DEVS Based Modeling of Shared Segmented Upload in LTE-A Mobile Networks

Misagh Tavanpour, Jan Mikhail, Gabriel Wainer, Mohammad Moallemi
Department of Systems and Computer Engineering, Carleton University
1125 Colonel By Dr., Ottawa, ON, K1S 5B6
{misagh, janmikhail, moallemi, gwainer}@sce.carleton.ca

Gary Boudreau, Ronald Casselman
Ericsson Canada
349 Terry Fox Dr., Ottawa, ON, K2K 2V6
{gary.boudreau, ronald.casselman}@ericsson.com

ABSTRACT

In recent years, there have been important efforts focusing on providing suitable services for mobile networks users and on supporting their increasing demands. One promising standard is called Long Term Evolution Advanced (LTE-A). LTE-A uses different techniques (such as Coordinated Multipoint - CoMP) to deal with bottlenecks to improve performance, in particular for users at the edge of the cells. Here, we show a simulation study focused on the Shared Segmented Upload (SSU) algorithm, which deals with these problems. We use the Discrete Event System Specification (DEVS) formalism to model two different approaches. The simulations show how the SSU algorithm improves services for cell-edge users. We show how to define these kinds of applications using a formal framework like DEVS.

Author Keywords

SSU; CoMP; LTE-Advanced; DEVS;

INTRODUCTION

These days, almost half of the world population use mobile networks (and almost one billion of these users were attracted by mobile networks operators in the last four years; the number of mobile connections reached almost 7 billion) [1]. The user demand for bandwidth is always increasing. Hence, providing high quality services is difficult, in particular because service providers want to guaranty uniform service over the covered geographical areas. To fulfill these requirements, service providers constantly research new protocols to improve the quality of service in both the User Equipment (UE) and the network.

To deal with these issues in Fourth Generation (4G) mobile networks, 3rd Generation Partnership Project (3GPP) introduced the Long Term Evolution standard (LTE) and more recently LTE-Advanced (LTE-A) [2, 3]. One of the main goals of the LTE-A standard is providing high data rate services for users regardless of their geographical location within the coverage area. To achieve high data rates for cell-edge users, a number of challenges need to be addressed. These challenges include low signal power due to the large distance between the users and their serving base stations, as well as a higher interference ratio near the cell borders. These problems decrease the effective signal-to-interference ratio, leading to lower data rate. To deal with these problems, LTE-A employs a number of technologies including Coordinated Multi Point (CoMP), considered as a key technology to enhance the performance of cell-edge users [2, 3]. CoMP uses a set of BSs, called the coordination set, that work together to reduce interference and enhance the received signal strength. This form of coordination is beneficial to UEs located close to the cell's edge.

We discuss the effects of using CoMP in urban mobile networks by implementing a novel algorithm called Shared Segmented Upload (SSU), and comparing its results with a conventional non-cooperative algorithm [4]. SSU is designed for uploading large data files from a UE to multiple BSs in a distributed CoMP architecture [4].

There are different computer networks simulators widely used for networking analysis, such as NS2, NS3 and OPNET. These simulators have some problems, including patching new extensions, poor tracing performance, large amount of memory and processing time required for large simulations, limited scalability, etc [5, 6, 7, 8]. Likewise, it is difficult to integrate the networking models with other models (for instance, models of traffic of pedestrians holding user equipment). In this paper, we are not going to compare different simulation and modeling techniques. Rather, we want to study the performance of SSU algorithm under urban area setting and we want to introduce a flexible M&S technique that can be used in cellular networks. To do so, the two models (SSU and non-cooperative) were implemented using the Discrete Event System Specification (DEVS) formalism. The hierarchical nature DEVS allowed us to capture precise information from different levels of the model. We measured the average file upload time and the average data rate for SSU and a non-cooperative method. The simulation results show that SSU provides services that are more consistent as the users move away from the cell center. The results reveal that with the non-cooperative algorithm, cell-edge users required more time to upload the same amount of data.

BACKGROUND

In mobile networks, there are number of transmission barriers that reduce the overall system performance. In particular, some of these barriers, such as Inter-Cell Interference (ICI), degrade the performance of cell-edge UEs and prevent the network performance from being close to their theoretical rates. To mitigate such problems and provide suitable services for users, the LTE-A standard uses different technologies such as CoMP [2, 3].

Figure 1 illustrates a simple geographical area that is divided in to a number of cells, and each cell has one base station (BS), located at the center of the cell. When a UE is within a reasonable distance from its serving BS (the BS in the same cell, which is located in the cell center), the received signal is strong enough to provide a high data rate communication. However, this situation changes when the UE moves away from the cell center and towards the cell's edge. Due to the increased distance between the UE and its serving BS, the signal power decreases. In addition, when the UE is located close to the cell's edges, the UE receives other interfering signals from the BSs in the neighbouring cells, which reduce the signal strength.

