Using Elected Coordination Stations for CSI Feedback on CoMP Downlink Transmissions

Baha Uddin Kazi, Mohammad Etemad, Gabriel Wainer
Dept. Systems and Computer Engineering
Carleton University. Ottawa, ON, Canada
{bahauddinkazi, mohammadetemad, gwainer}@sce.carleton.ca

Gary Boudreau
Ericsson Canada
3500 Carling Avenue
Ottawa, ON, Canada
{gary.boudreau}@ericsson.com

***Abstract*—The growing use of LTE Advanced (LTE-A) in mobile networks, higher demand for data traffic, and the emergence of new applications has made it challenging to maintain high data rates for users, in particular those located on the cell's edges. The Coordinated Multipoint (CoMP) technology adopted in LTE-A allows improving the cell's edge performance. In order to improve throughput performance gain in the downlink, the CoMP scheduler needs to know/deal the channel information for all the collaborating Base Stations. To do so, we propose a method for handling Channel State Information (CSI) feedback, named DCEC: Direct CSI feedback to Elected Coordination station. The DCEC architecture aims to reduce the overhead and latency of the network, and subsequently increase its throughput. To model the proposed architecture in the cellular network, we have used the Discrete Event System Specification (DEVS) formalism. The simulation results demonstrate that the DCEC architecture significantly decreases the number of CSI feedback packets being transmitted within the network and reduces the feedback latency resulting in higher data rates for users.**

Keywords*— *CoMP, CSI feedback, LTE-Advanced, DEVS, CD++.

I. Introduction

The demand of higher data transmission rates, reliable connection and uniform quality of service across the cell area in mobile communication systems are ever increasing. For instance, the growth in mobile data traffic between the 3rd quarter of 2014 and the 3rd quarter of 2015 was about 65 percent [1]. In order to meet this challenge, a reuse of radio resources in every cell is necessary. Consequently, systems using such techniques experience intercell interference (ICI), which limits the user throughput, in particular on the cell's edge.

The Coordinated MultiPoint (CoMP) transmission/ reception method, also known as Multipoint Cooperative Communication (MCC), can improve the network performance by boosting up the cell's edge throughput [2]. In CoMP enabled systems, the Base Stations (BSs) are grouped into cooperating clusters, each of which contains a subset of the network BSs. The BSs of each of these clusters exchange information and jointly process signals by forming virtual antenna arrays distributed in space. A cluster could form based on static or dynamic clustering algorithms [3]. Furthermore, multiple User Equipment (UEs) can simultaneously receive their signals from one or multiple transmission points in a coordinated or joint-processing manner. This technique is an effective way of managing the ICI.

In order to improve the performance of the network while mitigating the ICI, the UEs need to estimate the Channel State Information (CSI) and feed it back to a scheduler. In order to do so, the UEs calculate the CSI and report it to a BS to perform adaptive transmission and appropriate radio resource management (RRM) [2, 4]. This results in an increase in signaling overhead and latency of the network [5]. Another type of overhead related to CoMP is the *infrastructural overhead* [5, 6, 7]. (i.e., the network may require additional control units and low-latency links among the collaborating BSs, which might increase the network cost). These overheads depend on the CoMP control architectures.

There are two types of control architectures available for transmission and reception in CoMP: *centralized* and *distributed* [6, 7, 8]. In the centralized architecture, a central unit is responsible for handling radio resource scheduling by processing the feedback information from the cell sites. This framework suffers from signaling overhead and infrastructure overhead, as well as an increase in the network latency. On the other hand, in the distributed architecture, the coordinated cells exchange data and Channel State Information (CSI) over a star-like S1 network and a fully meshed signaling network using X2 interfaces. This architecture increases the feedback transmission and is more sensitive to error patterns. This could potentially cause further performance degradation [7].

With these issues in mind, we introduce a new CoMP control architecture named *Direct CSI-feedback to Elected Coordination-station* (DCEC), aiming to reduce the overhead and latency and subsequently increasing the throughput of the network [9]. As shown in [10], the throughput of the cell can increase by as much as 20% if the latency is reduced by 5ms. The DCEC architecture addresses the challenges of both the architectures mentioned above. In the DCEC architecture, one of the BSs in the CoMP cluster is dynamically elected to act as a Central Coordination Station (CCS) for the UEs. After a CCS is elected, all the UEs in the CoMP Cluster with the same CCS will send the CSI feedback to this CCS only. The CCS will then calculate the global CSI information, determining the cooperation set, and will be in charge of scheduling. It should be noted that a cooperation set is a set of BSs within the CoMP cluster that can jointly serve the UE [11].

The main goals of this DCEC architecture are:

(I) to reduce the latency of the network;

(II) to reduce the feedback overhead of the network;

(III) to avoid the additional infrastructure cost and;

(IV) to increase the cell's throughput.

There will also be no increase in the error pattern in this architecture since all the participating UEs send the CSI to only the CCS after the CoMP is established. Furthermore, no additional hardware is necessary for this solution, so the costs for switching to such architecture will be minimal.

In order to better understand the results of implementing the DCEC architecture and to be able to compare the performance of the DCEC, Centralized, and Distributed architectures, we set up simulation scenarios using the DEVS formalism. Based on the simulation results obtained, it can be concluded that the proposed DCEC approach reduces the CSI feedback packets transmitted within the CoMP network compared to the other two approaches. Although DCEC requires few more control packets to elect the CCS at the beginning, it outperforms the other two architectures as time advances. Furthermore, in DCEC the CSI feedback does not need to travel the X2 or S1 links, which results in lower feedback latency.

