file
   While (record) {
2:   Analyze next record
3:   Hash subscriber identity
4:   Classify subscriber
5:   Calculate subscriber location
     probability }
6: Update subscribers load on the
   network graph
End
```

In step 1, the subscriber privacy keeper receives the SIS/BS file/report from the operator SIS/BS. It analyzes each record (step 2) and calculates the subscriber's new identity from a one-way hash function. Then, it creates a new record in the subscriber database. In step 4, the subscriber record is further processed by the subscriber classifier. Identifying the subscriber device, services, usage patterns and billing information can be used to calculate the subscriber class with respect to its potential demand on the operator's network resources. In addition, in step 5, the subscriber location calculator uses the subscriber call record to estimate the user location probability function; that is, to calculate the probability that the subscriber will generate a call from each cell. This location probability is stored in the subscriber's record in the subscriber database. Finally, in step 6, the subscriber loader sub-module uses both the subscriber class and its location probability function to relate the subscriber's potential demand with the right network elements and links in the network graph.

IV. CONCLUSIONS AND FUTURE WORK

This paper describes a new system and methods to fully integrate the operator OSS and its network. By implementing this system, cellular operators can dynamically share resources between different operators; they can improve the quality of service provisioning, optimize resource allocation and network equipment purchases. The system makes traffic demand predictions on the network links and entities at multiple time scales. It is based on classifying users, devices, applications, services and on location estimates (in addition to conventional traffic parameters).

The advantages of the suggested Dynamic Resource Block extension to the OSS are: it provides the required missing functionality, it is very simple, and it is general and not limited to a specific network technology and/or any specific OSS.

The proposed solution is a suggestion for a real implementation for extending current OSS to provide what we believe is missing important functionality. As such, it is extremely difficult to evaluate its performance without at least implementing a prototype. However, in any system turning from static resource allocation scheme into dynamic one is likely to perform better. Clearly implementation of such a system is beyond the scope of academic research. Nevertheless, a possible simple and elegant solution can be implemented using the IBM InfoSphere Streams platform [2]

which is fully capable of supporting all DRB implementation requirements.

REFERENCES

- [1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019, available at <http://www.cisco.com/> last accessed May 2015.
- [2] R. Rea and K. Mamidipaka, "InfoSphere Streams: Enabling in-motion data analytics", <http://www.ibm.com/software/data/infosphere/streams/> last accessed May 2015.
- [3] W. Willinger, M. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Trans. Networking* 5, 1997, pp. 71–86.
- [4] S. String and H. Sirisena, "The influence of long-range dependence on traffic prediction," in *Proceedings of the IEEE International Conference on Communications (ICC'2001)*, 2001, pp. 1000–1005.
- [5] G. Gripenberg and I. Norros, "On the prediction of fractional Brownian motion," *J. Appl. Probab.* 33, 1996, pp. 400–410.
- [6] I. Norros, "On the use of fractional Brownian motion in the theory of connectionless networks," *IEEE J. Sel. Areas Commun.* 13, 1995, pp. 953–962.
- [7] M. Toy, "Future Directions in Cable Networks, Services and Management", chapter 6 in the book *Cable Networks, Services, and Management*. M. Toy editor, Vol. 13. John Wiley & Sons, 2015. ISBN: 978-1-118-83759-7, pp. 246-340.
- [8] J. Friman, L. Angelin, E. Drake, and M. Agarwal, "Next generation OSS/BSS architecture." *Ericsson Review*, 2013.
- [9] C. Diot, D. Walid, and C. Jon, "Multipoint communication: a survey of protocols, functions, and mechanisms." *IEEE Journal on Selected Areas in Communications*, vol 15.3, 1997, pp. 277-290.