

Mode-Controlled Dataflow Based Modeling & Analysis of a 4G-LTE Receiver

Hrishikesh Salunkhe
 TU Eindhoven, The Netherlands
 Email: h.l.salunkhe@tue.nl

Orlando Moreira
 ST-Ericsson, The Netherlands
 Email: orlando.moreira@ericsson.com

Kees van Berkel
 TU Eindhoven, The Netherlands
 Email: c.h.v.berkel@tue.nl

Abstract—Today’s smartphones and tablets contain multiple cellular modems to support 2G/3G/4G standards, including Long Term Evolution (LTE). They run on complex multi-processor hardware platforms and have to meet hard real-time constraints. Dataflow modeling can be used to design an LTE receiver. Static dataflow allows a rich set of analysis techniques, but is too restrictive to model the dynamic behavior in many realistic applications, including LTE receivers. Dynamic dataflow allows modeling of many realistic applications, but does not support rigorous temporal analysis. Mode-Controlled Dataflow (MCDF) is a restricted form of dynamic dataflow, and allows the same analysis techniques as static dataflow, in principle. We prove that MCDF is sufficiently expressive to handle the dynamic behavior of a realistic LTE receiver, by systematically and stepwise developing a complete MCDF model for an LTE receiver.

I. INTRODUCTION

Today’s smartphones and tablets have multiple cellular links for wireless communication. The latest cellular standard is Long Term Evolution (LTE, [1]), and will be introduced in Section II. State of the art modems (receiver + transmitter) need to support multiple standards. The involved signal and protocol processing is executed on a heterogeneous multi-processor architecture (Section II). The LTE standard specifies hard-real time constraints, including end-to-end throughput and latency constraints. Furthermore, resources are severely constrained to enable high-volume markets (low costs), and competing battery life (low power).

An LTE receiver can be modeled as a dataflow graph. In dataflow, an application is modeled as a directed graph, where nodes (actors) are processing elements and edges are data dependencies. Static dataflow is an effective programming and analysis model for streaming applications [2], [3]. It supports rigorous analysis of real-time constraints to make sure that they are always met. However, it cannot handle dynamic data-dependencies. For an LTE receiver, these dependencies follow from the standard. Hiding these dependencies by choosing a coarser, more abstract granularity is possible, but is generally limited by critical fine-grained latencies and resource allocation. Moreover, constantly evolving cellular standards with the needs to improve link performance and battery life is resulting into more and more dynamic behavior in LTE receivers.

With dynamic dataflow, e.g. Boolean Dataflow [4], it is relatively easy to capture this dynamic behavior. However, the price for this convenience is high: dynamic dataflow graphs cannot be analyzed for real-time behavior. Mode-Controlled

Dataflow (MCDF, [5]) is a restricted form of boolean dataflow. In an MCDF graph, a specific subgraph is chosen, depending on a mode selected by a so-called mode controller (Section II). In combination with a pre-specified set of so-called mode sequences, rigorous analysis of MCDF models is possible.

Key questions are: can a realistic LTE receiver be modeled accurately as an MCDF graph? And is the resulting graph manageable by the analysis tools? In Section III, we list a set of dataflow modeling challenges, mostly related to the dynamic behavior implied by the LTE standard. Most of these challenges have not been addressed in the related work (Section VI). In Section IV we address these challenges one by one, and construct a complete graph for an LTE receiver. Section V presents experiments and results.

II. PRELIMINARIES

A. LTE Receiver

In an LTE receiver, data is transmitted in terms of Radio Frames (10 msec in duration) [6]. They are divided into 10 sub-frames, each sub-frame being 1 msec long as shown in Figure 1. Each sub-frame is further divided into two slots, each of 0.5 msec. A slot may contain 6 or 7 OFDM symbols, depending on the normal or extended cyclic prefix used.

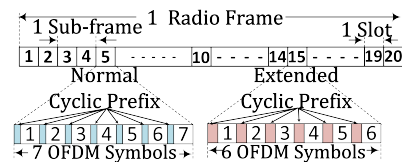


Fig. 1: LTE Radio Frame

The transmitted downlink signal consists of N_{BW} sub-carriers in a resource grid as shown in Figure 2. The value of N_{BW} depends on the system bandwidth (ranging from 1.25 to 20 MHz) and the sub-carrier bandwidth (15 kHz). A Physical Resource Block (PRB) is defined as consisting of 12 consecutive sub-carriers for one slot duration. Each location present in the resource grid is called a Resource Element, which is the basic unit of physical resources. LTE employs special Reference Signals (RS) in each resource block to facilitate channel estimation and timing synchronization. As shown in Figure 2, all the RSs are used for multiple antennas case and only highlighted RSs are used for single antenna case.

