System Integration for LTE Wireless Design

Dolesh Bhardwaj[1], H.K.S. Randhawa[2]
[1]Department of Electronic and Communication Engineering, PG student, Thapar University, Patiala, India
[2]Department of Electronic and Communication Engineering, Thapar University, Patiala, India
Email: yasvlsi05@gmail.com, dolesh_bhardwaj@yahoo.com

***Abstract* — Long Term Evolution (LTE) is one of the emerging technologies toward next generation mobile wireless networks. For LTE physical layer development, electronic system level (ESL) tools are widely used to assist design and verification processes. Among various modeling technologies underlying ESL tools, synchronous dataflow (SDF) and its related models of computation have been successfully used to model and simulate many wireless standards. However, LTE physical layer involves dynamically varying data processing rates that make SDF insufficient due to its constant-rate constraint. In this paper, we use a novel approach, called Mixed-mode Vector-based Dataflow (MVDF), to efficiently model and simulate LTE physical layer by exploring the matched-rate nature of LTE and by combining static and dynamic dataflow technologies. We have calculated the throughput of LTE physical layer in an ESL tool, called SystemVue, along with a complete LTE physical layer library. With the implementation, we are able to create LTE reference designs for performance measurements. Our simulation results successfully match the standard requirements.**

Index Terms — Mixed-mode dataflow, LTE physical layer, 3GPP.

I. Introduction

System-level design and verification using electronic system level (ESL) tools have become an important part of development flow for communication and DSP systems. ESL tools typically support one or more design methodologies, such as high-level languages and model-based design techniques. Among various ESL design methodologies, synchronous dataflow (SDF) [11] and timed synchronous dataflow (TSDF) [12] models of computation (MoC) have been used in model-based design tools over a decade [12]. In SDF, a system is represented as a dataflow graph consisting of functional models. For each execution of a model, the numbers of data consumed and produced are restricted to be constant positive integers. This restriction benefits SDF with static scheduling capabilities and makes SDF especially suited to model DSP systems. TSDF introduces the concept of time by associating constant sampling rates into SDF. With timing semantics, TSDF is very suited to model analog and RF systems along with complex envelope technology [12]. SDF and TSDF have been successfully used together to model mixed baseband and RF physical layer systems for many wireless standards, including GSM, EDGE, WLAN, CDMA2000, DTV, WCDMA, etc. [9, 8].

3GPP Long Term Evolution (LTE) [1] is an emerging technology toward next generation mobile wireless networks and had been stable for commercial development. The effort in using SDF/TSDF to develop physical layer reference designs, however, encounters fundamental limitations due to the constant rate constraint in SDF semantics. For example, in LTE, transmission resource allocations for different data and control streams change dynamically across subframes [14]. LTE physical layer uses *adaptive modulation and coding* (AMC) to dynamically adjust modulation scheme and transport block size to adapt to varying channel condition [15, 16]. LTE also incorporates *hybrid automatic repeat request* (HARQ) into physical layer: based on a feedback signal, a HARQ process either encodes and transmits a new transport block (a sequence of bits from MAC layer) or retransmits a previously-coded and buffered transport block [15 , 16]. All of these behavior results in dynamically varying data processing rates that makes conventional usage of SDF inadequate to model LTE physical layer.

Besides dynamically varying rates, however, LTE standard is well-designed such that the number of data produced and consumed across adjacent processing modules are matched with each other. In this paper, we used *Mixed-mode Vector-based Dataflow* (MVDF) to explore such dynamic but matched rate nature of LTE using variable-size structures and applying mixed-mode operations of static (SDF, TSDF) and dynamic dataflow models.

In this work, we use SystemVue [2] as the development platform. SystemVue is an ESL tool using dataflow (mainly SDF and TSDF) technology for wireless baseband and RF system design. Our objective is to enable LTE physical layer to be naturally mapped onto dataflow graphs and provide an efficient dataflow approach for modeling and simulating LTE physical layer. The approach should co-operate naturally with the existing SDF and TSDF design flow, and allow users to develop LTE models in an intuitive way.

The organization of this paper is as follows: We review dataflow background in Section II and discuss related work in Section II. In Section IV, we analyze LTE physical layer from dataflow point of view, then present mixed-mode vector-based dataflow in Section V. We demonstrate simulation results in Section VI and conclude at the end.

II. BACKGROUND

In the dataflow modeling paradigm, the computational behavior of a system is represented as a directed graph $G = (V, E)$. A node (*model*) $v \in V$ represents a computational module or a hierarchically nested subgraph. A directed edge $e \in E$ represents a FIFO buffer from its source model src(e) to its sink model $snk(e)$. A dataflow edge e can have a non-negative integer *delay* $del(e)$, and this delay value specifies the number of initial data samples. Dataflow graphs operate based on *data-driven* execution: a model *v* can execute (*fire*) only when it has sufficient number of data samples on all of its input edges. When firing, *v* consumes certain numbers of samples from its input edges and produces certain numbers of samples on its output edges.

In Dynamic dataflow (DDF) [18], the numbers of data production and consumption can vary dynamically at run-time, and can even be 0 to change execution paths [6]. DDF requires runtime scheduling mechanism to execute and synchronize models based on the data-driven property.

