The Open-Source SDR LTE Platform for First Responders

Software Radio Systems
www.softwareradiosystems.com
www.github.com/srslte
Outline

• SRS - Software Radio Systems

• NIST PSIAP and OpenFirst

• srsLTE – The Open-Source LTE Software Suite
Outline

• SRS - Software Radio Systems

• NIST PSIAP and OpenFirst

• srsLTE – The Open-Source LTE Software Suite
Paul Sutton
Director

Ismael Gomez
Director

Andre Puschmann
Senior Engineer

Linda Doyle
Director

Justin Tallon
Senior Engineer

Xavier Arteaga
Senior Engineer

Bastian Bloessl
Senior Engineer

Pedro Alvarez
Senior Engineer
Software Radio
A software radio is a radio system where components such as modulators/demodulators, filters, detectors etc. which have historically been implemented in hardware, are instead implemented in software upon a general-purpose processing platform. Software Radio Systems (SRS) provides high-performance software radio solutions for a range of wireless technologies, leveraging the low-cost and availability of general-purpose hardware to rapidly deliver powerful, flexible systems.

Satellite systems
SRS is developing solutions for a number of satellite communications projects in partnership with the European Space Agency and international collaborators. We provide high-performance software libraries for technologies including DVB-S2 and DVB-RCS2 and integration solutions for hybrid satellite/terrestrial networks.

LTE/ LTE-Advanced
SRS provides L1, L2 and L3 protocol stacks for UE and eNodeB. Our modular, portable srsLTE C library supports custom system design and development on a range of SDR platforms. Our tailored solutions include network test and diagnostic equipment and wide-band LTE scanners. We specialise in custom design and development for demanding applications and environments including machine to machine, airborne and high-speed deployments.

Internet of Things
In the area of machine to machine communications, SRS develops custom low-cost, low-energy waveforms and network protocols. Our SDR basestation designs afford the flexibility to support a diverse range of application domains using a wide range of frequency bands. We specialise in cognitive radio and dynamic spectrum access solutions designed to benefit from evolving regulatory policies in this area.
### srsLTE

**srsLTE** is a free and open-source LTE software suite developed by SRS ([www.softwareradiosystems.com](http://www.softwareradiosystems.com)).

It includes:

- **srsUE** - a complete SDR LTE UE application featuring all layers from PHY to IP
- **srsENB** - a complete SDR LTE eNodeB application
- A highly modular set of common libraries for PHY, MAC, RLC, PDCP, RRC, NAS, S1AP and GW layers.

srsLTE is released under the AGPLv3 license and uses software from the OpenLTE project ([http://sourceforge.net/projects/openlte](http://sourceforge.net/projects/openlte)) for some security functions and for RRC/NAS message parsing.
"OSMOSIS"

**OPTIMISATION OF STREAMED MEDIA OVER SATELLITE INFRASTRUCTURES**
SRS partners with SmartSky Networks to deliver true 4G inflight connectivity

Software Radio Systems (SRS) today announced a strategic partnership with SmartSky Networks, a high-performance air-to-ground connectivity network operator, in which SRS will provide test and validation solutions for SmartSky’s airborne products. Based on aviation-specific modifications to 4G wireless communications standards, SmartSky 4G delivers affordable and reliable office-like connectivity in the air. As a leading provider...

Details

15th February 2017 / Press / By Paul Sutton

Cork tech company to deliver 4G in the sky at Mobile World Congress

by John Kennedy

24 FEB 2017    133 SHARES
Outline

• SRS - Software Radio Systems

• NIST PSIAP and OpenFirst

• srsLTE – The Open-Source LTE Software Suite
Public Safety Innovation Accelerator Program

Welcome to the PSCR Program
PSCR provides research, development, testing, and evaluation to foster nationwide interoperability and advanced communications technology for the nation’s public safety community.

PSCR’s Innovation Accelerator Program Announces $30 million Grant Program to ‘Pull the Future Forward’ for Public Safety
NIST Awards $38.5 Million to Accelerate Public Safety Communications Technologies

June 13, 2017

BOULDER, Colo. – The U.S. Commerce Department’s National Institute of Standards and Technology (NIST) has awarded $38.5 million to 33 research and development (R&D) projects aimed at advancing broadband communications technologies for first responders.
An open-source end-to-end LTE network for public safety research & development.