In the following sections, we introduce the architecture for this new solution and discuss a simulation model built using the DEVS formalism [12] and the CD++ toolkit [13]. We will also show different simulation results that summarize the output of the new architecture.

II. Related Work

In 3GPP long-term evolution (LTE) release 11, CoMP started to be adapted as a key technology [2, 4, 14]. Accurate and updated channel information is a key factor for achieving better throughput performance gain in CoMP. The CSI feedback process is a method in which a UE calculates the channel information and reports that to the BS so that BS can perform adaptive transmission and appropriate Radio Resource Management (RRM).

As shown in Figure 1, there are two kinds of control architectures available for use in the CoMP transmission and reception with respect to how the channel information becomes available at different transmission points: centralized and distributed [6, 7, 8]. In the centralized architecture, a central unit is responsible for handling radio resource scheduling by centrally processing the feedback information from the cell sites. The UEs estimate the CSI related to all the cooperating BSs and feed it back to their serving BS. Their serving BSs then forward the local CSI to the central unit (CU). Finally, this CU calculates the global CSI and makes a decision based on the calculated information. This information is then sent back to the BSs. This framework suffers from signaling overhead and infrastructural overhead (as discussed earlier, this means that the network may require additional control units and links among the collaborating BSs), as well as an increase in the network latency.

On the other hand, in the distributed architecture, the coordinated cells exchange data and Channel State Information (CSI) over a star-like S1 network and a fully meshed signaling network using X2 interfaces. Prior to the download, the UEs estimate the CSI related to all the cooperating BSs and feed it back to all cooperating BSs. The BSs then independently perform scheduling tasks based on their acquired global CSI. This architecture increases the feedback transmission, and is more sensitive to error patterns. This could potentially cause further performance degradation [7].

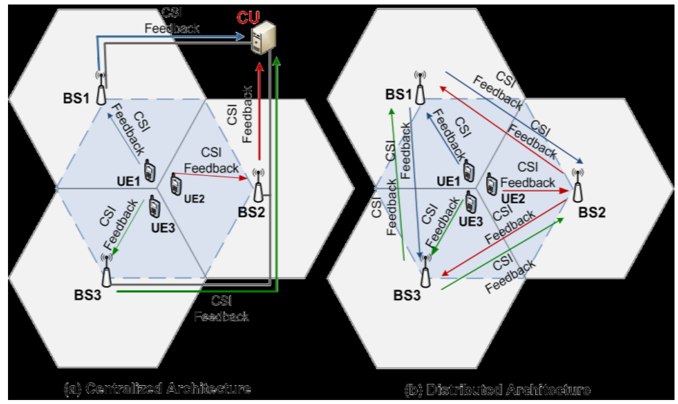

Fig. 1. Standard CoMP architectures

In [15], the authors propose a centralized MAC approach for CoMP joint transmission. In this approach, the authors group the BSs into clusters. Within a cluster, one of the cells is preconfigured as the head sector and the others act as proxies. Gao et al. propose a modified version of an existing algorithm for dynamic cell selection in CoMP [16]. They extended the dynamic cell selection method to a Multi-Cell scenario, which originally is limited to one chosen transmission cell. In [17], the authors propose a distributed architecture for CoMP Joint Transmission (JT), which works over an IP backhaul network between BSs. They introduce two levels of time scales: (1) radio resources for CoMP JT are allocated every several 100s of milliseconds; (2) modulation and coding schemes for link adaptation are calculated every millisecond.

In our proposed CoMP control architecture, one of the BS will be elected dynamically as a Central Coordination Station (CCS). The elected CCS will be responsible for receiving the CSI feedback messages from all the participating UEs in the CoMP session. These UEs will be sending the CSI feedback message directly to their CCS. This will result in a reduction of overhead and latency of the CSI feedback. In this architecture, the CCS is responsible for scheduling resources [9].

As we mentioned earlier, we built various models and ran simulations of our DCEC architecture. To do so, we used Discrete Event Systems Specifications (DEVS), a formal modeling and simulation methodology [18]. DEVS methodology is based on dynamic systems theory. DEVS models are organized hierarchically, using modular descriptions, supporting discrete event approximation of continuous systems and allowing model reuse.

A real system modeled with DEVS is described as composed of atomic (behavioral) models and hierarchically combined coupled (structural) models. The atomic component of the model is the basic building block of the system, which represents the behavior of a part of the system.

CD++ is an open source toolkit that implements the DEVS theory [19]. The toolkit has been developed as a hierarchy of models, each related with a simulation entity. The hierarchical and modular nature of DEVS allows the description of the multiple levels, and allows models to be extended easily as needed for our model. DEVS is a formal specification, which is useful to improve the security and development cost of a simulation. The same model can be extended with different DEVS-based simulators, allowing for portability and interoperability at a high level of abstraction. It also allows accurate timing representation as it uses a continuous time base. Finally, the use of formal modeling techniques enables automated model verification [18].

In [19], we discussed various applications of DEVS for modeling and simulation of wireless networks. Particularly, modeling and simulation of a cellular network including a wide geographical area using various Cells and varied UEs. In [18], we also showed that DEVS could be a useful tool for performing modeling and simulation of large-scale web search engines. In the following sections, we will discuss our newly proposed algorithms and the modeling and simulation analysis of its implementation using DEVS and CD++.

III. Coordination Station Election and CSI Feedback

The control architecture of CoMP can be defined as the way participating cell sites coordinate to handle interference and scheduling. As discussed in Section 2, there are two kinds of architectures for CoMP (with respect to the way in which this information is made available to the different transmission points). Based on these control architectures, we need different CSI feedback processes.