In Synchronous dataflow (SDF) [11], the number of samples produced onto (consumed from) an edge *e* by a firing of *src*(*e*) (*snk*(*e*)) is restricted to be a constant positive integer that must be known at compile time; this integer is referred to as the *production rate* (*consumption rate*) of *e* and is denoted as *prd*(*e*) (*cns*(*e*)). This restriction equips SDF with compile-time capabilities such as deadlock detection, bounded memory determination, and static scheduling.

An SDF graph G = (V , *E*) has a valid schedule if it is free from deadlock and there is a positive integer solution to the vector ***x*** *in the balance equations:*

$$\forall e \in \mathrm{E},\ prd(e) \times \mathbf{x}[src(e)] = cns(e) \times \mathbf{x}[snk(e)]. \qquad (1)$$

When it exists, the minimum solution for ***x*** is called the *repetitions vector* of *G*, and is denoted by $\boldsymbol{q}_G$. For each model *v*, $\boldsymbol{q}_G[v]$ is referred to as the *repetition count* of *v*. A *valid minimal periodic schedule* is then a sequence of firings in which each model *v* is fired $\boldsymbol{q}_G[v]$ times, and the firing sequence obeys the data-driven property imposed by *G*. Timed synchronous dataflow (TSDF) [12] introduces timing concept into SDF. A TSDF graph *G* = (V , *E*) is essentially an SDF graph, where an edge $e \in E$ is associated with a sampling rate $f_s(e)$. In TSDF, flow of data samples can be viewed as sampled version of continuous-time signal by associating time stamps that increase continuously at a constant time step 1/*fs*(*e*). Given a consistent TSDF graph *G* = (V , *E*), there exists a positive constant C_G such that

$$\begin{aligned}\forall e \in E,\ f_s(e) &= prd(e) \times \boldsymbol{q}_G[src(e)] \times C_G \\ &= cns(e) \times \boldsymbol{q}_G[snk(e)] \times C_G \qquad (2)\end{aligned}$$

Once CG is defined (or equivalently, once the sampling rate for a particular edge is defined), the sampling rates for all edges can be computed. In complex envelope technology [12], a real signal x(t) is represented as

$$\begin{aligned}x(t) &= Re\{x_c(t)exp(j2\pi fct)\} \\ &= x_i(t)cos(2\pi fct) - x_q(t)sin(2\pi fct)\ , \qquad (3)\end{aligned}$$

where $x_c(t) = x_i(t) + jx_q(t)$ is defined as the complex envelope of *x*(*t*), $x_i(t)$ and $x_q(t)$ are real-valued, in-phase and quadrature components, and f_c is the characterization frequency associated with the complex envelope. Suppose *x*(*t*) represents a modulated RF signal carrying information at bandwidth Δ*w*. By setting f_c equal or close to the carrier frequency f_o of *x*(*t*) (where $f_o >> \Delta w$), the sampling rate required for $x_c(t)$ is in the range of Δ*w*, which is order of magnitude smaller than the sampling rate required for *x*(*t*).

TSDF together with complex envelope technology provide very efficient discrete-time simulation for RF systems due to the small sampling rate requirement.

III. RELATED WORK

Various dataflow MoCs have been developed to model dynamic behavior beyond the territory of SDF and its related MoCs. For example, parameterized dataflow (PDF) [3] has been proposed as a meta-modeling technique to support limited forms of reconfigurations across schedule iterations. Blocked dataflow [10] that builds on top of PDF has been presented to model block-based processing for video and image. However, PDF imposes restrictions on how applications are configured, and in general, is difficult for existing environment to adopt such configuration.

Core functional dataflow [17] is a form of dynamic dataflow that attempts to derive efficient schedules by analyzing mode transitions within and across models and by exploring fixed production and consumption rates associated with each execution mode. However, in LTE, such rates vary at runtime based on operations that dynamically adapt to indeterministic channel and network traffic. As a result, the number of possible modes is in general unmanageable, and deriving mode transitions and fixed rates for all possible modes are very difficult for LTE models.

As discussed in Section II, dynamic dataflow (DDF) [18] is the most general form of dataflow that relies on runtime scheduling and dynamic buffering and makes no attempt to statically analyze deadlock and boundedness. However, based on our analysis in Section IV, modeling LTE modules in DDF in order to process variable numbers of samples, in general, causes significant overheads and poor usability.

IV. LTE PHYSICAL LAYER ANALYSIS

In this section, we analyze LTE physical layer from dataflow point of view. The analysis is presented in the context of downlink shared channel transmission, but the same concept is applicable to both downlink and uplink, transmission and reception. For formal LTE specifications, we refer the readers to [14, 15, 16]. In LTE physical layer, downlink (base station to user equipment) and uplink transmissions are organized into radio *frames*. Each LTE frame is 10 ms in time and is divided into 10 1-ms *subframes*. A subframe is further divided into 2 0.5-ms *slot*. As shown in Figure 1, a downlink slot is described by a *resource grid* consisting of $N_{RB}^{DL}\ N_{sc}^{RB}$ subcarriers in frequency and N_{symb}^{DL} OFDM symbols in time, where N_{RB}^{DL} is the downlink bandwidth configuration in terms of number of *resource blocks*, N_{sc}^{DL} is the size of a resource block in frequency in terms of number of subcarriers, and N_{symb}^{DL} is the number of OFDM symbols in a slot, which is also the size of a resource block in time. A grid element (k, l), where $0 \leq k < N_{RB}^{DL}\ N_{sc}^{DL}$ and $0 \leq l < N_{symb}^{DL}$, is called a *resource element*.