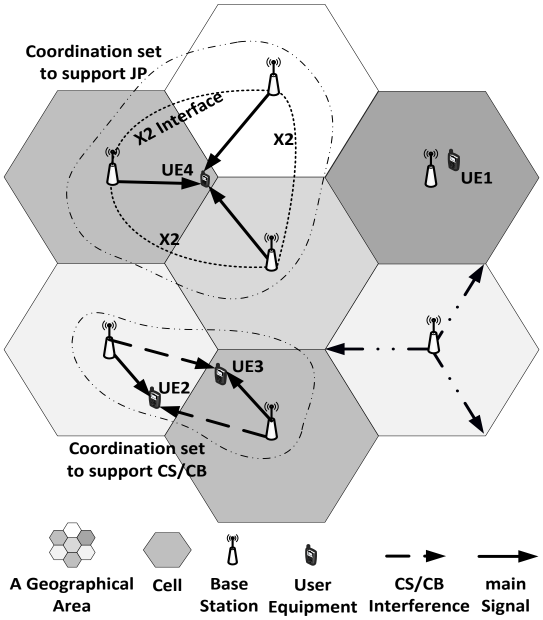


Figure 1. CS/CB and JP transmissions in LTE-Advanced.

CoMP was introduced by 3GPP in Release 11 as a key feature to increase the performance of cell-edge users [2, 3]. Essentially, CoMP employs the BSs in a coordinated manner. These BSs establish a coordination set, which attempts to reduce the interfering signal levels as well as increase the desired signal level [4]. As seen in Figure 1, neighbouring BSs can exchange information and data through the high-speed 3GPP standard interfaces called X2 [2, 3].

There are two different approaches for CoMP based on the way the data and scheduling information are made available at the BSs in a coordination set: Coordinated scheduling/Beamforming (CS/CB), and Joint Processing (JP) [4]. Also, there are two architectures for implementing CoMP in LTE-A, depending on how scheduling decisions are made: centralized and distributed [4]. In centralized CoMP there is central entity where all the UEs' data and channel information is available. The central unit is responsible for the scheduling operations. In the distributed CoMP architecture, the UEs feed back their channel state information to their serving BS, which in turn, shares this information with other BSs in the coordination set. Each BS in the coordination set performs the scheduling operations independently.

The SSU algorithms is used to upload user data files to a set of coordinated BSs using distributed CoMP, and it has a few similarities with the BitTorrent peer-to-peer protocol [4, 9]. BitTorrent is used to speed up the download of large files by allowing users to join a swarm of hosts to download and upload from each other simultaneously. This method increases the download speed compared to traditional methods, in which users download a file from a single server. SSU employs a similar approach to that used in BitTorrent to improve data upload from one UE to a set of coordinating BSs. When a cell-edge UE wants to upload a large file, it divides the file into a number of small segments. Then, the UE uploads each of these segments to the BSs in the coordination set. Finally, the segments are gathered by the serving BS, and joined to recreate the data file. In this approach, a UE transfers segments of the file to multiple BSs in the coordination set independently. As a result, we can expect a dynamic adjustment of the data flow in which BSs with better communication channels receive more segments from the UE. This results in a faster and more efficient upload for cell-edge UEs [4].

There have been numerous investigations of new standards and protocols to fulfil the users' requirements and increase their performance. Testing the effectiveness of the proposed methods and investigating different aspects of their operation is easier when using Modeling and Simulation (M&S): one can build a model for a network, simulate it, and test and evaluate the proposed approaches under different test configurations and scenarios. In [10], the authors mentioned that testing is one of the recurring problems in LTE networks. They used NS2 to build an LTE/SAE model to test different parameters. In [11], the authors used OPNET as their platform to study discontinuous reception in LTE networks. They show that this approach leads to better battery power usage of the UEs with a potential increase in latency. In [12], the authors used NS3 to implement carrier aggregation to study scalable video multicast to LTE-A user groups.

Here, we use DEVS [13] to model the mobile network as well as SSU and the conventional non-cooperative algorithms. DEVS theory is a methodology to represent models and it provides an abstract description of the system of interest. The Coupled components maintain the hierarchical structure of the system, while each Atomic component represents a behavior of a part of the system. Atomic components are the basic building blocks of the system, which are composed of I/O ports and a finite state timed automaton representing the behavior of the model [13]. In particular, we used the CD++ toolkit as the platform to implement the

DEVS models. This toolkit provides a built-in specification language to implement the DEVS Atomic models. Users can implement model definitions using C++. A Model file is used to define the hierarchical structure of the Coupled models and to initialize the atomic models' parameters [2, 13].

The SSU Algorithm

One of the main objectives for service providers is to deliver high data rates for their users. However, as discussed in the previous sections, this is challenging for cell-edge users. Therefore, uploading large files to the network is a slow process if the users are located near the cell's edge. SSU tries to improve the upload speed by allowing UEs to spread their data transfer over multiple links to a number of BSs, rather than only communicating with the serving BS. The non-serving BSs send the pieces they receive from the UE to the serving BS through the X2 backhaul links.

Before the SSU upload process begins, the data file that the UE intends to upload is fragmented into a number of pieces. Based on the file size, the number of BSs in the coordination set, and the conditions of the uplink channels, the size of the piece is defined. Before uploading file pieces to the network, the UE creates and uploads a file descriptor, (*MetaInfo*), which includes the size of the actual data file (in bytes), the number of pieces, as well as other information describing the data file. The UE can quickly transfer the MetaInfo to its serving BS due to its small size [4]. Figure 2 shows the various steps of the SSU upload process [4].

