Sora: High Performance Software Radio using General Purpose Multi-Core Processors

Kun Tan[†] Jiansong Zhang[†] Ji Fang[‡] He Liu[§] Yusheng Ye[§]
Shen Wang[§] Yongguang Zhang[†] Haitao Wu[†] Wei Wang[†]
Geoffrey M. Voelker[◊]

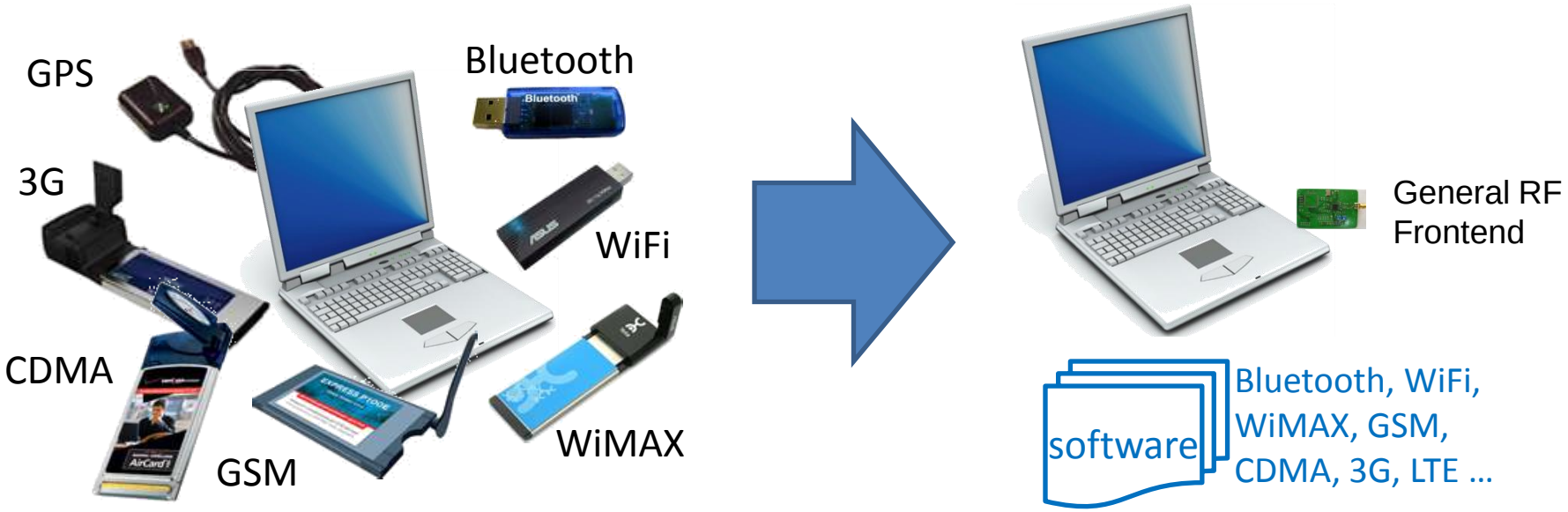
[†] Microsoft Research Asia

[‡] Tsinghua University, Beijing, China

[§] Beijing Jiaotong University, Beijing, China

[◊] UCSD, La Jolla, USA

Software Radio

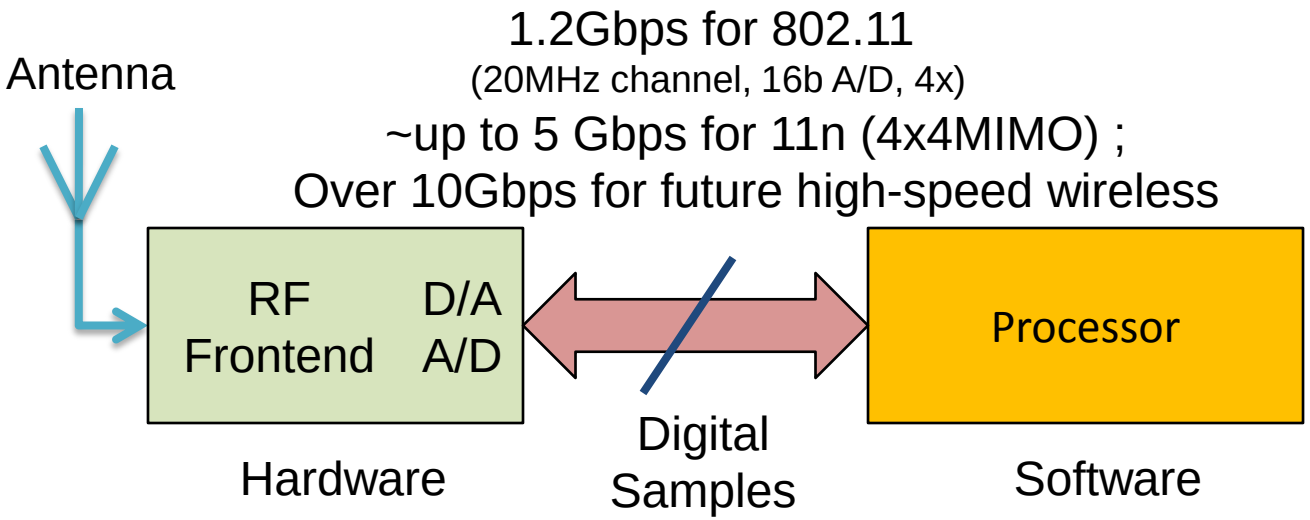


Benefits