- A reference implementation of key LTE features for first responders.
- Enabling, supporting and growing the public safety broadband development ecosystem.
- Providing a commercialization path for public safety LTE using proven business models.
- Building upon the proven srsLTE suite of open-source libraries, tools and applications.
An open-source end-to-end LTE network for public safety research & development.

Key Requirements:
- Ease of use / Ease of programming new capabilities
- Clarity and completeness of documentation
- Long-term sustainability / Development ecosystem
- Availability for follow-on research / Potential for commercialization

R&D Platform Requirements:
- Sufficiently complete implementation / Interoperability
- Sufficient set of components
- Performance comparable to likely operational implementations
- Network scale
- Path towards operational implementation
• GNU Affero General Public License (AGPLv3)
• Ensuring dissemination of the technology
• Maximizing usability
• Safeguarding availability
• Guaranteeing sustainability

www.github.com/srslte
**Complete end-to-end LTE network:**

- Core network (EPC)
- Basestations (eNodeBs)
- Mobile terminals (UEs)

**Key features for public safety:**

- Priority, Pre-emption and Quality of Service (QPP).
- Proximity Services (ProSe)
- Evolved Multimedia Broadcast/Multicast Service (eMBMS)
- Multimedia broadcast single frequency networks (MBSFN)
- Single cell point-to-multipoint (SC-PTM).
- Carrier aggregation.
- IP Multimedia Subsystem (IMS) and Voice-over-LTE (VoLTE)
Outline

- SRS - Software Radio Systems
- NIST PSIAP and OpenFirst
- srsLTE – The Open-Source LTE Software Suite
By the end of March 2017 GSA reports there were:

- **774** operators investing in LTE in **202** countries
- **591** commercially launched LTE or LTE-Advanced networks in **189** countries (GSA forecasts **646** commercial LTE networks by end-2017)
- **97** LTE-TDD (TD-LTE) networks launched in **56** countries
- **106** commercial VoLTE networks in **55** countries, and **167** operators investing in VoLTE in **74** countries
- **195** launched LTE-Advanced or LTE-Advanced Pro networks, in **95** countries
- **4** NB-IoT and **2** LTE-M networks commercially launched, with **40** NB-IoT and **12** LTE-M networks planned or being trialled
- **18** operators (at least) that have made public commitments to deploy pre-standards '5G' networks, in **13** countries

* Taken from the GSA report Evolution from LTE to 5G April 2017. To download the full report, visit www.gsa.com

---

**Global Subscriptions by Technology**

- GSMA
- CDMA
- WCDMA incl. HSPA
- LTE

*Source: Strategy Analytics, May 2012*
<table>
<thead>
<tr>
<th>PRB</th>
<th>0</th>
<th>11</th>
<th>12</th>
<th>23</th>
<th>24</th>
<th>35</th>
<th>36</th>
<th>47</th>
<th>48</th>
<th>59</th>
<th>60</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subcarrier</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Radio frame n (system frame number n = 0,1023)</th>
<th>Subframe 0</th>
<th>Subframe 1</th>
<th>Subframe 2</th>
<th>Subframe 3</th>
<th>Subframe 4</th>
<th>Subframe 5</th>
<th>Subframe 6</th>
<th>Subframe 7</th>
<th>Subframe 8</th>
<th>Subframe 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot 0</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
</tr>
<tr>
<td>Slot 1</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
<td>Sym</td>
<td>0</td>
</tr>
</tbody>
</table>

Legend:
- **PSCH** (Primary Synchronization Channel)
- **SSCH** (Secondary Synchronization Channel)
- **PBCH** (Physical Broadcast Channel)
- **RS** (cell-specific Reference Signal) for selected Tx antenna port
- **Reserved for TDD uplink**
- **Unused by selected Tx antenna port, or undefined for all ports**
- **MBSFN** (Multicast/Broadcast over Single Frequency Network) region
- **PCFICH** (Physical Control Format Indicator Channel)
- **PHICH** (Physical Hybrid ARQ (Automatic Repeat reQuest) Indicator Channel)
- **PDCCH** (Physical Downlink Control Channel)
- **Available for PDSCH** (Physical Downlink Shared Channel)
- **TDD guard period in special subframe**
- **Available for PMCH** (Physical Multicast Channel)
LTE Uplink Transmitter (UE-side)