In Figure 2, the shared channel (user data) processing for transmitting a downlink subframe starts with a *transport block* from MAC layer. A transport block, denoted as $\vec{a} = a_0, a_1, \ldots, a_{A-1}$, where A is the transport block size, is a sequence of bits to be transmitted in the shared channel to a user or a set of users. First, the CRC parity bits for $\vec{a}$ is appended, yielding a CRC attached bit vector $\vec{b} = b_0, b_1, \ldots, b_{B-1}$. Depending on the size B, $\vec{b}$ is segmented into C number of *code blocks*, and each code block is attached with additional CRC parity bits. The output from code block segmentation consists of C code blocks, $\vec{c}_0, \vec{c}_1, \ldots, \vec{c}_{C-1}$, where $\vec{c}_r = \vec{c}_{r0}, \vec{c}_{r1}, \ldots, \vec{c}_{r(Kr-1)}$ represents the rth code block with K_r bits for $r = 0, 1, \ldots, C-1$. Next, each code block $\vec{c}_r$ is coded by a rate 1/3 turbo encoder, resulting in a turbo-coded code block containing three code streams, $\vec{d}_r = \vec{d}_r^0, \vec{d}_r^1, \vec{d}_r^2$, where each code stream $\vec{d}_r^i = \vec{d}_{r0}^i, \vec{d}_{r1}^i, \ldots, \vec{d}_{r(Dr-1)}^i$ has D_r bits. The turbo coding output consists of C turbo-coded code blocks $\vec{d}_0, \vec{d}_1, \vec{d}_2, \ldots, \vec{d}_{C-1}$.

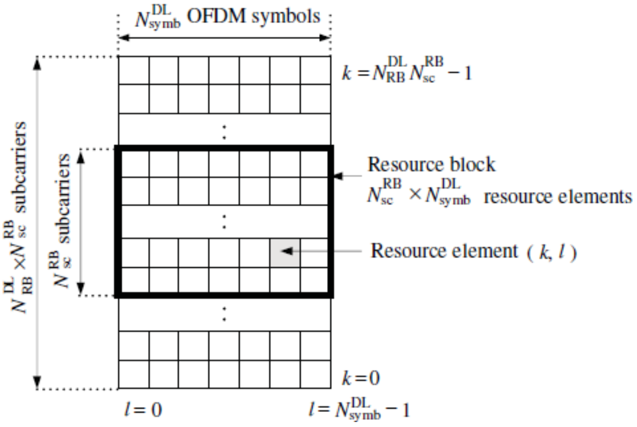
N_{symb}^{DL} OFDM symbols
$k = N_{RB}^{DL} N_{sc}^{RB} - 1$
Resource block
$N_{sc}^{RB} \times N_{symb}^{DL}$ resource elements
Resource element (k, l)
$k = 0$
$l = 0$
$l = N_{symb}^{DL} - 1$
$N_{RB}^{DL} \times N_{sc}^{RB}$ subcarriers
N_{sc}^{RB} subcarriers

Figure 1: Downlink slot resource grid [14].

In the rate matching process, each turbo-coded block $\vec{d}_r$ is interleaved and padded if necessary into a W_r bit sequence $\vec{w}_r = wr_0, wr_1, \ldots, w_r(W_r-1)$ representing a channelcoded code block ready for transmission. The C channelcoded code blocks $\vec{w}_0, \vec{w}_1, \ldots, \vec{w}_{C-1}$ are stored individually in a set of rate matching buffers $\mathbf{W}_{hqid}$ associated with the current HARQ process hq_{id} for possible HARQ retransmissions. After that, a subset of bits $\vec{e}_r = e_{r0}, e_{r1}, \ldots, e_{r(Er-1)}$ are selected from each $\vec{w}_r$ such that the total number of bits $\sum_{r=0}^{C-1} E_r$ in the output sequences $\vec{e}_0, \vec{e}_1, \ldots, \vec{e}_{C-1}$ matches the number of bits M_{bit} that is available for transmitting the shared channel in the current subframe. Finally, ratematched code blocks $\vec{e}_0, \vec{e}_1, \ldots, \vec{e}_{C-1}$ are concatenated into a M_{bit}-bit *code word* $\vec{f} = f_0, f_1, \ldots, f_{M_{bit}-1}$.