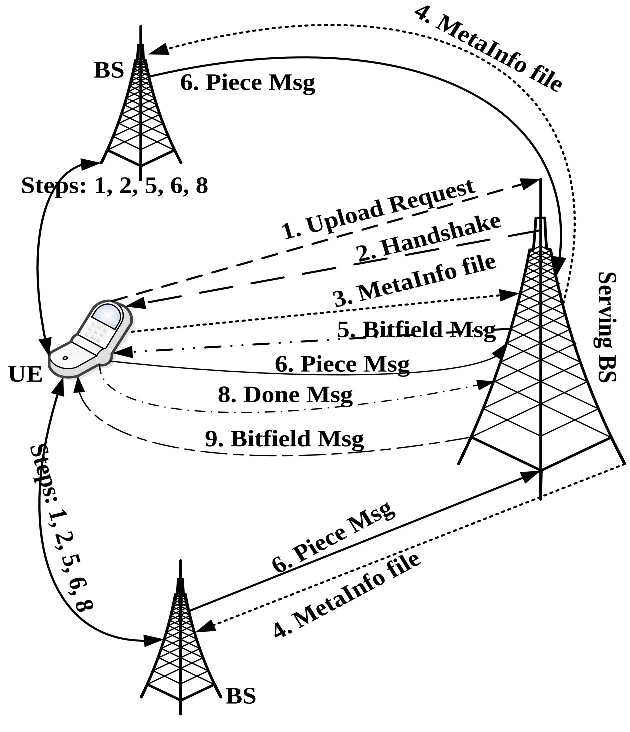


Figure 2. Shared Segmented Upload steps [4].

The SSU algorithm steps are as follows:

1. The UE sends an *Upload Request* message to all BSs in its coordination set.
2. The BSs reply with a *Handshake* message.
3. The UE sends the *MetaInfo* file to the serving BS.
4. The Serving BS forwards the *MetaInfo* file to other BSs in the coordination set.
5. The BSs use the *Bitfield* message to acknowledge the receipt of this file. The *Bitfield* message also tells the UE about the pieces available on the BSs.
6. The UE starts sending the pieces (*Piece* message).
7. Once the BSs receive the pieces, they send them to the serving BS through the backhaul.
8. Once all the pieces are transferred to the BSs in the coordination set, the UE sends a *Done* message.
9. The serving BS acknowledges the receipts of all the pieces by sending a *Bitfield* message to the UE.
10. If the *Bitfield* message is not received, the UE continues sending the missing pieces and repeats steps 8 through 10.

MODELING OF MOBILE NETWORK IN DEVS

We designed a DEVS model to examine the performance of a mobile network with multiple upload protocols, including the proposed SSU algorithm. The structure of this model is shown in Figure 3 [4].

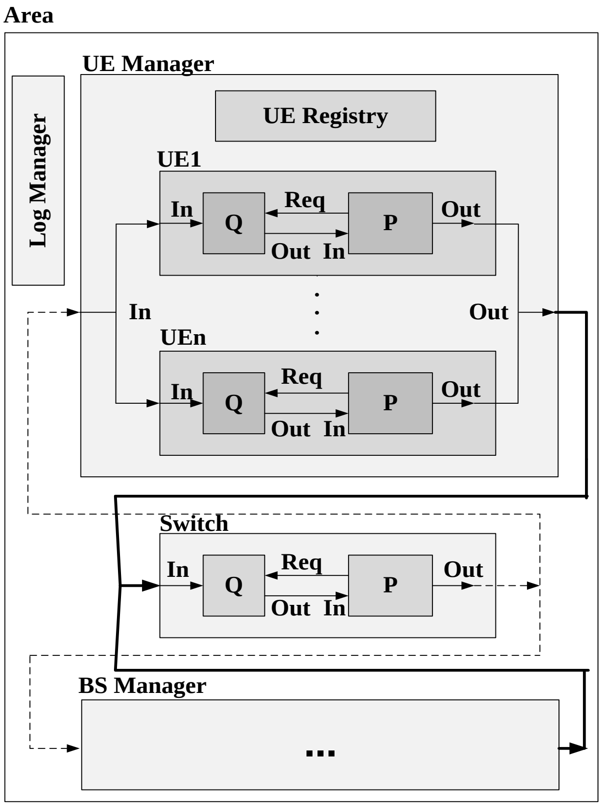


Figure 3. Simplified DEVS model hierarchy for mobile network model (Q: Queue, P: Processor) [4].

The top-level model is called *Area*, and consists of one Atomic model (*LogManager*) and other Coupled models: *Switch*, *UEManager*, and *BSManager* [4]. *UEManager* contains a registry model responsible for coordinating and updating UE variables, along with a number of UE models. UE variables include its position, speed, direction, links to BSs, and transmission parameters. *BSManager* has a similar structure but handles all the operations for the Area's BSs. Each BS and UE model is composed of two Atomic models: *Queue* and *Processor*. All the *Queue* Atomic models operate in the same manner: they accept incoming messages with a valid destination address, store them in a list, and release the message with the least delay time to the processor when it is ready to process a new message. In this way, *Queue* is responsible for modeling network propagation delays. On the other hand, the *Processor* model operates based on the definition of the algorithm being modeled. The *Switch* Coupled model facilitates the communications between the UEs and BSs. Instead of defining interconnections between each pair of models (which can grow quickly as the number of UEs and BSs increase), the *Switch* is used to receive all the sent messages, and to broadcast them to all the other models. The models can then select their messages based on the destination address defined in each message. This helps simplifying the model definition. Finally, the *LogManager* is used to collect statistics and log simulation events into log files.