- ✓Promise of universal connectivity and cost saving
- ✓Programmability => faster development cycle, faster to market
- ✓Open platform for wireless research

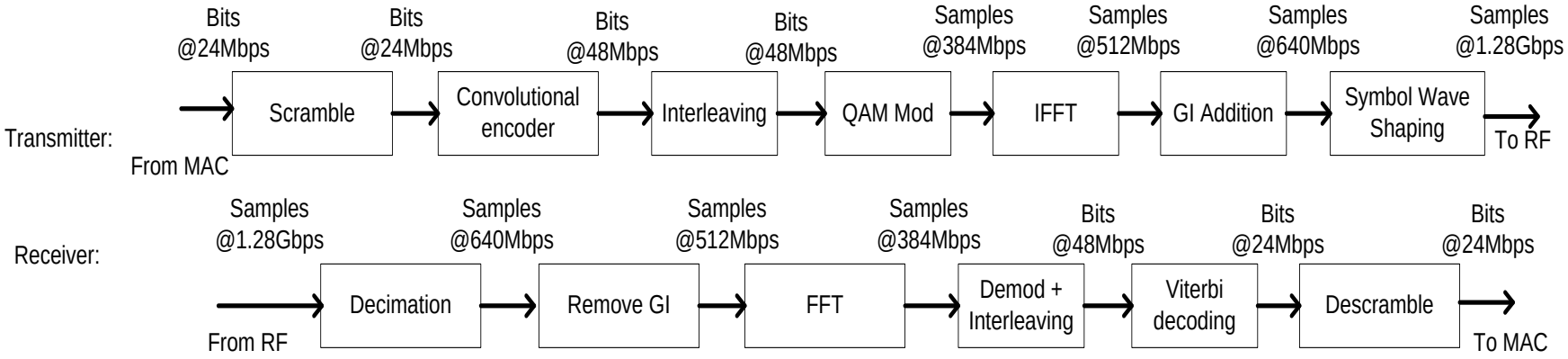
Fundamental Challenges

- Large volume of high-fidelity digital signals
  - Require a high-speed system I/O



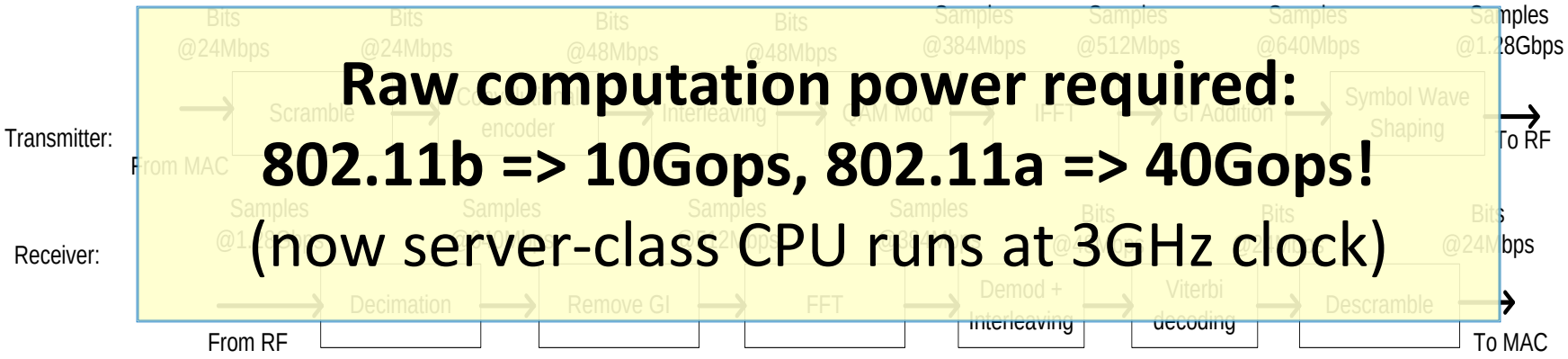
Fundamental Challenges

- Large volume of high-fidelity digital signals
  - Require a high-speed system I/O
- Computation-intensive signal processing