ULSCH
- Transport Block CRC
- Segmentation CRC
- Turbo Coding
- Interleaving & Rate Matching
- Code Block Concatenation
- Data & Control Multiplexing

Channel Interleaver

PUSCH Format Processing
- CQI
- RI
- HI

UCI on PUSCH

UCI on PUCCH
- HI
- SR
- CQI
- CQI + HI

Transport Channels

Physical Channels

Scrambling
- Modulation: QPSK, 16QAM, 64QAM
- Transform Precoding

Subframe Generation

SCFDMA Modulation

Base Sequence Generator
- Sequence Generator
- Sequence Generator
- Sequence Generator

DRS for PUSCH

DRS for PUCCH

SRS
LTE Overview – protocol stacks

- NAS
- IP
- RRC
- PDCP
- RLC
- MAC
- PHY
LTE Overview – protocol stacks
LTE Overview – protocol stacks

User

- PHY
- MAC
- RLC
- PDCP
- RRC
- IP
- NAS
LTE Overview – protocol stacks

Physical Layer

- Tx/Rx across air interface
- Very flexible, many multiplexed signals
- OFDM / SC-FDMA
LTE Overview – protocol stacks

Medium Access Control Layer

- Prioritize and multiplex logical channel data
- Scheduling
- Link adaptation
  - modulation scheme and coding rate
  - transport block size

Hybrid Automatic Repeat reQuest (HARQ)
Radio Link Control Layer

- Segment and concatenate packets
- Error correction through ARQ
- In-sequence packet delivery
- Three modes:
  - Transparent (TM)
  - Unacknowledged (UM)
  -Acknowledged (AM)
Packet Data Convergence Protocol Layer

- IP packet header compression (RoHC)
- Ciphering of control and data plane traffic
- Integrity protection of control plane traffic
Radio Resource Control Layer

• Control plane functions:
  • Paging
  • Management of RRC connection with eNodeB
  • Mobility management
  • QoS management
LTE Overview – protocol stacks

Non-Access Stratum Layer

- Control plane connection UE <-> MME:
  - Network attach/detach
  - Authentication
  - Security mode management
  - Identity management
  - Tracking area updates
  - Bearer management
  - PDN connectivity management
LTE Overview – protocol stacks

User plane protocol stacks
LTE Overview – protocol stacks

Control plane protocol stacks
PHY-layer DSP library - srsLTE
UE application - srsUE
eNodeB application - srsENB
Class/Layer Design

• srsUE/srsENB written in C++, srsLTE in C

• Each layer (PHY, MAC, RLC, PDCP, GW) is implemented in a single class
  – Some complex layers use auxiliary sub-classes

• Each layer provides a separate clean C++ pure virtual interface to any other class that make use of it (e.g. passing messages/data between layers)

• Threads only for performance or priority management reasons
Processing Latency Constraint in LTE

In LTE the basic time unit is 1 ms = 1 subframe

The processing latency constraint or *critical time* is 4 ms, given by:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>$N+1$</td>
<td>$N+2$</td>
<td>$N+3$</td>
<td>$N+4$</td>
</tr>
</tbody>
</table>

a)  
$N$: Reception of DL grant through PDCCH
$N$: Decoding of PDSCH
$N+4$: Transmission of ACK/NACK HARQ through PUCCH
In LTE the basic time unit is 1 ms = 1 subframe

The processing latency constraint or *critical time* is 4 ms, given by:

<table>
<thead>
<tr>
<th>$N$</th>
<th>$N+1$</th>
<th>$N+2$</th>
<th>$N+3$</th>
<th>$N+4$</th>
</tr>
</thead>
</table>

b) N: Reception of UL grant or NACK HARQ
N+4: (re)-Transmission of PUSCH
Processing Latency Constraint in LTE

BUT worse case happens in case of a union of a) and b), then:

