GNU Radio and OSSIE

Gesly A George
Summer 2005
GNU Radio

- Framework for building SDRs
- Consists of the core signal processing library and the radio hardware (USRP)
- Library provides signal processing and filters
Universal Software Radio Peripheral (USRP)

- ADCs, RF front end and FPGA
- FPGA – for high bandwidth computations (interpolations, digital up and down conversions)
- Rest of the signal processing done on the PC
- Interfaces to the PC using USB2
GNU Radio

- Radio represented as a flow graph (DAG)
- Vertex: Signal processing block
- Scheduler executes the flow graph – data driven
- Provides message queues (MQs) for asynchronous communication (can be used for control messages)
GNU Radio

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNU Radio Core</td>
</tr>
<tr>
<td>GNU Radio/USRP Glue</td>
</tr>
<tr>
<td>USRP</td>
</tr>
</tbody>
</table>

(a) The typical GNU Radio application stack
Issues with GNU Radio

- Designed for unidirectional work
- Lack of timing facilities
- Needs continuous data flow to work properly
- Bandwidth issues with USRP (32 MB/s split for full duplex operation)
- Data buffering issues
Software Communications Architecture (SCA)

- SCA specifies a communication architecture to ensure portability and configurability of software and hardware (developed by JTRS – Joint Tactical Radio System)
- Interface specifications, APIs etc.
- Core Framework (CF) – essential application interfaces and services
- CORBA middleware – message passing
- Specifies APIs for Radio building blocks – PHY, MAC, LLC, I/O
Open Source SCA Implementation:: Embedded (OSSIE)

- A C++ implementation of the SCA framework
- Implementation of the Core Framework of SCA is available currently
- Building blocks (PHY, MAC) have to be built from this framework
- Sample application provided does basic waveform modulation and demodulation
OSSIE

- Real time processing possible (at least theoretically)
- Virginia Tech. has interfaced OSSIE with USRP – very basic test