Fundamental Challenges

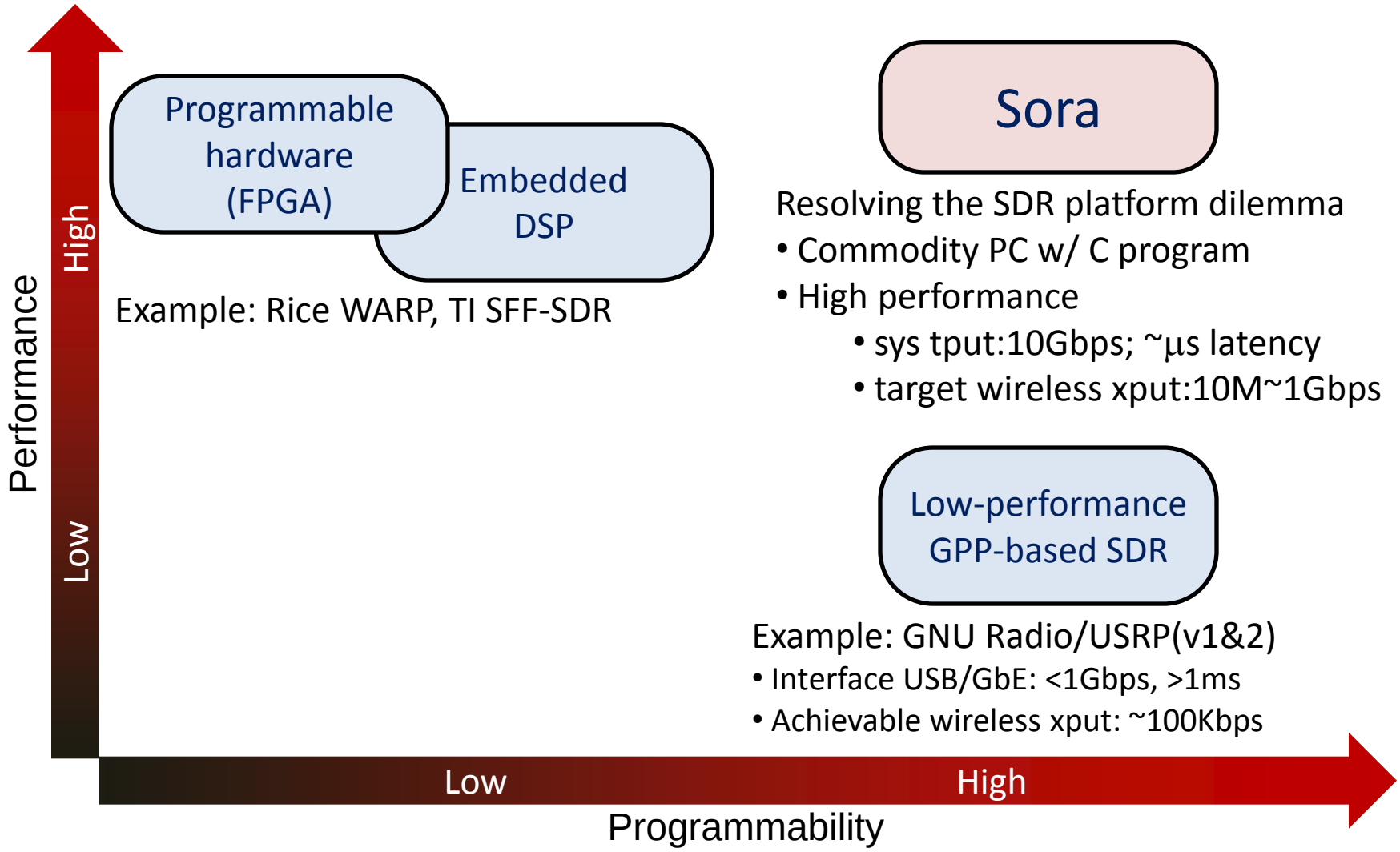
- Large volume of high-fidelity digital signals
  - Require a high-speed system I/O
- Computation-intensive signal processing