For the downlink, the CoMP signaling overheads are a result of the need of the Channel State Information (CSI) at the transmitter end [5]. This global CSI feedback process could be different based on the architecture of the CoMP. Two major challenges of the above architectures are latency and overhead. Latency is inversely related to the throughput of the network, in particular for the coordinated schemes. However, if the latency of the network in CoMP is greater than the CSI feedback periodicity, then the scheduler will receive backdated CSI. Hence, latency is a concern for CoMP. One of our goals is to reduce this latency in order to improve the cell's throughput.

The main mechanisms reducing latency and overhead include network architecture optimization, shorter transmission time interval (TTI), faster feedback processing, and QoS load differentiations [20]. We propose the use of an elected coordination station for CSI feedback, which addresses both of the abovementioned challenges (*latency and overhead*). This control architecture, named *Direct CSI-feedback to Elected Coordination-station* (DCEC), dynamically uses one of the BSs in the CoMP cluster as a *Central Coordination Station* (CCS) for the UE.

The CCS is chosen based on an election algorithm, which will be described in detail later in this section. All the UEs in the CoMP cluster having same cooperation set send the CSI feedback to the CCS only. Therefore, this signal does not need to travel any additional X2 or S1 channels, avoiding extra latency of the CSI feedback transmission and reducing the feedback overhead in the network. Figure 2 shows a simplified view of the proposed CSI feedback architecture after the CCS has been elected within the CoMP set.

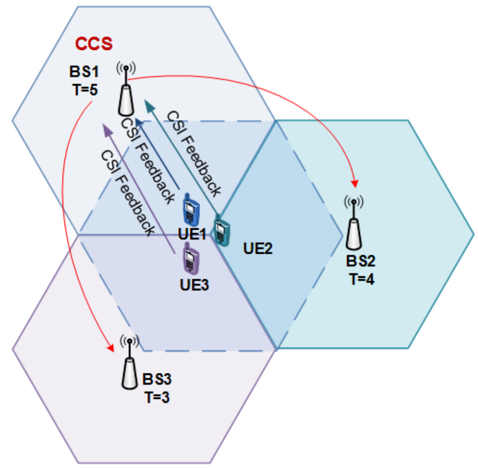

Fig. 2. A simplified view of CSI feedback of the DCEC architecture

To elect a Centralized Coordination Station dynamically we use the following algorithm:

1. *If the CoMP set contains more than one BS, the serving BS in the CoMP Cluster declares itself as a CCS.*
2. *This CCS sends a CCS-Declaration message to the other BSs in the CoMP cooperation set, containing the ID of the sender, the ID of the CCS, and its cell throughput.*
3. *After the other BSs receive the declaration message, they compare their throughput to the received CCS throughput.*
   a. *If the received CCS throughput is higher than the recipient's throughput (or the current), the CCS ID will change to the received ID. The recipient will forward the new CCS information (all the BSs in the cooperation set except for the sender).*
   b. *If the received CCS throughput is equal to its own throughput (or the current) and the CCS ID is smaller than its own ID (or the current), the current CCS ID will become the received CCS ID. The recipient will forward the new information to the BSs in the cooperation set.*
   c. *If the received CCS throughput and ID are equal to the current one, the CCS has been elected.*
   c. *Otherwise, the recipient BS declares itself as the new CCS and sends a CCS-Declaration message to the other BSs in the CoMP cooperation set.*
4. *If the cell throughput or the cooperation set change, then go back to step 1.*

Figure 3 shows a simplified signaling procedure of the proposed scheme. In the beginning, UE1 reports the CSI feedback to its serving BS (BS1). Then, BS1 calculates the cooperation set for UE1. To do so, BS1 checks the channel quality and compares the predefined CoMP threshold (3dB as discussed in [21]) with the data received. If the cooperation set contains more than one BS, BS1 initiates the algorithm to select the CCS by sending a CoMP request message to other BSs in the cooperation set (BS2 and BS3) with his own cell throughput. For simplicity, we assume that all three BSs are in the cooperation set. After receiving the CoMP request message, BS2 and BS3 check their own resources and compare the received throughput with their own throughput. Based on the availability of resources they will send back a request grant/reject message, including the highest throughput. After receiving the responses from other BSs, BS1 will make a decision about who is the CCS, and it will advertise it to BS2, BS3, and to UE1 using a CoMP notification message. Finally, the UE will reply using the ACK message and it will switch to the CoMP mode. After the CoMP is established and the CCS is elected, the UEs will send the CSI feedback only to the CCS, as shown in Figure 2.

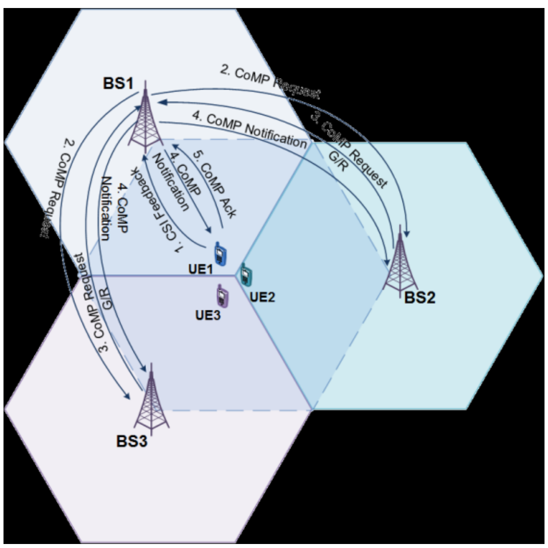

Fig. 3. Message transfer to establish CoMP with CCS election in DCEC

IV. Modeling the CoMP Network Using DEVS

We have defined a DEVS model in order to analyze the control architecture of the CoMP network employing the DCEC algorithm, the model consists of various atomic and coupled models as shown in Figure 4.