Figure 4 shows a simplified segment of a model file. The model file is used to define a DEVS Coupled model and its hierarchical structure using the CD++ tool [22]. In the model file, Coupled models list their components and links between them, and Atomic models list their parameters.

```
[top]
components: logManager@LogManager  switch
components: UEmanager BSmanager
...
Link: out@switch in@UEmanager
...
[logManager]
areaConfiguration: rural
bsCounter: 16ueCounter: 64
...
[UEmanager]
components: UEregistry@UERegistry UE1 UE2...
UE64
...
[UEregistry]
areaConfiguration: rural
...
[UE1]
components: UE1Queue@UEQueue
UE1Processor@Node
...
[UE1Processor]
UEId: 1     currentX: 23729  currentY: 14210
...
```

Figure 4. Simplified model file of an area [4].

Each Atomic model described above is defined in a C++ class. Aside from these, a number of other passive classes have been defined. Figure 5 below depicts a simplified UML class diagram of the model. The *BS* class represents a *BSProcessor* by its id, type, coordinates, height from ground, height from the average rooftop, carrier frequency, transmission power, antenna gain, and a list of connections with the UEs in range. The *Node* class characterizes a *UEProcessor* using its id, current coordinates, destination coordinates, speed, transmission power, antenna gain, and a list of connections to the in-range BSs. The *UELink* and *BSLink* classes are linked lists held by every *BS* and *UE*, and it contains the uplink and downlink parameters, respectively. These metrics include the link separation distance, path loss, and the received power. *UELink* and *BSLink* have methods to compute these metrics in an urban area setting, given the properties of the link's sender and receiver.

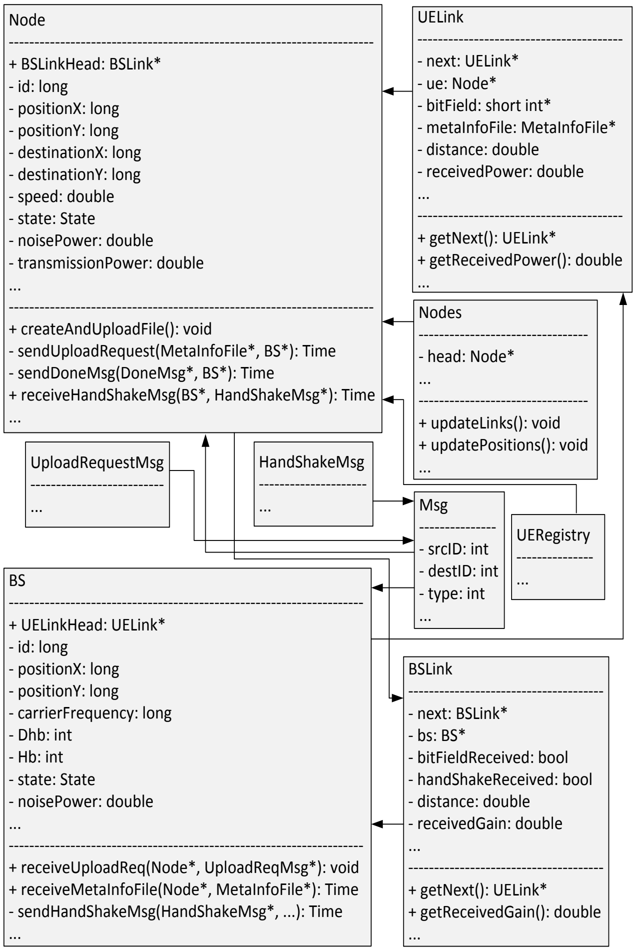
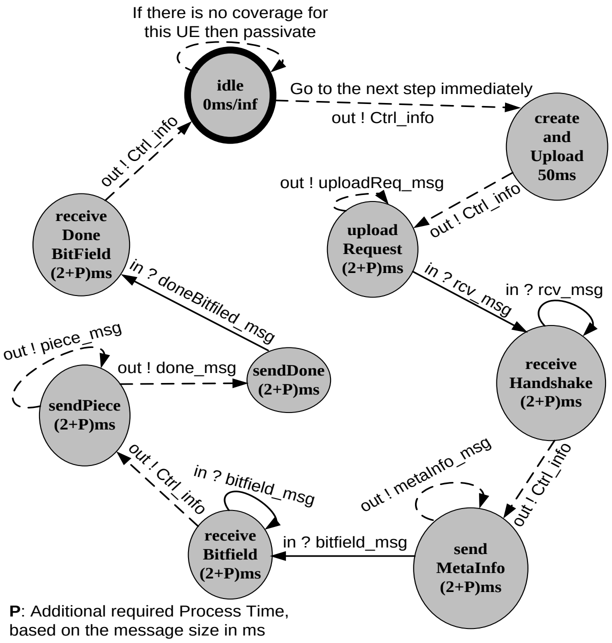


Figure 5. Simplified class diagram of the model.

In each simulation run, each UE only uploads a single file. The state diagram of UEProcessor is shown in Figure 6. *UEProcessor* has 9 states: *Idle*, *CreateAndUpload*, *UploadRequest*, *RcvHandshake*, *SendMetaInfo*, *RcvBitField*, *SendPiece*, *SendDone*, and *RcvDoneBitField*. On the other hand, a BS receives and sends messages to multiple BSs and UEs during the upload processes and therefore, its state transitions are more complex. To simplify its implementation, a *BSProcessor* only cycles through four states every time an external message is received: *Idle*, *Receive*, *Process*, and *Send*. The operation of the *BSProcessor* depends on the state of the communication link between the BS and the sender of the message. The *UELink* class holds this state.