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>N+1</th>
<th>N+2</th>
<th>N+3</th>
<th>N+4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a+b)</td>
<td>N:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reception of DL grant + UL grant or NACK HARQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>N:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decoding of PDSCH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>N+4:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(re)-Transmission of PUSCH + ACK/NACK HARQ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this case, we have 4 ms to receive samples from ADC, decode PDSCH, encode PUSCH and transmit samples to ADC in time.
DL and UL grants received through PDCCH

DL grant only, ACK through PUCCH

UL grant only

ADC RX

ADC RX

ADC RX

ADC RX

ADC RX

ADC RX

DAC TX

DAC TX

DAC TX

SYNC

COPY

COPY

COPY

PDSCH + PUSCH + ACK

PDSCH + PUCCH ACK

PUSCH

4 ms
Some Considerations on Threading

• The maximum useful pipeline depth is 3 stages (3 threads).
• Dividing uplink and downlink in two threads is also inefficient because uplink thread has to wait for downlink thread (i.e. there is no parallelization gain!)
• If more cores are available, we may divide each DSP thread and process multiple streams or codeblocks in parallel.

• Breakdown of the 4 ms deadline:
  – 1.0 ms for RX buffering
  – 0.5 ms for USRP -> Host transport
  – 2.0 ms left for processing
  – 0.5 ms for Host -> USRP transport
Processing Requirements

DSP optimizations

2 essential tools: SIMD and Precompute

• SIMD intrinsics wherever possible
  – e.g. integer arithmetic DSP, turbo decoder, soft demodulator, etc.
  – At compile time, choose generic/SSE4/AVX

• Initially target GPPs, so assume memory is almost free
• Use LUTs or pregenerate signals extensively:
  – Scrambling sequences for each subframe
  – PUCCH signals for each subframe
  – DL/UL reference signals for each subframe
  – CRC
  – Rate matching interleaver
  – ...

SRS
SOFTWARE RADIO SYSTEMS
DSP optimizations

• We use libfftw, which is fast enough even with non power of 2 sizes. This allows ¾ sampling rates:
  – 10 MHz BW: FFT 768 samples, 11.52 Msamples/s
  – 20 MHz BW: FFT 1536 samples, 23.04 Msamples/s

• This constrains us to use 32-bit complex float for transport and FFT processing.

• Yet to find a good open source integer FFT library...
The Turbo Decoder is the most demanding component:

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Percentage of CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>75 Mbps</td>
</tr>
<tr>
<td>Turbo decoder (1 iteration)</td>
<td>78.14 %</td>
</tr>
<tr>
<td>OFDM receive processing</td>
<td>6.08 %</td>
</tr>
<tr>
<td>Resource Element de-mapping</td>
<td>4.92 %</td>
</tr>
<tr>
<td>Rate recovery</td>
<td>4.49 %</td>
</tr>
<tr>
<td>CRC checksum</td>
<td>2.92 %</td>
</tr>
<tr>
<td>Soft demodulation</td>
<td>1.76 %</td>
</tr>
<tr>
<td>Equalization</td>
<td>0.16 %</td>
</tr>
<tr>
<td>Others</td>
<td>1.53 %</td>
</tr>
<tr>
<td><strong>Total Execution Time</strong></td>
<td><strong>954 (\mu s)</strong></td>
</tr>
</tbody>
</table>
• We used 16-bit integer arithmetic and 128-bit SSE4 instructions to compute all the trellis (8 states) in parallel
• With 8-bit arithmetic could do 2 codeblocks in parallel
• With AVX2 could do 2 or 4 codeblocks in parallel
Execution time of PDSCH only @ 20 MHz
srsLTE

srsLTE is a free and open-source LTE software suite developed by SRS (www.softwareradiosystems.com).

It includes:

- srsUE - a complete SDR LTE UE application featuring all layers from PHY to IP
- srsENB - a complete SDR LTE eNodeB application
- a highly modular set of common libraries for PHY, MAC, RLC, PDCP, RRC, NAS, S1AP and GW layers.

srsLTE is released under the AGPLv3 license and uses software from the OpenLTE project (http://sourceforge.net/projects/openlte) for some security functions and for RRC/NAS message parsing.