The top-level coupled model is the CoMP cluster that includes three cells. We have considered three cells, since most of the research on CoMP performance analysis shows that CoMP provides the best performance with the cooperation of three cells [22]. We will not discuss the details of the clustering procedure and formation [3] as this is a well-known procedure and it is out of the scope of this paper.

We have defined each cell as a coupled model with two components: *BS* (base station) and *UE* (user equipment). The cell may contain only one BS but multiple UEs. For simplicity, we only show one UE model in each cell in Figure 4. The dotted links connecting the BSs represent the X2 link. Each BS and UE coupled model is composed of two atomic models named *Buf* and *Proc*. The UE Proc generates the CSI feedback based on the signal strength received from the cooperating BSs every 10 ms, and sends the CSI feedback message to its serving BS Buf through the output port *Out*. The BS Buf pushes the message into the queue and forwards the message to the BS Proc when the Proc send a request message. Once the BS Proc receives a message, it executes the algorithm discussed earlier to establish a CoMP cooperation set and select a CCS.

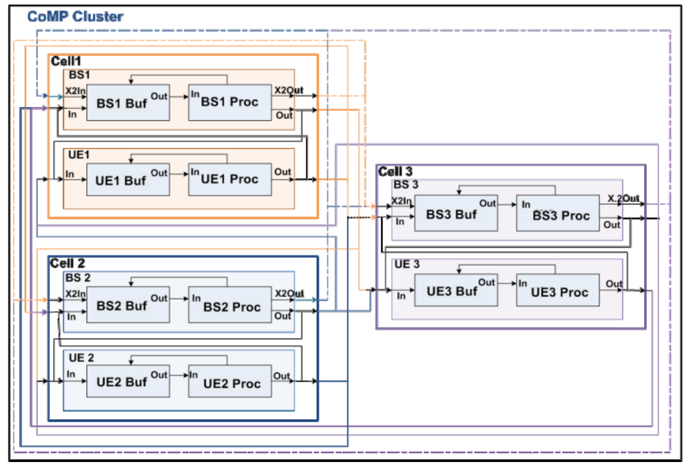

Fig. 4. Simplified coupled model for CoMP control architecture analysis

Figure 5 shows a sample code snippet, which is a part of the BS Proc Atomic Model. We use five types of messages:

- CSI_FEEDBACK: contains the channel state information sent from the UE to the BS
- COMP_REQ: a request sent from the serving BS to other BSs in the CoMP cooperation set to join CoMP.
- COMP_REQ_G_R: a grant/reject message sent from the recipient BSs to the serving BS based on the availability of resources.
- CoMP_COMMAND: a command sent from the serving BS to the UE informing about the elected CCS.
- COMP_NOTIFICATION: a notification sent from the serving BS to other BSs to notify about the establishment of CoMP and the elected CCS.
- COMP_ACK: an acknowledgement of the receipt of the command sent from the UE to the serving BS.

```
BS::BS( const std::string &name ) : Atomic( name ),
  In(addInputPort("In")), Out(addOutputPort("Out") ),
    X2out(addOutputPort("X2out")),
         Req(addOutputPort("Req"))

Model &BS::externalFunction( ExternalMessage &msg ){
   if (port()==In) {
      if (valueO()->getMsgT()==CSI_FEEDBACK)
         ...
      else if (valueO()->getMsgT()==COMP_REQ)   {
         CoMPReqFlag = true;
         CoMPReq* message = (CoMPReq*)valueO();
         RecThroughput = message->getThroughput();
         DestID = message->getSrcID();
```

```
          SourceueID = message->getSourceCSIueID();
        }
        else if (valueO()->getMsgT()==COMP_ACK)
          ...
        state = SendPack;
    } }
Model &BS::outputFunction( InternalMessage &msg ){
  if (state == RecPack)
        sendOutput(time(), Req, 1, NULL);
  else if (state == SendPack)
        if (CSIFeedbackFlag == true)  {
            if(Threshold1 <= COMP_THRESHOLD &&
               Threshold2 <= COMP_THRESHOLD &&
        CoMPRequestStartOrStopFlag[SourceueID] == 0) {
          requestMsg1 = CoMPReq(id,Nid1,1,MyThroughput,
                     id, SourceueID);
          requestMsg2 = CoMPReq(id,Nid2,1,MyThroughput,
                     id, SourceueID);
          sendOutput(time(), X2out, NULL, requestMsg1);
          sendOutput(time(), X2out, NULL, requestMsg2);
        }
        ...
        CSIFeedbackFlag = false;
    }

    if (CoMPReqFlag == true)      {
        if (MyThroughput > RecThroughput)
           requestGRMsg =  CoMPReqGR(id, DestID, 1,
                       MyThroughput, id, SourceueID);
        else
           requestGRMsg =  CoMPReqGR(id, DestID, 1, Rec
                      Throughput, DestID, SourceueID);
        sendOutput(time(), X2out, NULL, requestGRMsg);
        CoMPReqFlag = false;
    }
    if (CoMPRGFlag == true) {
     if ((CompRGValue12[i] == 1) && id == 1
              && (CompRGValue13[i] == 1))        {
       if (CoMPReqGRThroughput1>CoMPReqGRThroughput2){
        notifyMsg1 = CoMPNotify(id,CoMPReqGRSourceID1,
          1, CoMPReqGRCCSID1, CoMPReqGRThroughput1,
              SourceueID );
        CoMPcommand = CoMPCmd(id, SourceueID, 1,
              CoMPReqGRCCSID1);
       }
       else
         ...
    }
          sendOutput(time(), X2out, NULL, notifyMsg1);
          sendOutput(time(), Out, NULL, CoMPcommand);
        }
        ...
        CoMPRGFlag = false;
    }
    sendOutput(time(), Req, 2, NULL);
}

Model &BS::internalFunction(InternalMessage &){
  switch (state){
    case Idle:
        state = RecPack;
        holdIn(Atomic::active, ProcessTime); break;
    case RecPack:
        state = Idle; break;
    case SendPack:
        state = RecPack;
        holdIn(Atomic::active, ProcessTime); break;
    }
}
```