LTE downlink supports one or two code words to be transmitted in parallel. In Figure 3, let $\vec{f}^q = f_0^q, f_1^q, \ldots, f_{M_{bit}^q-1}^q$ denotes the qth code word $q \in \{0\}$ or $q \in \{0, 1\}$, generated from Figure 2. Code word $\vec{f}^q$ is first scrambled into $\vec{g}^q = g_0^q, g_1^q, \ldots, g_{M_{bit}^q-1}^q$Next, each consecutive Q_m bits in $\vec{g}^q$ is mapped to a complex-valued modulation *symbol*, yielding a modulated code word $\vec{m}^q = m_0^q, m_1^q, \ldots, m_{M_{symb}^q-1}^q$ with M_{symb}^q symbols. $Q_m \in \{2, 4, 6\}$ (corresponding to modulation schemes, QPSK, 16QAM, and 64QAM, respectively) is the modulation order scheduled for the shared channel in the current subframe.

Next, the modulated code words $\{\vec{m}^q\}$ are mapped to V number of *layers*. Each layer $\vec{x}^v = x_0^v, x_1^v, \ldots, x_{M_{symb}^{layer}-1}^v$, $v = 0, 1, \ldots, V-1$, contains M_{symb}^{layer} symbols. The layer mapping process results in a sequence $\vec{\mathbf{x}} = \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{M_{symb}^{layer}-1}$ of M_{symb}^{layer} *layered vectors*, where $\mathbf{x}_i = [x_i^0, x_i^1, \ldots, x_i^{V-1}]^T$ is the ith layered vector for $i = 0, 1, \ldots, M_{symb}^{layer} - 1$. After layer mapping, the precoding process maps V layers onto P *antenna ports*. In the precoding, each $\mathbf{x}_i = [x_i^0, x_i^1, \ldots, x_i^{V-1}]^T$ is mapped to $\mathbf{y}_i = [y_i^0, y_i^1, \ldots, y_i^{P-1}]^T$ using a P-by-V precoding matrix, yielding P sequences $\vec{y}^0, \vec{y}^1, \ldots, \vec{y}^{P-1}$, where each sequence $\vec{y}^p = y_0^p, y_1^p, \ldots, y_{M_{symb}^{ap}-1}^p$, $p = 0,1,\ldots,P-1$, represents M_{symb}^{ap}

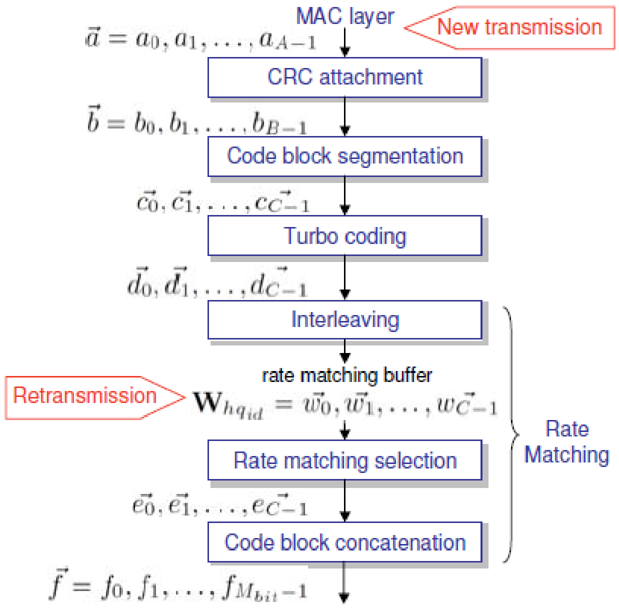
MAC layer
New transmission
$\vec{a} = a_0, a_1, \ldots, a_{A-1}$
CRC attachment
$\vec{b} = b_0, b_1, \ldots, b_{B-1}$
Code block segmentation
$\vec{c_0}, \vec{c_1}, \ldots, \vec{c_{C-1}}$
Turbo coding
$\vec{d_0}, \vec{d_1}, \ldots, \vec{d_{C-1}}$
Interleaving
rate matching buffer
Retransmission
$\mathbf{W}_{hq_{id}} = \vec{w_0}, \vec{w_1}, \ldots, \vec{w_{C-1}}$
Rate matching selection
$\vec{e_0}, \vec{e_1}, \ldots, \vec{e_{C-1}}$
Code block concatenation
$\vec{f} = f_0, f_1, \ldots, f_{M_{bit}-1}$
Rate Matching

Figure 2: Transport downlink shared channel processing [15].

number of symbols to be transmitted on antenna port *p* for the shared channel.

For each antenna port *p*, $\vec{y}^p = y_0^p, y_1^p, \ldots, y^p_{M^{ap}_{symb}-1}$ are mapped to resource elements that are allocated for the shared channel in the current subframe. The resource mapping process also maps other channels and signals onto designated resource elements, yielding a resource grid **sf**^{p} containing $N_{RB}^{DL} N_{sc}^{RB} \times 2N_{symb}^{DL}$ resource elements for a subframe (2 slots) on port *p*. On antenna port *p*, the baseband OFDM signal $s_l^p(t)$ for the *l*th OFDM symbol is generated by inverse Fourier transform and cyclic prefix of the *l*th column in the resource grid **sf**^{p}. Finally, $s_l^p(t)$ is modulated onto carrier frequency f_0, resulting in a passband LTE signal $Re\{s_l^p(t)\}cos(2\pi f_0 t) - Im\{s_l^p(t)\}sin(2\pi f_0 t)$.