P: Additional required Process Time, based on the message size in ms

Figure 6. UEProcessor DEVS graph.

In order to evaluate the performance of the SSU algorithm, another conventional non-cooperative algorithm was implemented: each UE only interacts with its serving BS instead of all the BSs in its coordination set. A UE starts the upload with a request message, similar to the one employed in SSU. Once the BS acknowledges the request, the UE starts uploading the data file in variable sized packets, which depend on the available bandwidth and data rate for the link. The upload processes ends when the UE sends the BS a Done message following the last data packet. It is worth mentioning that, to further evaluate the performance of the SSU algorithm, we require implementing other LTE-A CoMP algorithms as well.

SIMULATION CASE STUDIES

In [4], we introduced the SSU algorithm to solve the problem of user upload in LTE-A networks. We showed a rural area setting as the first scenario to evaluate the performance of the proposed algorithm. We also need to test the functionality of the SSU algorithm (and measure parameters such as latency) under an urban area setting. In this section, we discuss how we test correctness and performance of the SSU algorithm in an urban area scenario, whose conditions are completely different, compared to those of rural areas.

As discussed earlier, the main objective of using CoMP in LTE-A networks is to provide consistent service for the users regardless of their location. Therefore, our first group of simulations focuses on the users' performance as a function of their distance from their serving BS. The UEs are distributed randomly in a narrow range of distances from the cell center (where their serving BS is located). In each iteration, we increase this distance, allowing us to study the effects on the performance of the SSU algorithm [4]. The second group of simulations investigate only the cell-edge users' performance. To do so, the UEs are distributed randomly close to the edge of the cells. In each simulation, we injected a different number of UEs in the network area.

The configurations that are required to run both scenarios and the formulas to compute parameters for an urban area setting are extracted from [14]. Two carrier frequencies are considered: 900 and 2000 MHz. The maximum UE and BS transmission powers are 21 and 43 dBm respectively. The rest of the parameters are listed in Table 1 [14].

Parameter	Value	
Frequency	900 MHz	2000 MHz
BS Antenna Gain	12 dB	15 dB
Transmission bandwidth	5 MHz	
Noise Density	-174 dBm/Hz	
MCL	70 dB	
BS Antenna Height above rooftop (Dhb)	15 m	
BS Antenna Height above ground (Hb)	45 m	
LogF	10dB	
File size	[0.5MB, 64MB]	
Maximum BS power	43	
Maximum power per DL traffic channel	30 dBm	
Minimum BS power per user	15 dBm	
BS Noise figure	5 dB	
Maximum UE power	21 dBm	
Minimum UE power	-50 dBm	
UE Noise figure	9 dB	

Table 1. Key simulation parameters.

The available data rate for the UEs at each communication channel can be calculated based on the Eq 1:

$$data\,rate = B\ \log_2(1 + \frac{R_{pwr}}{N_0 \times B}) \qquad Eq\ 1$$

where B and N_0 are the transmission bandwidth and the noise variance. R_{pwr} is the received signal power, measured as in Eq 2 (in which T_{pwr} is the transmitted signal power, T_{Gain} is the transmitter antenna gain, R_{Gain} is the receiver antenna gain and MCL is the minimum coupling loss).

$$R_{pwr} = T_{pwr} - Max(pathloss – T_{Gain} - R_{Gain}, \mathrm{MCL}) \qquad Eq\ 2$$

As seen in Eq 2, in order to calculate R_{pwr} we need to know the value of $pathloss$ first. Eq 3 shows that this parameter is measured based on the log-normally distributed shadowing with standard deviation of 10dB ($LogF$) and macro cell propagation model for urban area (L_{urban}).

$$pathloss = L_{urban} + LogF \qquad Eq\ 3$$

The macro cell propagation model for urban area, which is denoted as L_{urban}, is calculated using Eq 4:

$$L_{urban} = (40 * (1 - (4.10^{-3} * Dhb)) * log_{10} R) - (18 * log_{10} Dhb) + (21 * log_{10} f) + 80dB \qquad Eq\ 4$$

where R, f and Dhb are the BS-UE separation (km), the carrier frequency (MHz) and the BS antenna height (m).

The *UERegistry* Atomic model takes care of the periodical update of the UEs' positions in the covered area. This can be done based on the UEs' locations, their random destinations and their speed. This process updates the macro cell propagation model of the links between each pair of BSs and UEs as well. Consequently, we can use the updated propagation model to calculate the received signal power at the receiver's side. Finally, the available data rate at the link between UEs and BSs can be calculated based on Eq 1. Besides these parameters, another study is required to investigate the influence of the SSU algorithm on the network backhaul to provide a more complete overall evaluation of the performance of the SSU algorithm.

In the first set of simulations, we used 17 BSs to provide radio coverage over a geographical area of 2800 x 3000 m. There are 64 active UEs and each of them uploads one file during the simulation. In each simulation, the UEs are located at a predefined distance range from their serving BSs. The width of this distance range in which UEs are located initially is 50 m. This means that in the first simulation the UEs are within the first 50 m around their serving BSs, and in the second iteration, they are located between 50 and 100 m from the serving BS, and so on. This way, we are able to study the effects of upload algorithms on the UEs performance while the UEs' distance from their serving BSs increases. It is worth to mention that each of the simulations continues until all the UEs complete their upload.