Fig. 5. Sample code snippet for the the BS Proc Atomic Model in CD++

The model advances through three states for the BS: *Idle*, *SendPack* (in which the BS starts sending a message to either another BS or a UE), and *RecPack* (in which the BS waits to receive a message from either another BS or a UE). Starting from Idle, if the BS receives an external message, it will be processed based on its type. The BS then changes its state to SendPack to get ready to send out the processed information. Once the processed information is sent, the BS will change its state to RecPack, sends a request to its buffer to forward it a new message, and switches to Idle to wait for a message. The UE follows the same procedure.

V. SIMULATION RESULTS

In this section, we present different simulation scenarios and the results we obtained for the control architecture based on the algorithm introduced in the previous sections. Figure 6 shows the architecture of a sample scenario with three cell-edge UEs in three different cells and one BS in each cell.

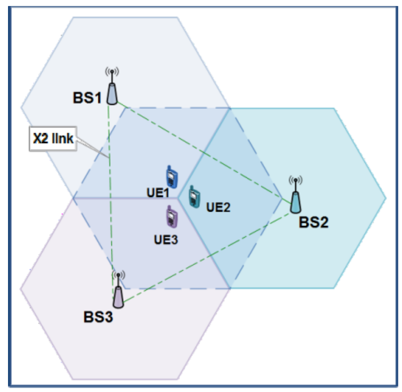

Fig. 5. The system architecture for a basic simulation scenario

To assess the potential of the DCEC control architecture, we ran a series of simulations on this model, based on the initial conditions summarized in Table 1. We have chosen our cell radius and antenna gain parameters to align with the specifications outlined in LTE release 12. As [23] suggests, the carrier frequency has been set to 900 MHz, the cell radius has been set to 500 m, and the antenna gain has been set to 14 dBi for the BS and 0 dBi for the UE. Based on [8, 24] the CSI feedback frequency has been set to 10 ms. In our simulations, cells are considered as macro cells in an urban area. A typical transmission power for such macro cell is normally between 43 dBm to 48 dBm. Hence, we set the transmit power for a BS to 46dBm [25].

TABLE 1. SIMULATION ASSUMPTIONS

Parameters	Values
Number of macro cells	3
BS Transmit Power	46 dBm
Carrier Frequency	900 MHz
Cell Radius	500 m
Antenna gain	14 dBi (BS); 0 dBi (UE)
Cell Throughput	Randomly generated
CSI Feedback Frequency	10 ms
CoMP Threshold	3 dB

The received signal power at each UE is calculated based on the following formula [26]:

$$P_r = P_t - \text{Max}\ (L_{path} - G_t - G_r\ ,\ MCL)$$

in which P_r is the received signal power, P_t is the transmitted signal power of the BS, G_t is the transmitting antenna gain, G_r is the receiver antenna gain and L_{path} is the path loss. The Minimum Coupling Loss (MCL) is set to 70 dB, the BS antenna gain is set to 14 dBi, and the UE antenna gain is considered to be 0 dBi [26]. After calculating the received power, a CSI feedback message is generated and sent to the BSs. We configured the BSs to generate the cell throughput at random. Based on the literature in this area, we considered the CoMP threshold as 3dB [21] and we find the CoMP cooperation set dynamically. A cooperating set consist of a number of BSs within the CoMP cluster that can serve the UE in a joined manner [11].

In order to be able to analyze the advantages and disadvantages of the DCEC architecture over the distributed and centralized CoMP, we simulated the algorithms using the scenarios above. To simulate the distributed CoMP architecture we assume that the BSs are synchronized.

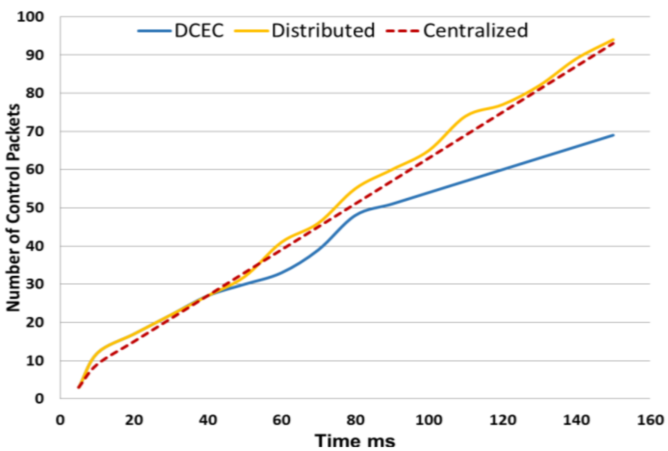

Fig. 6. Comparison of DCEC, Centralized, and Distributed architectures in CoMP based on accumulative number of control packets over time

Figure 7 shows the accumulative number of control packets transmitted in the network related to CoMP as a function of time for the DCEC and the other two conventional architectures. Here, we consider that there are three UEs (one in each cell) located in the same CoMP region that send a CSI feedback message every 10ms. In Figure 7 we can see that after some time, the DCEC architecture has a slower linear growth compared to the other two architectures in terms of the number of packets. This shows that although initially DCEC may require more control packets, it will outperform the other conventional architectures. This is because in the conventional approaches, the CSI feedback messages are still required to travel the X2 or S1 links even after the establishment of CoMP.