In LTE, multiple *HARQ processes* work alternatively to transmit data in different subframes. Suppose a *HARQ process* hq_{id} transmits a new transport block. After receiving the subframe, the designated receiver sends either an acknowledgement (ACK) or a negative acknowledgement (NACK) back to hq_{id} based on whether it can successfully decode the received transport block. When process hq_{id} takes turn again, if the feedback is ACK, it transmits a new transport block from MAC layer. Otherwise, it retransmits the transport block by selecting $\vec{e}_0, \vec{e}_1, \ldots, \vec{e}_{C\text{-}1}$ from the rate matching buffer $\mathbf{W}_{hqid}$ (see Figure 2), and the selection is subject to M_{bit} and the redundancy version rv_{idx} scheduled for the retransmission subframe. This round-trip HARQ communication must be performed in designated subframes [16], and that defines the latency of the LTE system.

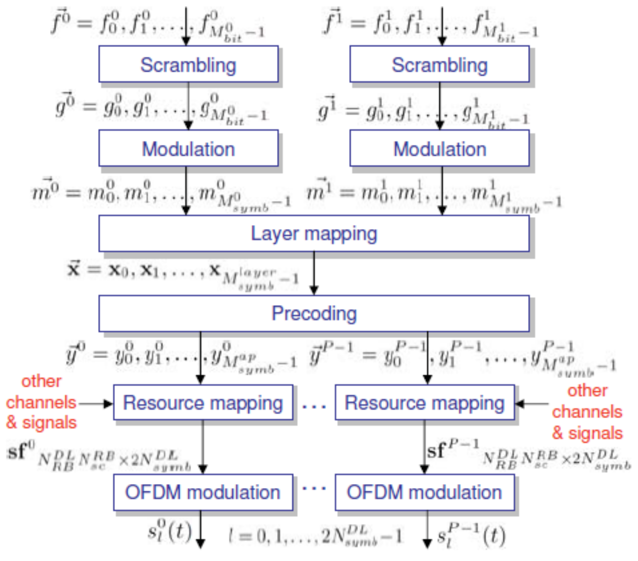
$\vec{f^0} = f_0^0, f_1^0, \ldots, f^0_{M^0_{bit}-1}$
$\vec{f^1} = f_0^1, f_1^1, \ldots, f^1_{M^1_{bit}-1}$
Scrambling
Scrambling
$\vec{g^0} = g_0^0, g_1^0, \ldots, g^0_{M^0_{bit}-1}$
$\vec{g^1} = g_0^1, g_1^1, \ldots, g^1_{M^1_{bit}-1}$
Modulation
Modulation
$\vec{m^0} = m_0^0, m_1^0, \ldots, m^0_{M^0_{symb}-1}$
$\vec{m^1} = m_0^1, m_1^1, \ldots, m^1_{M^1_{symb}-1}$
Layer mapping
$\vec{\mathbf{x}} = \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{M^{layer}_{symb}-1}$
Precoding
$\vec{y^0} = y_0^0, y_1^0, \ldots, y^0_{M^{ap}_{symb}-1}$
$\vec{y}^{P-1} = y_0^{P-1}, y_1^{P-1}, \ldots, y^{P-1}_{M^{ap}_{symb}-1}$
other channels & signals
Resource mapping
Resource mapping
other channels & signals
$\mathbf{sf}^0$ $N_{RB}^{DL} N_{sc}^{RB} \times 2N_{symb}^{DL}$
$\mathbf{sf}^{P-1}$ $N_{RB}^{DL} N_{sc}^{RB} \times 2N_{symb}^{DL}$
OFDM modulation
OFDM modulation
$s_l^0(t)$
$l = 0, 1, \ldots, 2N_{symb}^{DL} - 1$
$s_l^{P-1}(t)$

Figure 3: Physical Downlink Shared Channel Processing [14].

In LTE AMC, a base station uses channel quality information reported by user equipments to dynamically adjust modulation scheme Q_m and transport block size *A* through the index of modulation and coding scheme I_{MCS}. For example, under good channel condition, higher order modulation scheme and larger transport block size are used to transmit more bits with less redundancy.

In practice, N_{RB}^{DL}, N_{sc}^{RB}, and N_{symb}^{DL} are fixed once the system is configured. However, due to HARQ, AMC, different resource block allocations, and different resource element assignment for higher-precedence channels, the numbers of data processed by LTE modules, e.g., *A*, *B*, M_{bit}^q, N_{symb}^q, etc., vary dynamically across different subframes. As a result, SDF is insufficient to model LTE physical layer.

V. MIXED-MODE VECTOR-BASED DATAFLOW

In LTE, even though the numbers of data processed vary dynamically at runtime, standards are well-defined in a way that for a subframe, the numbers of data produced and consumed between adjacent processing modules are properly matched. Figure 4 illustrates the dataflow representation of Figure 2, where the variables, *A*, *B*, *K*, *D*, and M_{bit}, annotated on each side of a connection denote the numbers of data produced and consumed in a subframe, where K = $\sum_{r=0}^{C-1} K_r$ and *D* = $\sum_{r=0}^{C-1} 3K_r$.