In simulations where UEs use the SSU algorithm to upload their data, they solely use their serving BS to upload their file while they are within a reasonable distance from their serving BS. As this distance increases, they may use a set of BSs to upload their file. This trend for those simulations in which UEs use the traditional non-cooperative algorithm is a bit different. In this kind of simulations, regardless of the UEs' distance from their serving BS, they just communicate with their serving BS.

In Figures 7-12, the horizontal axis shows the average UEs distance from their serving BSs. Figure 7 and Figure 8 show the average number of BSs that each UE communicates with during the uploading process for urban configuration with 900 MHz and 2000 MHz as carrier frequency. It is clear that for the standard scenarios (those ones that UEs use a traditional non-cooperative algorithm) there is just one BS that UEs communicate with all the time. In case of the SSU algorithm, as the UE distance from serving BS increases there is potential for the UEs to communicate with more than one BS during the upload process.

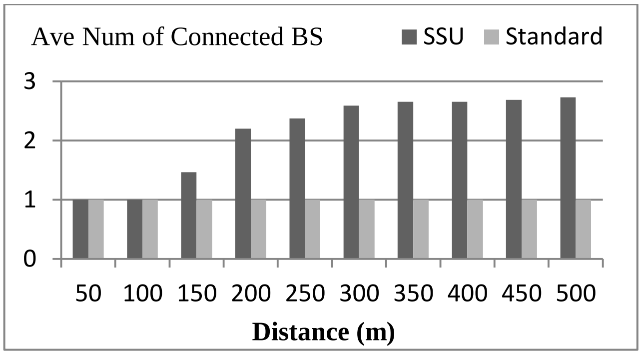


Figure 7. Average number of connected BSs (900 MHz)

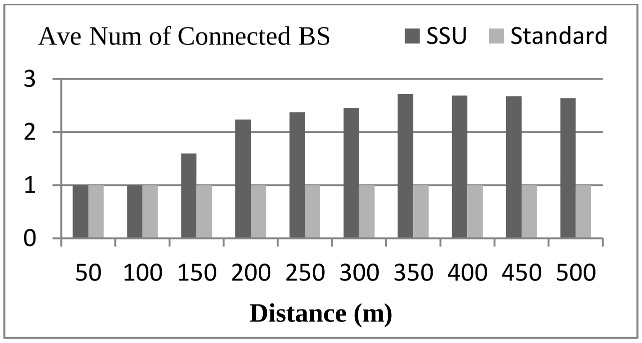


Figure 8. Average number of connected BSs (2000 MHz)

Figures 9 and 10 show the average upload time as a function of distance (for SSU and the conventional non-cooperative algorithms). The upload process starts with the *UploadRequest* message from the UE (step 1 in Figure 2) and it ends when the UE receives the *BitField* message from its serving BS (step 9 in Figure 2).

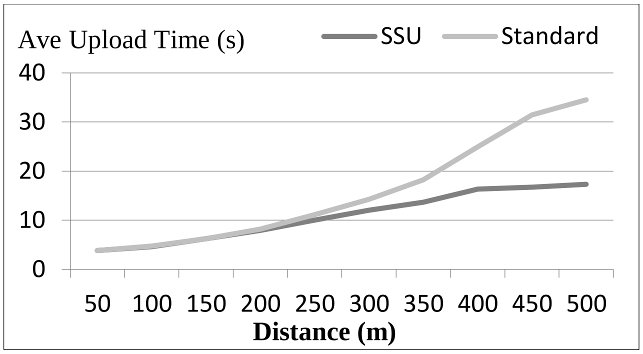


Figure 9. File upload time vs. Distance (900 MHz)

Figures 9 and 10 reveal that as the UEs' distance from the cell center increases, the rate of increase in average upload time for the standard algorithm is higher than that of the SSU. This means that SSU provides a better performance for its users. The effects of SSU on the upload process can be seen more clearly, when the UEs are about 300 to 500 m away from the cell center. Figures 11 and 12 show the comparison between SSU and the standard algorithm with respect to the average data rate they provide for the UEs during the simulations. It can be seen in these figures that the average data rate that SSU algorithm provides for its cell-edge users (located around 500 m from the cell center) is almost double the average data rate that the standard algorithm provides. Again, this shows the performance gains made available by the SSU algorithm for the uplink channels of cell-edge users.

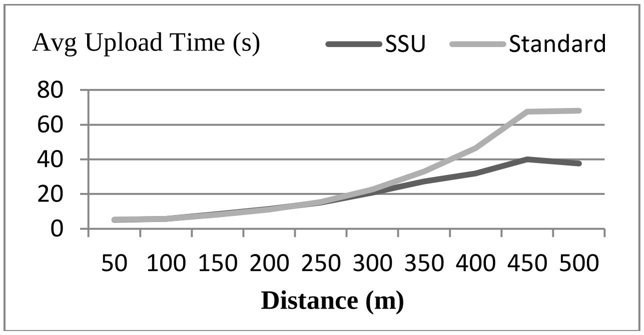


Figure 10. File upload time vs. Distance (2000 MHz)