Figure 8 shows a comparison among the CoMP architectures with respect to the number of control packets in the network as a function of time for a different scenario. In this case, three BSs and ten UEs are present. The CSI feedback frequency is 10ms, and the BS process time is assumed to be very low (1ms). In this scenario at 80ms, 2 UEs leave the CoMP set, and at 120 ms and 130 ms 3 more UEs and 1 more UE join the CoMP set respectively. As observed, when the two UEs leave the CoMP session, the accumulative number of control packets slows down for all the architectures (with a delay due to the fact that the UEs leaving CoMP takes time to penetrate through the entire network). On the other hand, when four more UEs join the CoMP session at 120ms to 130ms, DCEC takes a rapid increase and then slows down, but the other two architectures increase continuously. DCEC is more sensitive to change in the short term (i.e. throughput changes or UEs joining/leaving the CoMP region), but it recovers rapidly (and in the long run it outperforms the other two architectures). If the UEs stay in the CoMP session for a longer period, DCEC would need less control packets with respect to other two approaches.

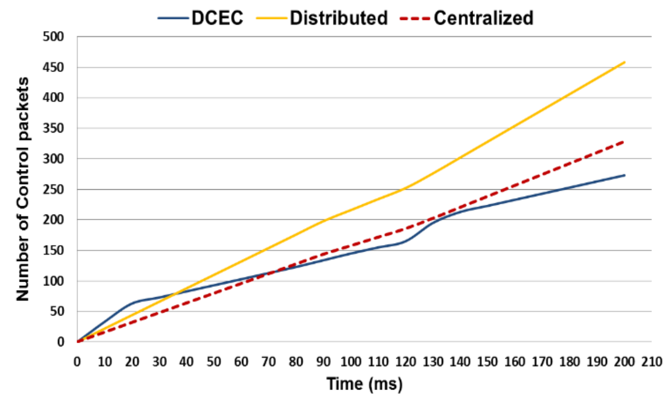

Fig. 7. DCEC, Centralized, and Distributed architectures based on the number of control packets in the network (3 BS and 10 UE)

Figure 9 shows the accumulative number of control packets transmitted up to a certain time for each of the architectures in the dB scale. The DCEC architecture is represented by three instances to see how CCS changes affect the number of control packets transmitted. In the first case, we assume that the throughput is constant, that is, the CCS does not change throughout the simulation. In the second and third cases, the CCS is set to change every 1s and every 100ms respectively.

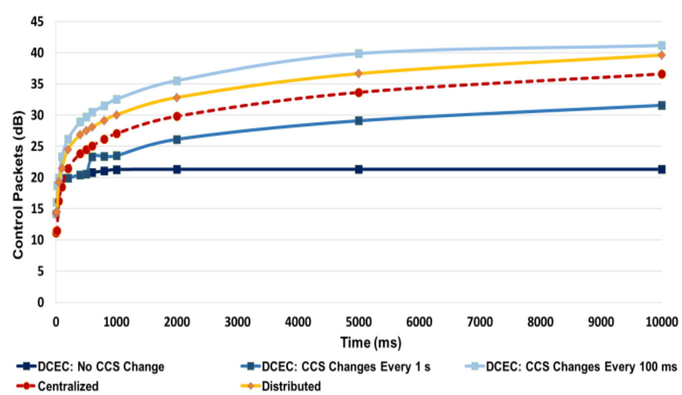

Fig. 8. Accumulative control packets for DCEC without CCS change, DCEC with CCS change every 1s/100ms, Centralized, and Distributed architectures

As we can see, DCEC with no CCS changes or with changes every 1s outperform the Centralized and Distributed architectures. If the CCS change occurs very rapidly, for example every 100ms, DCEC will be less efficient than the traditional approaches. As mentioned earlier, DCEC is more sensitive to change compared to the other two architectures. Given enough time to recover from the change, DCEC can outperform the other two conventional methods, although, if the rate of changes is very high, DCEC will perform worse than the Centralized and Distributed architectures. In practice, the CCS change does not occur that frequently for most of the UEs since the maximum movement speed of a UE suggested by the 3GPP release 11 for CoMP deployment is 3km/h [11].

According to the results of the simulation, as seen in Figures 7, 8 and 9 the DCEC architecture has the potential to reduce the number of feedback overhead within the CoMP network compared to the other two conventional approaches, without the need to change the frequency of the CSI feedback.