Different (control and data) types of information are mapped to different physical channels (a subset of the resource

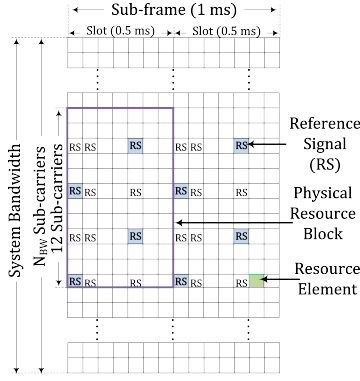


Fig. 2: LTE Resource grid structure

elements in the resource grid) in a sub-frame. We consider three physical channels: the Physical Control Format Indicator Channel (**PCFICH**), the Physical Downlink Control Channel (**PDCCH**) and the Physical Downlink Shared Channel (**PDSCH**). The PCFICH (C_1 channel) is a control channel that carries a Control Format Indicator (CFI) message which contains information about the structure and size of PDCCH. A PDCCH (C_2 channel) carries a Downlink Control Information (DCI) message which includes resource assignments and other control information for one or more User Equipments (UE). The PDSCH (D channel) is the main shared (among all UEs) data channel which carries all the user data.

B. Mode-Controlled Dataflow

In static dataflow, e.g. Synchronous Dataflow (SDF) [2], actors have fixed execution times and communicate with each other using tokens through edges (FIFO channels). In each firing, an actor consumes/produces a fixed amount of tokens from/to its input/output edges. These amounts are called as consumption/production rates. For Single Rate Dataflow (SRDF), these rates for each firing of an actor are 1. The initial state of an edge is specified by initial tokens, shown as a dot(s). Dataflow graphs are iterative, they run continuously processing virtually infinite input sequence in a pipelined manner.

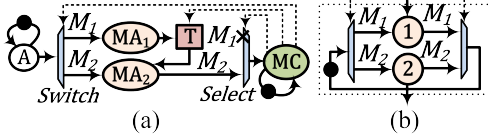


Fig. 3: MCDF: (a) Example (b) Tunnel T

Mode-Controlled Dataflow (MCDF) is a restriction of Boolean Dataflow [4] that supports mode switching as well as temporal analysis [5]. In an MCDF graph, based on a mode value produced by a so-called Mode Controller (MC), actors belonging to a specific, pre-defined sub-graph of the complete MCDF graph are fired. After all actors of the sub-graph have fired, the graph returns to the initial token distribution. A typical MCDF graph is comprised of a special actor MC, arbitrary number of Switch (SW), Select (SL) and single-rate actors as shown in Figure 3a. The MCDF graph consists of 2

modes: M_1 and M_2 . A is an amodal actor which fires for every iteration. MA_1 (MA_2) is a modal actor which fires only when mode M_1 (M_2) is fired. Switch/Select produces/consumes a token on/from the modal output/input edge which is associated with the received control token from MC. When mode M_1 (M_2) is fired by MC , a subgraph consisting of all the amodal actors, switch, select and MA_1 (MA_2) are fired. Tunnel is used for inter-modal communication. T is a tunnel that passes a token from mode M_1 to M_2 and is comprised of MCDF actors as shown in Figure 3b. When a modal port of a switch/select is unconnected then it fires with a default (saved as an internal state) token corresponding to the unconnected mode [5]. A specific mode sequence selected by MC models a specific application scenario. Tunnel sizes (in terms of tokens) can be determined through mode sequence analysis.

We generalize switches and selects by allowing assignment of a mode set to a modal edge instead of a single mode. We call modal edges and modal actors as multi-modal edges and multi-modal actors. Each multi-modal edge of a switch/select is annotated with the mode set it belongs to. Mode sets of any two edges of a switch/select are disjoint. Union of mode sets of all the edges of a switch/select is the mode set of the input graph. For each multi-modal actor, all input/output edges must have the same mode set. We use these multi-modal abstractions when multiple modes have the same behavior thereby abstracting the graph complexity.