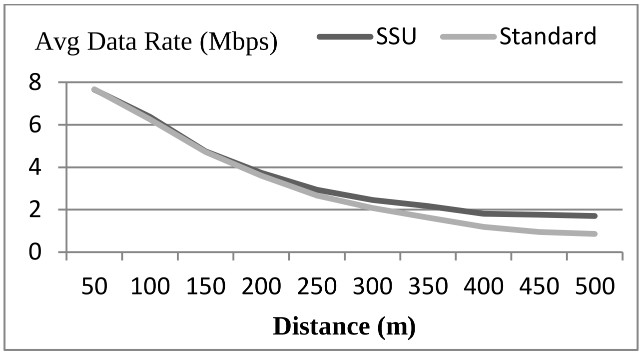


Figure 11. Data rate vs. Distance from BS (900 MHz)

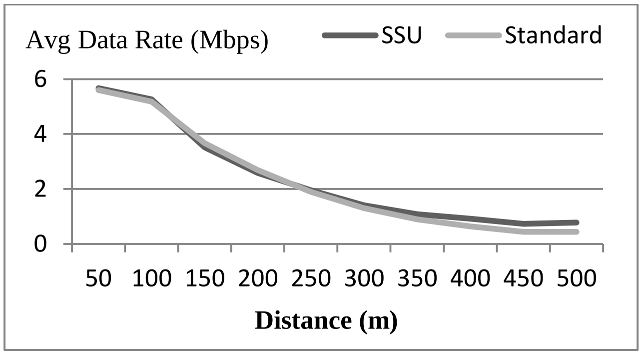


Figure 12. Data rate vs. Distance from BS (2000 MHz)

In the second set of simulations, the designated urban area required 16 BSs to provide coverage for the entire area. The main goal of this set of simulation is to study the effect of increasing the number of cell-edge UEs on the performance of SSU. We considered a limited scale scenario. We began with 16 cell-edge UEs in the first round of simulations and we increased the number of the UEs on each of the following iterations by 16. In the last round of simulations, we had 160 active cell-edge UEs. In a CoMP scenario, as the number of the cell-edge UEs increases, the number of the required coordination sets increases as well. This means a BS may need to deal with more UEs (the UEs belong to this BS cell and the UEs from neighboring cells) compared to conventional non-cooperative algorithm (in which each BS only deals with the UEs of its cell). The important point is that the increase in the number of coordination sets imposes an overhead on the processing resources (BSs) and on the backhaul (X2 links). Therefore, there should be a trade-off between the number of coordination sets, and the overhead on the BSs and the backhaul.

In Figures 13-16, the horizontal axis shows the number of the UEs in the coverage area. Figure 13 shows that the SSU algorithm helped the cell-edge UEs upload their data in a shorter period of time compared to the standard method. In addition, SSU provided better services for its users even when the number of UEs increased. Similarly, in Figure 14, SSU provided a higher data rate for its users and maintained its quality of service while the number of UEs increased. Figures 15 and 16 provide similar conclusions regarding the performance of SSU. These figures show that the SSU algorithm provides lower latency and higher data rate for its users, compared to the standard method and it could offer an equivalent quality of service even when we increased the number of the UEs in the coverage area. Although these simulations show the effectiveness of SSU algorithm in the limited scale scenarios, but as the next step, we need to perform additional simulations with a larger number of UEs to study the performance of SSU and its effect on the mobile network backhaul in more depth.

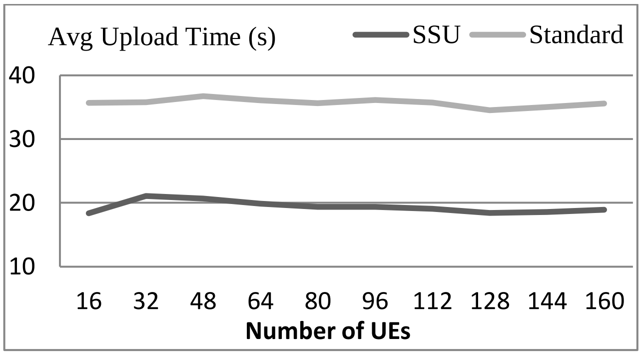


Figure 13. Upload time vs. number of UEs (900 MHz)

CONCLUSION

We have used the Discrete Event System Specification formalism (DEVS) for modeling and simulating LTE-A mobile networks using two protocols: Shared Segmented Upload algorithm (SSU) and a conventional non-cooperative method. The CD++ software was used as the platform to implement and test both algorithms in an urban area setting. The SSU is an uplink schema for LTE-A networks. The simulation results in this study confirmed that in an urban area, the cell-edge users would have better performance by using SSU for uploading their data, compared to using the conventional non-cooperative method. This schema enhances the cell-edge user performance in an upload process by letting the UEs transfer large files in small segments to a set of BSs in the coordination set. The SSU algorithm provides higher average data rate for its users in urban areas, which leads to reduced average upload time. Further investigation is necessary on the influence of piece sizes on the overhead of the algorithm on the backhaul. In addition, to provide a more in-depth evaluation of the performance of SSU, other LTE-A CoMP algorithms will need to be modeled and simulated.