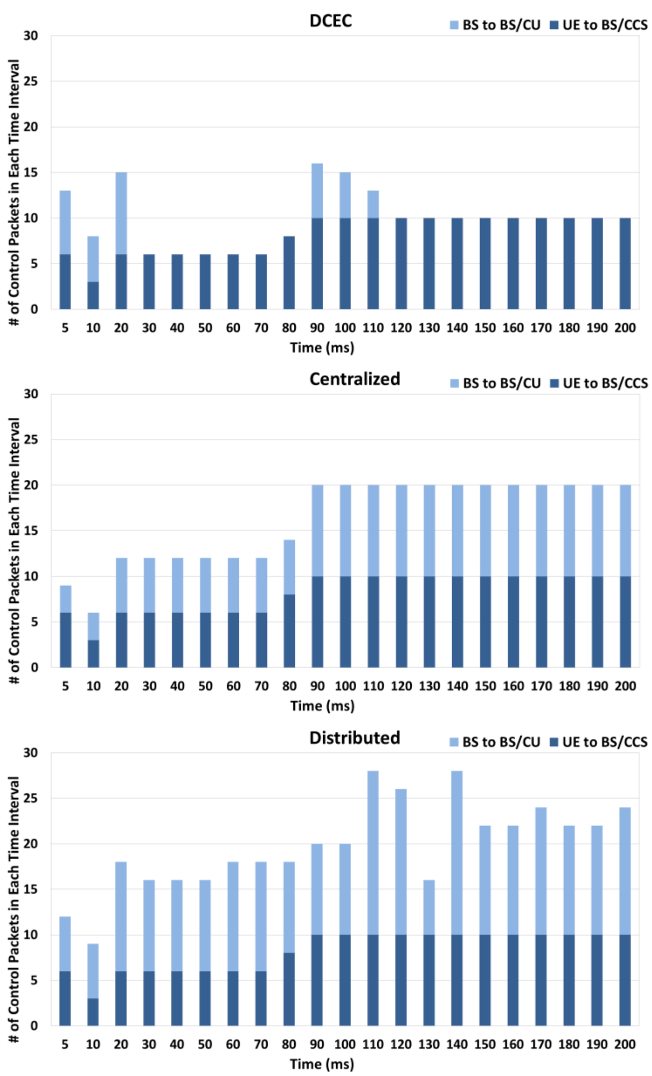

Fig. 9. Number of control packets at different time intervals for DCEC, Centralized and Distributed architectures: packets over the backhaul/ air links.

Figure 10 shows the number of control packets related to the CoMP download transmission traveled from the UEs to the CCS/BSs and the Backhaul in specific time intervals, for all the three control architectures. The darker part of each bar shows the number of CSI Feedback packets which travel from UE to BS and UE to RRH. The lighter part shows the CSI feedback forwards from BS to BS, BS to CU and the overhead related to the election algorithm. Each bar represents 10ms of execution, except for the first two groups, which are 5ms each. In this scenario, the UEs send the CSI feedback their serving BS or CCS every 10ms. The results of the simulations show that in the beginning of the establishment of CoMP, the DCEC control architecture will require additional control packets to be sent over the backhaul; but after the CCS has been elected there will be no control packets transmitted except for the CSI feedback from the UE to the CCS. As it is clearly seen in Figure 10, no additional control packets will be transmitted within the 30ms to 80ms timeframe. In the 90ms to 120ms timeframe, several new UEs join the CoMP, which results in additional control packets being transmitted through the backhaul to elect the CCS. After 120ms, there will be no additional control packets required in the DCEC architecture, since the CCS selection is completed. On the other hand, the other two conventional architectures need the CSI feedback to be forwarded over the backhaul every time. Here we use the feedback delay as the total time between measuring the CSI at the UE and using it during the scheduling. In most practical systems, the CSI feedback consists of processing time, transmission time and waiting time for the scheduler [27]. Therefore, according to the result of the simulation, as seen in Figure 10, we can see that the DCEC architecture reduces the CSI feedback latency related to the backhaul compared to the other two control architectures. Moreover, based on the above results we can say that the DCEC control architecture can reduce the CSI feedback overhead in the network as well as the CSI feedback latency. The reduction of overhead and CSI feedback latency eventually improve the network throughput [28, 29].

VI. Conclusion

The main goal of the CoMP approach is to improve the data rate especially for the cell edge users as well as to increase the throughput of the network. However, the two standard architectures (Centralized and Distributed) of CoMP face some challenges such as latency, signaling overhead and infrastructural overhead. In this work, we introduced a new CSI feedback scheme based on elected Central Coordination Station (CCS) for CoMP to reduce the latency and the overhead so that the overall throughput of the network could be improved. The Central Coordination Station (CCS) election algorithm has been implemented with CoMP in the scenarios mentioned in the previous sections of this paper. We have also shown how this DCEC control architecture for CoMP reduces the CSI overhead and the CSI feedback latency compared to two other standard CoMP approaches. A potential possibility to expand this work is to extend the DCEC approach for heterogeneous network defined in 3GPP LTE release 11 for CoMP as well as 5G cellular networks. We could also investigate device to device (D2D) multicast among the UEs within the CoMP region for download streaming, since the elected scheduler knows all the participating BSs and UEs in that region. Therefore, D2D multicast within CoMP with elected CCS could be a potential approach to decrease the cell edge interference farther as well as to increase the data rates.

REFERENCES

[1] Ericsson, "Ericsson Mobility Report," 2015. [Online]. Available: http://www.ericsson.com/res/docs/2015/mobility-report/ericsson-mobility-report-nov-2015.pdf. [Accessed 26 March 2016].

[2] M. Ding and H. Luo, Multi-point Cooperative Communication Systems: Theory and Applications, Shanghai Jiao Tong University Press, Shanghai and Springer-Verlag Berlin Heidelberg, 2013.

[3] P. Marsch and G. P. Fettweis, Coordinated Multi-Point in Mobile Communications: From Theory to Practice, Cambridge University Press, 2011.

[4] S. Sun, Q. Gao, Y. Peng, Y. Wang and L. Song, "Interference Management Through CoMP in 3GPP LTE-Advanced Networks," *IEEE Wireless Communications,* vol. 20, no. 1, pp. 59-66, 2013.

[5] B. Özbek and D. L. Ruyet, "Feedback Strategies for Multicell Systems," in *Feedback Strategies for Wireless Communication*, New York, Springer, 2014, pp. 249-293.