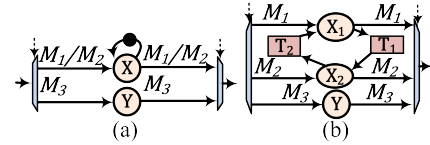


Fig. 4: Converting (a) multi-modal actors to (b) modal actors

A conversion of multi-modal MCDF graph into canonical MCDF graph is shown with an example in Figure 4. Figure 4a shows an example of a multi-modal MCDF graph. In canonical MCDF, multi-modal actor X is represented as two separate copies X_1 and X_2 for mode M_1 and M_2 respectively. Moreover, two tunnels T_1 and T_2 are added between X_1 and X_2 to model the initial token. In our modeling work, to keep the model simple, we will not show the (dotted) control edges.

C. Hardware Architecture

In general, baseband (physical layer) processing of a digital radio is carried out on a Heterogeneous Multi-Processor (HMP) platform. An HMP platform consists of a shared memory and a set of processing elements (programmable processors and accelerators) which form different baseband processing stages of the LTE Receiver. Our model partitioning assumes the architecture given in [7].

III. PROBLEM DESCRIPTION

Variable sub-frame formats: Mapping of control and data channels to the symbols in a sub-frame varies over time.
Channel Estimator Behavior: Channel estimator (ChEst) exhibits a varying token consumption pattern.
Channel estimation and decoding stage dependency: When

a symbol containing the reference signals is arrived, only then ChEst can interpolate and decode a previous symbol.

Variation in resource block allocation: The number of resource blocks in the LTE resource grid varies over time.

The resulting model must respect the following constraints:

- a)** The model must allow an actor to be scheduled (fine grained) strictly on a single processing element. **b)** The graph complexity must be manageable by the analysis tools.

IV. LTE RECEIVER MODELING

This section motivates and explains modeling of the LTE receiver using MCDF.

Variable sub-frame formats: Variation in the mapping of C_1 , C_2 and D channels to the symbols in a sub-frame give rise to different sub-frame formats. The 1st symbol always has C_1 mapped on it. C_2 may occupy the remaining part of the 1st symbol and also can occupy 2nd and even 3rd symbol. Depending on the C_2 mapping, D channel occupies the remaining symbols in a sub-frame from the 2nd or 3rd or 4th to the 14th symbol. Consequently, C_2 and D channel decoders (C_2 DEC and D DEC) consume a varying number of symbols depending on a sub-frame format. This behavior cannot be modeled correctly in Static Dataflow (StDF). For instance, in StDF, C_2 DEC must fire 3 times (the maximum number of C_2 channel symbols in any sub-frame) for every sub-frame, even if the current sub-frame contains only 1 or 2 C_2 channel symbols, often leading to a pessimistic schedule.

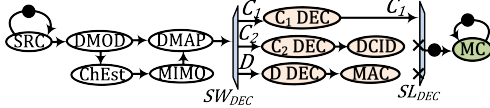


Fig. 5: LTE Receiver model: C_1 , C_2 and D modes

However, MCDF can model the variation in the consumption rates efficiently. We devise 3 modes i.e. C_1 , C_2 and D to process the respective channels as shown in Figure 5. SRC (Source) includes receiver antennas. Then the signal is demodulated by the DMOD (OFDM Demodulator). Channel estimates are computed by ChEst. MIMO (Multiple Input Multiple Output) computes the combined response in case of multiple antennas. DMAP (OFDM Demapper) demaps the symbols to softbits with the help of combined response from MIMO. Depending on the mode (type of channel) selected by MC, the data is forwarded to the appropriate decoder. DCID (DCI Done) extracts D channel information from C_2 channel. MAC is a higher layer interface (sink actor). Every firing of the graph or MC corresponds to processing a single symbol in a sub-frame and hence to process an entire sub-frame, the graph will fire 14 times i.e. with a mode sequence of length 14. E.g. a mode sequence $[C_1, C_2, C_2, D, D, D, D, D, D, D, D, D, D, D]$ represents the sub-frame having 1 C_1 , 2 C_2 and 11 D channel symbols respectively. For any sub-frame, MC is fired with the mode C_1 for the first symbol. After decoding the C_1 channel, MC determines the mode sequence for the next 13 symbols for that sub-frame (without needing the inputs from the other modes).

Channel Estimator Behavior: ChEst only consumes the symbols having the Reference Signals (RS), whose positions in any sub-frame are fixed. However, multiple cyclic prefix lengths impose 12 to 14 symbols in a sub-frame which forces the RS symbols to have varying positions in a sub-frame and ChEst to have a varying input consumption pattern. In StDF, the ChEst is forced to consume a symbol from ODEM even if the current symbol in the current sub-frame does not contain RS.