To efficiently model such dynamic but matched rate behavior, MVDF uses *variable-size structures*, such as *vectors*, *matrices*, or in general, any form of data structure, to encapsulate variable numbers of data samples transferred between adjacent matched-rate modules. Under this approach, an LTE module is modeled to process a vector of bits (or symbols) at a time. The vector size can change dynamically at runtime, and that forces the LTE model to dynamically adjust its behavior based on the input vector size. From dataflow point of view, this *variable-size vector-based* modelling approach results in SDF models with constant production and consumption rates at the granularity of vector structure. The bottom path in Figure 5 illustrates the SDF representation of Figure 2 under vector-based processing.

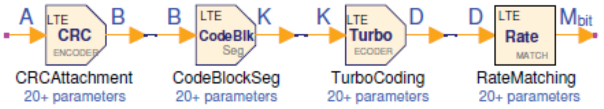

Figure 4: Sample-based channel coding models.

For LTE, and in general, wireless communication types of applications, this variable-size vector-based modeling approach fits well with how DSP algorithm designers develop the functional modules. For example, on a DSP processor, the CRC attachment in Figure 2, is commonly implemented as a C function that simply takes a pointer to a bit array and the size of the array as inputs, which is conceptually equivalent to a vector structure.

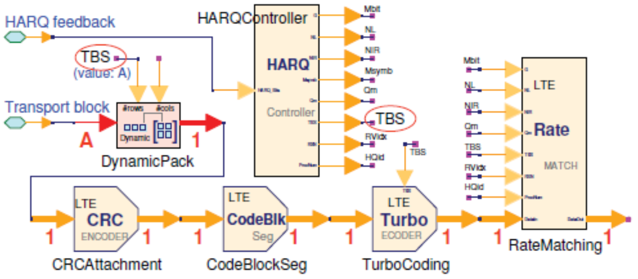

Figure 5: Vector-based channel coding models.

Traditional SDF and DDF approaches require models to compute production and consumption rates at the granularity of individual samples (e.g., bits, symbols) for scheduling purpose. However, for most LTE models, the determination of such rates is in general outside the control of the model and require many additional parameters that are irrelevant to the core functionality. For example, the sample-based CRC Attachment model in Figure 4 requires at least 20 irrelevant parameters to just compute *A* and *B*. In contrast, the vector-based CRC Attachment in Figure 5 requires just a single parameter for the generator polynomial, and can take a vector of bits with arbitrary size and produces the resulting vector by appending the parity bits. In our development of LTE library, we found that vector-based modelling approach largely improves usability and simulation efficiency.

A potential issue arises when connecting variable-size vector-based models with constant-rate, sample-based models. MVDF adopts dynamic dataflow (DDF) to dynamically pack and unpack variable numbers of data to and from variable-size structures. In this work, two special DDF models *DynamicPack* and *DynamicUnpack* are developed to enable such variable-rate conversions between individual samples and *variable-size matrics* (in our implementation, vector is represented as one-dimensional matrix).

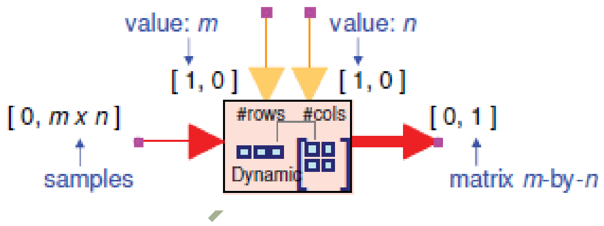

Figure 6: Dynamic Pack

DynamicPack, as shown in Figure 6, consists of two phases of executions. In the first phase, it reads one sample from port *#rows* and one sample from port *#cols*, and the values specify the number of rows *m* and the number of columns *n* respectively for the matrix to be packed. In the second phase, it reads *m* × *n* number of samples from input, then packs them into a *m*-by-*n* output matrix. Figure 6 illustrates the production and consumption rates of DynamicPack in the combined form of CSDF [5] (for modelling phased rate behaviour) and DDF. DynamicUnpack, on the other hand, performs reverse operation that reads a *m*-by-*n* matrix in the first mode and writes *m* × *n* samples in the second mode.

Figure 7 shows the top level schematic for LTE downlink 2x2 MIMO throughput measurement using MVDF. DL_Tx in Figure 7 represents a subnetwork of physical layer baseband transmitter models, which covers Figure 2, Figure 3, and processing for other channels and signals. The baseband complex-valued signals $s^0(t)$ and $s^1(t)$ generated from DL_Tx are modulated, transmitted through fading and noisy channel, and then demodulated using TSDF models in complex envelope format. DL_Rx represents a subnetwork of baseband receiver models. The upper feedback path in Figure 7 models the feedback of HARQ signal from DL_Rx to DL_Tx, and the delay component takes care of the roundtrip subframe delays and the alternation of HARQ processes.