[6] A. Papadogiannis, H. Bang, D. Gesbert and E. Hardouin, "Efficient Selective Feedback Design for Multicell Cooperative Networks," *IEEE Transactions on Vehicular Technology,* vol. 60, no. 1, pp. 196-205, 2011.

[7] A. Papadogiannis, E. Hardouin and D. Gesbert, "Decentralising Multicell Cooperative Processing: A Novel Robust Framework," *EURASIP Journal onWireless Communications and Networking,* vol. 2009, April 2009.

[8] I. F. Akyildiz, D. M. Gutierrez-Estevez and E. C. Reyes, "The evolution to 4G cellular systems: LTE-Advanced," *Physical Communication,* vol. 3, pp. 217-244, 2010.

[9] B. U. Kazi, G. A. Wainer, G. Boudreau and R. Casselman, "Coordinated Multi-Point (CoMP) Method And System Using a Coordination Station". Canada Patent P47112 US2, 2015.

[10] "Backhauling X2," Cambridge Broadband Networks, April 2011. [Online]. Available: http://cbnl.com/resources/backhauling-x2. [Accessed February 2016].

[11] 3GPP, "3GPP TR 36.819 version 11.2.0," 09 2013. [Online]. [Accessed September 2015].

[12] B. Zeigler, H. Praehofer and T. Kim, Theory of modeling and simulation, San Diego, CA: Academic Press, 2000.

[13] G. A. Wainer, Discrete Event Modeling and Simulation A Practitioner's approach, Boca Raton, FL: CRC Press, Taylor & Francis Group, 2009.

[14] J. Lee, Y. Kim, H. Lee, B. L. Ng, D. Mazzarese, J. Liu, W. Xiao and Y. Zhou, "Coordinated Multipoint Transmission and Reception in LTE-Advanced Systems," *IEEE Communications Magazine,* vol. 50, no. 11, pp. 44-50, 2012.

[15] S. Brueck, L. Zhao, J. Giese and A. Amin, "Centralized Scheduling for Joint Transmission Coordinated Multi-Point in LTE-Advanced," in *2010 International ITG Workshop on Smart Antennas (WSA)*, Bremen , 2010.

[16] Y. Gao, Y. Li, H. Yu and S. Gao, "Performance of Dynamic CoMP Cell Selection in 3GPP LTE System Level Simulation," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, 2011.

[17] T. Okamawari, H. Hayashi and T. Fujii, "A Proposal on Network Control Architecture for CoMP JT with IP Network between eNBs," in *Vehicaular Technology Conference (VTC Spring), 2012 IEEE 75th*, Yakohama, 2012.

[18] A. Inostrosa-Psijas, G. Wainer, V. Gil-Costa and M. Marin, "DEVS Modeling of Large Scale Web Search Engines," in *Proceedings of the 2014 Winter Simulation Conference*, Savannah, GA, 2014.

[19] G. A. Wainer, M. Tavanpour and E. Broutin, "Application of the DEVS and Cell-DEVS formalisms for modeling networking applications," in *Proceedings of the Winter Simulation Conference (WSC)*, Washington, DC, 2013 .

[20] T. Blajic, D. Nogulic and M. Druzijanic, "Latency Improvements in 3G Long Term Evolution," 2007. [Online]. Available: http://www.ericsson.com/hr/about/events/archieve/2007/mipro_2007/mipro_1137.pdf. [Accessed February 2016].

[21] G. Cili, H. Yanikomeroglu and F. R. Yu, "Cell Switch Off Technique Combined with Coordinated Multi-Point (CoMP) Transmission for Energy Efficiency in Beyond-LTE Cellular Networks," in *Communications (ICC), 2012 IEEE International Conference on*, Ottawa, ON, 2012.

[22] X. Zhang and X. Zhou, LTE-Advanced Air Interface Technology, CRC Press, 2012.

[23] ETSI, "3GPP TS 36.211 version 13.0.0 Release 13," 01 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/136200_136299/136211/13.00.00_60/ts_136211v130000p.pdf. [Accessed February 2016].

[24] ETSI, "3GPP TR 36.942 version 13.0.0 Release 13," 01 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_tr/136900_136999/136942/13.00.00_60/tr_136942v130000p.pdf. [Accessed February 2016].

[25] B. Mondal, E. Visotsky, T. Thomas, X. Wang and A. Ghosh, "Performance of downlink comp in LTE under practical constraints," in *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Sydney, NSW , 2012.

[26] ETSI, "3GPP TR 25.942 version 13.0.0 Release 13," 01 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_tr/125900_125999/125942/13.00.00_60/tr_125942v130000p.pdf. [Accessed February 2016].

[27] S. Brueck, L. Zhao, J. Giese and A. Amin, "Centralized Scheduling for Joint Transmission Coordinated Multi-Point in LTE-Advanced," in *International ITG Workshop on Smart Antennas (WSA)*, Bremen, 2010.

[28] B. Mielczarek and W. Krzymien, "Influence of CSI Feedback Delay on Capacity of Linear Multi-User MIMO Systems," in *IEE Wireless Communications and Networking Conference, 2007.WCNC 2007*, Kowloon, 2007.

[29] L. Shi, Z. Hu, T. Zhang and Z. Zeng, "Performance Analysis of Delayed Limited Feedback Based on Per-cell Codebook in CoMP Systems," in *2015 IEEE Wireless Communications and Networking Conference (WCNC 2015)*, New Orleans, 2015.

[30] B. U. Kazi, G. A. Wainer, G. Boudreau and R. Casselman, "Coordinated Multi--Point (CoMP) Method And Systems Using a Coordination Station". Canada Patent P47112 US2, 2015.