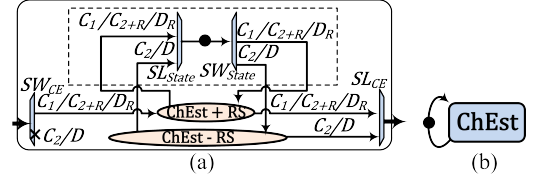


Fig. 6: ChEst behavior: (a) ChEst modeling (b) ChEst Symbol

We solve this issue by modeling the ChEst using MCDF constructs as shown in Figure 6a. We devise two new modes named C_{2+R} and D_R to handle RS symbols having C_2 and D channels respectively. If the current symbol contains RSs then depending on the channel mapping, an appropriate mode associated to ($ChEst + RS$) is selected. Otherwise, an appropriate mode associated to ($ChEst - RS$) is selected. The structure shown in a dotted rectangle allows communication between any mode pair (estimate sharing) and hence it is called a Multi-modal tunnel. In LTE Receiver model, ChEst model is abstracted using a symbol shown in Figure 6b.

Channel estimation and decoding stage dependency: RSs are only transmitted for few resource elements in the LTE resource grid (Figure 2). Hence to obtain the channel estimates for every resource element, we need to carry out interpolation along frequency direction and then along time direction. This forces the ChEst to run 6 tokens ahead of decoding stage to decode the current symbol. Moreover, fast time filtering in ChEst allows decoding the control (C) channels faster than the data (D) channels. Consequently, the ChEst stages for C and D channels run 4 and 6 symbols ahead of their respective decoding stages respectively. Since the number of C and D channel symbols vary per sub-frame, these dependencies have to be adjusted accordingly. However, as discussed earlier, StDF cannot model the varying number of C and D channel symbols accurately. Therefore, in StDF, these dependencies have to be forced for all possible C and D channel symbols.

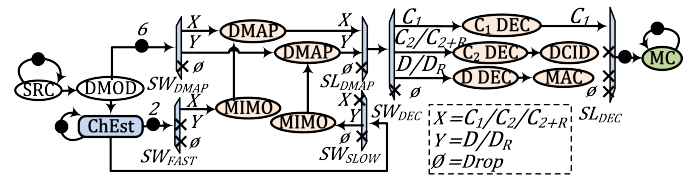


Fig. 7: LTE Receiver model: ChEst and Dec dependencies

We solve this problem by modeling these dependencies by introducing a new mode, called Drop as shown in Figure 7. Initially, for the first six firings, MC fires with the Drop mode. SW_DMAP, SW_SLOW and SW_FAST discard the six initial tokens from DMOD to SW_DMAP edge, the first six channel estimates from ChEst and two initial tokens + the first four

channel estimates from *ChEst* using drop mode respectively. This ensures that for the n^{th} symbol from *DMOD*, *DMAP* for control and data channels will receive the $(n+4)^{\text{th}}$ and the $(n+6)^{\text{th}}$ channel estimates respectively. One of the ways of handling the initial delays in the implementation is by using a counter: SW_{DMAP} sets a counter to six. In this way, the first six firings of SW_{DMAP} will not consume any token.

Variation in resource block allocation: The number of resource blocks allocated to a user equipment vary (from 6 to 100) over time depending on the channel conditions. These in turn vary execution time and energy requirements of *DDEC*. *StDF* cannot model this variation accurately. It can be modeled using *MCDF*. However, the number of modes (around 94) will make the graph complex. Moreover, these 94 modes form cartesian product with the existing modes which further explodes the graph complexity, making it hard to take advantage of these fine grained modeling. Therefore, currently we do not model the varying resource block allocation.

Sub-frame symbol assembly for decoders: To decode any channel, all the demapped symbols (by *DMAP*) must be assembled and then fed to its decoder. We solve this problem by introducing several new modes and by assembling demapped symbols using tunnels as shown in Figure 8. Since C_2 and D channels need assembling of one or more demapped symbols (except C_1 channel), they need one extra mode each, namely C_{2L} and D_L , as shown in Figure 8. These modes are fired for the last symbol occupied by its respective channel in that sub-frame. A mode L_{2L+R} is added to handle the scenario where the last symbol of C_2 channel contains RSs. For the