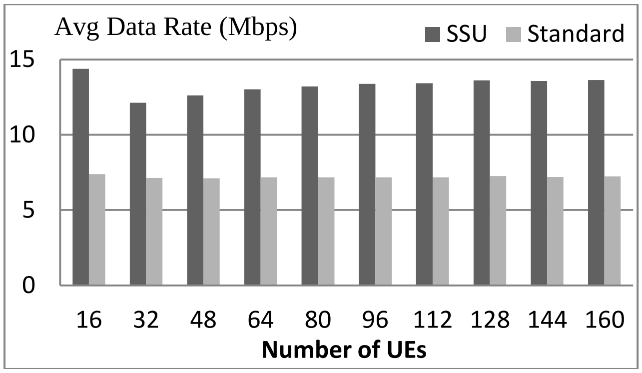


Figure 14. Data rate vs. number of UEs (900 MHz)

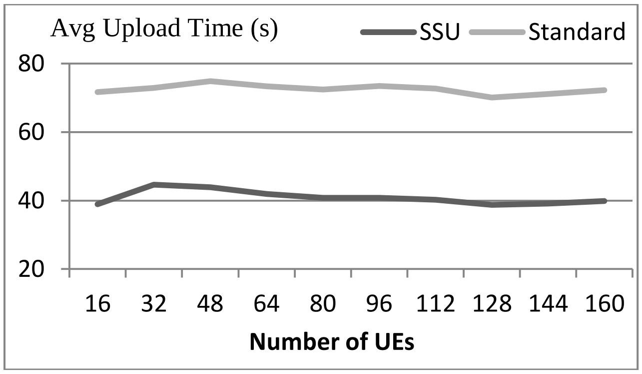


Figure 15. Upload time vs. number of UEs (2000 MHz)

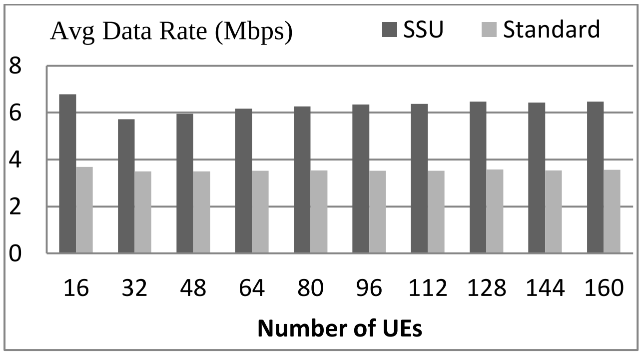


Figure 16. Data rate vs. number of UEs (2000 MHz)

REFERENCES

1. MobiThinking Latest Mobile Stats, http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a. Accessed on October 2014.
2. M. Tavanpour, G. Wainer, G. Boudreau and R. Casselman, “DEVS-based Modeling of Coordinated Multipoint Techniques for LTE-Advanced.” Proceedings of 2013 SCS/ACM 16th Communications and Networking Symposium (CNS'13). San Diego, US, April 2013.
3. I. F. Akyildiz, D. M. Gutierrz-Estevez, E. C. Reyes, “The Evolution to 4G cellular system: LTE-Advanced.” Physical Communication, Volume 3, No 4, PP 217-244, 2010.
4. M. Tavanpour, J. Mikhail, M. Moallemi, G. Wainer, G. Boudreau and R. Casselman, “Shared Segmented Upload in Mobile Networks using Coordinated Multipoint.” Proceedings of 2014 Summer Simulation Multi-Conference. Monterey, US, July 2014.
5. X. Xian, W. Shi and He Huang, “Comparison of OMNET++ and Other Simulator for WSN Simulation”, Proceeding of 3rd IEEE Conference on Industrial Electronics and Applications (ICIEA), PP 1439-1443, June 2008, Singapore.
6. M. M. Koksal, “A Survey of Network Simulators Supporting Wireless Networks”, http://www.ceng.metu.edu.tr/~e1595354/A%20Survey%20of%20Network%20Simulators%20Supporting%20Wireless%20Networks.pdf , Accessed on October 2014.
7. N. Kamoltham, K. N. Nakorn, K Rojviboonchai, “From NS-2 to NS-3 - Implementation and evaluation”, Computing, Communications and Applications Conference (ComComAp), PP 35-40, January 2012, Hong Kong.
8. P. Rajankumar, P Nimisha, P Kamboj, “A comparative study and simulation of AODV MANET routing protocol in NS2 & NS3”, International Conference on Computing for Sustainable Global Development (INDIACom), PP 889 – 894, March 2014, New Delhi.
9. M. Tavanpour, J. Mikhail, M. Moallemi, G. Wainer, G. Boudreau and R. Casselman, “Shared Segmented Upload in Mobile Networks using Coordinated Multipoint,” Provisional Patent P43130, 2014.
10. Q. Qiu, J. Chen, L. Ping, Q. Zhang, X. Pan, “LTE/SAE Model and its Implementation in NS 2.” 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '09). PP 299 – 303, Fujian, Dec. 2009.
11. A. Koc, S. Jha, R. Vannithamby, M. Torlak, “Device Power Saving and Latency Optimization in LTE-A Networks Through DRX Configuration,” IEEE Transactions on Wireless Communications, Volume: PP, Issue: 99, PP 1 – 12, March 2014.
12. R. Sivaraj, A. Pande, P. Mohapatra, “Spectrum-aware Radio Resource Management for Scalable Video Multicast in LTE-Advanced Systems,”IFIP Networking Conference PP 1-9, Brooklyn, NY. 2013.
13. G. A Wainer, “Discrete-Event Modeling and Simulation: A Practitioner’s Approach”, CRC Press, 2009, ISBN 978-1-4200-5336-4.
14. “3rd Generation Partnership Project,” Technical Report 36.942, V10.2.0”, (2010-12).