Figure 5 shows the schematic inside the DL_Tx subnetwork to model Figure 2. The HARQ feedback path in Figure 7 goes into the HARQ Controller. If the HARQ signal is ACK for a particular HARQ process, the HARQ Controller computes the transport block size (TBS) *A* for the Dynamic Pack to pack *A* number of bits into a vector. The transport block is then processed by the following models in vector format. If the HARQ signal is NACK, the HARQ Controller informs the Rate Matching to start retransmission, and also sets TBS to 0 to force DynamicPack to hold the incoming bits and output empty vector that stops the processing for CRC Attachment, CodeBlockSeg, and TurboCoding.

The variable-size vector-based processing inside the DL_Tx subnetwork starts from CRC Attachment and ends at the Resource Mapping model (see Figure 3). After Resource Mapping, signal is represented in samples because Resource Mapping outputs fixed $N_{RB}^{DL} N_{sc}^{RB} \times 2N_{symb}^{DL}$ number of resource elements for a 1 ms subframe. The conversion back to fixed rate, sampled-based processing allows TSDF to model the RF transmission in Figure 7 in complex envelope format.

By adopting MVDF, LTE physical layer processings can be constructed naturally in dataflow graphs, and that map one-to-one to the block diagrams in the standard [14, 15].

Using MVDF to model LTE physical layer, most models fall in SDF (TSDF) domain, and only a few DDF models are required for interfacing upper layer data streams. To enable different domains of models to coexist in a unified simulation environment, we developed a model. The model focuses on simulation efficiency and strategically combines static

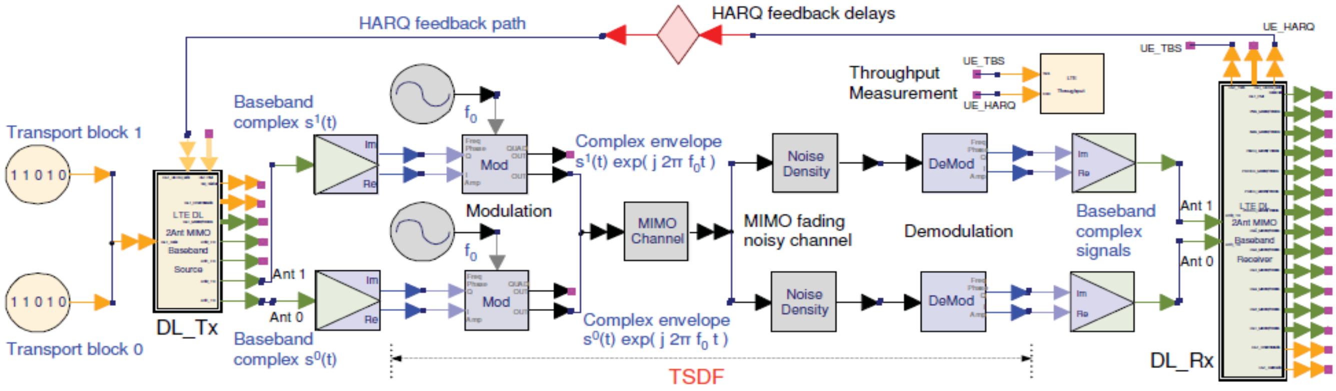

Figure 7: LTE downlink 2x2 MIMO throughput

clustering [4] for SDF models, static scheduling for SDF clusters, and runtime scheduling for the remaining dynamic portion of the graph.

Given a mixed-mode (SDF, DDF) dataflow graph, the idea is to cluster as many SDF models as possible such that 1) the clustering does not introduce deadlock — our model is carefully designed to handle cycles and multirate systems [9], and 2) the clustered subgraph does not contain any DDF model — because DDF models are left for runtime scheduling. The clustered SDF subgraph is statically scheduled using the combination of strategies developed in [7], and the buffer memories inside the subgraph are allocated statically. After clustering, the remaining graph is subject to runtime scheduling (RS) based on the data-driven property. In simulation, firing of a supernode (cluster) in the RS graph corresponds to executing a minimum periodic schedule of the clustered subgraph. The buffer memories in the RS graph are adjusted at runtime if necessary. The runtime scheduler is modified from the work in [8] to take dynamic dataflow rates and simulation efficiency into account.

VI. SIMULATION RESULTS

We have calculated the throughput in SystemVue and developed a complete set of LTE physical layer library using variable-size vector-based modeling approach. With MVDF as an anabling technology, we are able to create reference solutions for generating LTE signal and measuring physical layer performance based on the standard specifications [13].

In wireless communication, due to channel impairments such as fading, noise, and interference, received transport blocks might not be decoded correctly. Physical layer developers are free to employ algorithms such as channel estimation to compensate the impairments, thus, improve the chance of decoding success and reduce the needs for HARQ retransmission. Throughput, measured as the percentage of the number of successfully decoded transport block bits over the total number of transmitted transport block bits, is an important performance metric. LTE standard, however, specifies minimum throughput requirements that the end products must meet under various test scenarios [13]. Table 1 shows LTE downlink throughput requirements and simulation results for 7 different test cases. Columns 1 through 8 specify the settings, and column 9 specifies the minimum throughput requirement for each test case according to [13]. For accurate throughput simulation, LTE physical layer dynamic behavior such as HARQ must be modeled properly. The throughput simulation results from our implementation are shown in the last column, which closely match or outperform the minimum requirements. The ability to perform such simulation justifies the capability of MVDF.