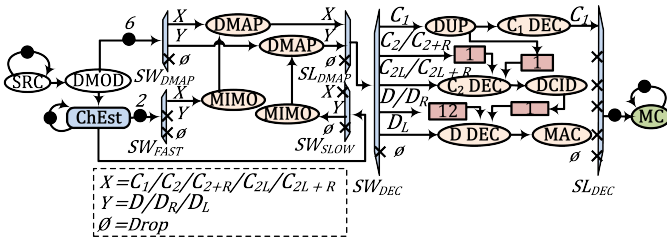


Fig. 8: LTE Receiver model: Sub-frame symbol assembly

last symbol in a sub-frame, *DDEC* is fired through mode D_L . Except the last symbol, all the demapped data channel symbols in a sub-frame (at the most 12) are produced on the D/D_R of SW_{Dec} edge. These 12 symbols are assembled from D/D_R output of SW_{Dec} and feed it to *DDEC* using a tunnel. The number present inside a tunnel depicts its capacity. Likewise, we add tunnels to accumulate symbols for C_2Dec . Moreover, the tunnel from *DCID* to *DDEC* passes the information about the data channel in a sub-frame to *DDEC*.

V. EXPERIMENTS & RESULTS

The LTE receiver model is implemented in Heracles [5], a temporal analysis tool developed at Ericsson. We carried out and compared throughput analysis between the *MCDF* and *SRDF* based models of the LTE receiver. The *SRDF* model approximates the LTE behavior by forcing (the worst-case scenario) actors firings for 3 C and 13 D channel symbols for a sub-frame. We reduce energy consumption of a processor by reducing frequency (increasing execution times)

while preserving timing guarantees. We only considered actors scheduled on *EVP* (Embedded Vector Processor) from the *HMP* platform [7]. At every stage, we increased execution times of the *ChEst* and *MIMO* (both are statically ordered on the *EVP*) by 10% till the point where the graphs cannot keep up with the source i.e. they cannot be scheduled using a finite memory. Results show that the throughput analysis for the *SRDF* model failed when the execution times are increased by 30% while for the *MCDF* model, it failed at 40%. This implies that the *MCDF* model is more accurate and produce tighter results compared to the *SRDF* model. This is the result of unnecessary firings of *EVP* based actors in the *SRDF* model.

VI. RELATED WORK

Modeling of an LTE receiver using dataflow have been attempted earlier using *FSM-SADF* [8], *MMVB* [9] and *PSDF* [10] based dataflow. *MMVB*, *PSDF* and *SPDF* are less expressive compared to *MCDF* as they require parameterized, non-zero and equal production/consumption rates for a channel. *FSM-SADF* is an analysis model whereas *MCDF* is both a programming and analysis model. Table I, shows the modeling challenges involved in LTE receiver and compare other LTE models. YES/NO/NA imply the challenge is Modeled/Not Modeled/Not Available respectively.

Modeling challenges	MCDF	FSM-SADF	MMVB	PSDF
Separate control and data channels	YES	YES	NO	YES
Separate C_1 and C_2 channels	YES	YES	NO	NA
Data decoding dependency	YES	YES	NO	NA
Control decoding dependency	YES	NO	NO	NO
Variable consumption of <i>ChEst</i>	YES	NO	NO	NO
MIMO modeling	YES	YES	YES	NA
Variable cyclic prefix lengths	YES	NO	NO	YES
Varying resource block allocation	NO	NO	NO	NO

TABLE I: Comparison of LTE models

VII. CONCLUSION

We have developed an analyzable, programmable and accurate model of a realistic LTE receiver using Mode-Controlled Dataflow.

REFERENCES

- [1] J. Zyren, "White paper overview of the 3gpp long term evolution physical layer," 2007.
- [2] E. A. Lee and et al., "Synchronous data flow," 1987.
- [3] S. S. Bhattacharyya et al, *Embedded Multiprocessors: Scheduling and Synchronization*. 2000.
- [4] J. Buck, "Static scheduling and code generation from dynamic dataflow graphs with integer-valued control streams," in *SSC*, 1994.
- [5] O. Moreira, *Temporal Analysis and Scheduling of Hard Real-Time Radios on a Multi-Processor*. PhD thesis, TU/e, Eindhoven, NL, 2011.
- [6] S. Sesia et al, *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.
- [7] M. Breschel et al., "A multi-standard 2g/3g/4g cellular modem supporting carrier aggregation in 28nm cmos," in *ISSCC*, 2014.
- [8] F. Siyoum et al., "Analyzing synchronous dataflow scenarios for dynamic software-defined radio applications," in *SoC*, 2011.
- [9] C.-J. Hsu et al., "A mixed-mode vector-based dataflow approach for modeling and simulating lte physical layer," in *DAC*, 2010.
- [10] H. Kee et al., "Fpga-based design and implementation of the 3gpp-lte physical layer using parameterized synchronous dataflow techniques," in *ICASSP*, 2010.