Table 1: LTE downlink throughput simulation results.

Tx Mode	Bandwidth	# Rsc.	Mod.	Code	Propagation	Corr. Mtx. &	SNR	Minimum	Simulation
	(MHz)	Blocks	Scheme	Rate	Condition	Ant. Conf.	(dB)	Throughput	Throughput
Mode 1	1.4	6	QPSK	1/3	EVA5	1x2 Low	-0.5	70%	70%
Mode 1	10	50	16QAM	1/2	EVA5	1x2 Low	6.7	70%	81%
Mode 2	10	50	16QAM	1/2	EVA5	2x2 Medium	6.8	70%	88%
Mode 3	10	50	QPSK	1/3	EVA5	4x2 Low	-3.4	70%	95%
Mode 3	10	50	QPSK	1/3	EVA5	2x2 Low	-2.5	70%	96%
Mode 4	10	50	16QAM	1/2	ETU70	2x2 Low	14.3	70%	75%
Mode 4	10	50	16QAM	1/2	EVA5	2x2 Low	12.9	70%	71%

Simulation settings: downlink, frequency division duplex (FDD), 3000 subframes. Transmission mode: 1 single antenna port, 2 trans transmit diversity, 3 closed-loop single-layer spatial multiplexing, and 4 closed-loop multi-layer spatial multiplexing.

VII. CONCLUSION

In this paper, we have discussed the challenges in modelling LTE physical layer using traditional SDF approach. We have analyzed LTE physical layer from dataflow point of view. Based on the analysis, we have then used Mixed mode Vector-based Dataflow (MVDF) as an enabling technology that uses variable-size, vector-based

SDF models to model physical layer baseband processings, and sampled based TSDF models to model RF transmission in complex envelope, along with special DDF models to interface upper layer data streams at the physical layer boundary. We have implemented LTE physical layer library in SystemVue, and demonstrated the throughput simulation results. We have shown the efficiency and usability of using MVDF to model and simulate LTE physical layer and believe it can be applied to similar wireless systems.

VIII. REFERENCES

[1] 3GPP. *LTE*. http://www.3gpp.org/article/lte.

[2] Agilent EEsof. *SystemVue*. http://www.agilent.com/find/eesof-systemvue.

[3] B. Bhattacharya and S. S. Bhattacharyya. Parameterized dataflow modeling for DSP systems. *IEEE Transactions on Signal Processing*, 49(10):2408–2421, October 2001.

[4] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee. *Software Synthesis from Dataflow Graphs*. Kluwer Academic Publishers, 1996.

[5] G. Bilsen, M. Engels, R. Lauwereins, and J. A. Peperstraete. Cyclo-static dataflow. *IEEE Transactions on Signal Processing*, 44(2):397–408, February 1996.

[6] J. T. Buck. Scheduling dynamic dataflow graphs with bounded memory using the token flow model. Ph.D. Thesis UCB/ERL 93/69, Dept. of EECS, U. C. Berkeley, 1993.

[7] C. Hsu. *Dataflow Integration and Simulation Techniques for DSP System Design Tools*. PhD thesis, Department of Electrical and Computer Engineering, University of Maryland, College Park, April 2007.

[8] C. Hsu, J. L. Pino, and S. S. Bhattacharyya. Multithreaded simulation for synchronous dataflow graphs. In *Proceedings of the Design Automation Conference*, Anaheim, CA, 2008.

[9] C. Hsu, S. Ramasubbu, M. Ko, J. L. Pino, and S. S. Bhattacharyya. Efficient simulation of critical synchronous dataflow graphs. In *Proceedings of the Design Automation Conference*, pages 893–898, San Francisco, CA, July 2006.

[10] D. Ko and S. S. Bhattacharyya. Modeling of block-based DSP systems. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 40(3):289–299, July 2005.

[11] E. A. Lee and D. G. Messerschmitt. Synchronous dataflow. *Proceedings of the IEEE*, 75(9):1235–1245, September 1987.

[12] J. L. Pino and K. Kalbasi. Cosimulating synchronous DSP applications with analog RF circuits. In *Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 1998.

[13] 3GPP TS 36.101 V8.6.0. User Equipment (UE) Radio Transmission and Reception, June 2009.

[14] 3GPP TS 36.211 V8.6.0. *Physical Channels and Modulation*, March 2009.

[15] 3GPP TS 36.212 V8.6.0. *Multiplexing and Channel Coding*, March 2009.

[16] 3GPP TS 36.213 V8.6.0. *Physical Layer Procedures*, March 2009.

[17] N. Sane W. Plishker and S. S. Bhattacharyya. Mode grouping for more effective generalized scheduling of dynamic dataflow applications. In *Proceedings of the Design Automation Conference*, San Francisco, CA, July 2009.

[18] G. Zhou. Dynamic dataflow modeling in ptolemy II. Master's thesis, U.C. Berkeley, December 2004.