Fundamental Challenges

- Large volume of high-fidelity digital signals
  - Require a high-speed system I/O
- Computation-intensive signal processing
- Hard deadline and accurate timing control
  - 802.11 MAC requires response within a few μs
  - Event trigger timing accuracy at μs level

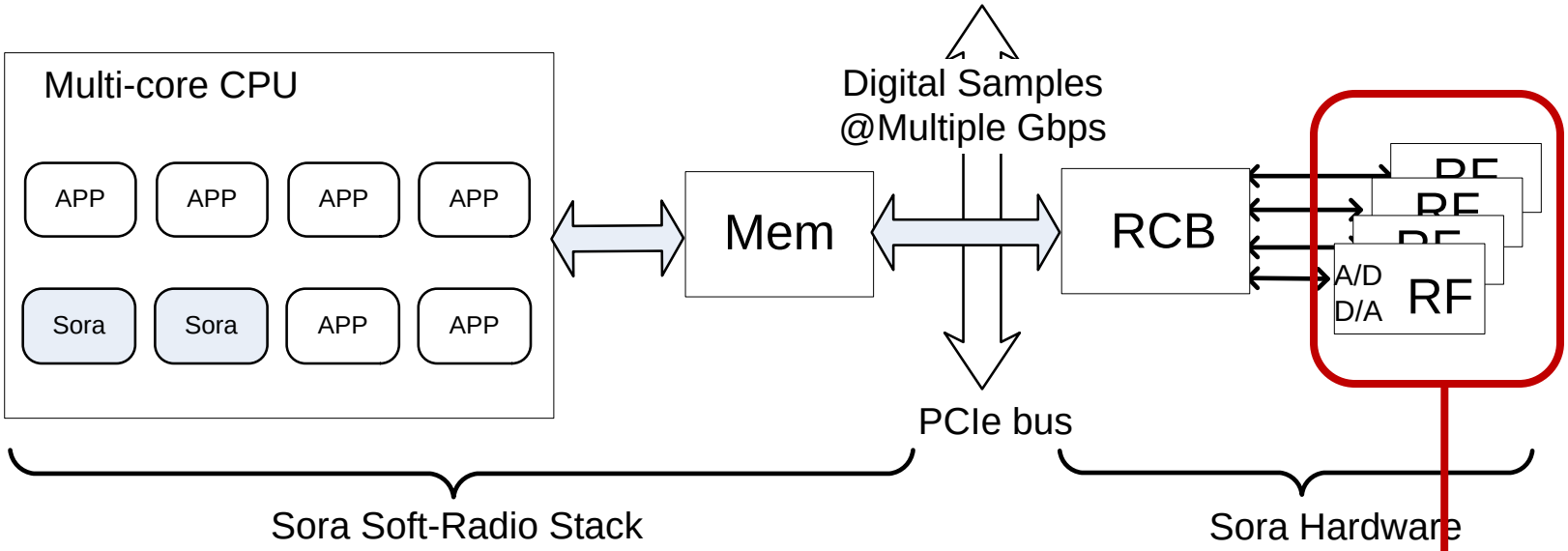
Approaches


Performance
High
Low
Programmable hardware (FPGA)
Embedded DSP
Example: Rice WARP, TI SFF-SDR
Sora
Resolving the SDR platform dilemma
• Commodity PC w/ C program
• High performance
• sys tput:10Gbps; ~µs latency
• target wireless xput:10M~1Gbps
Low-performance GPP-based SDR
Example: GNU Radio/USRP(v1&2)
• Interface USB/GbE: <1Gbps, >1ms
• Achievable wireless xput: ~100Kbps
Low
High
Programmability

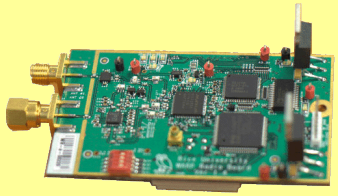
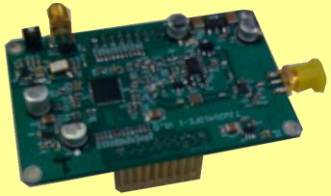
Sora Approach

- New PCIe-based Interface card => high system throughput
- New optimizations to implement PHY algorithms and streamline processing on multi-core CPU=> efficient PHY processing
- Core dedication => real-time support

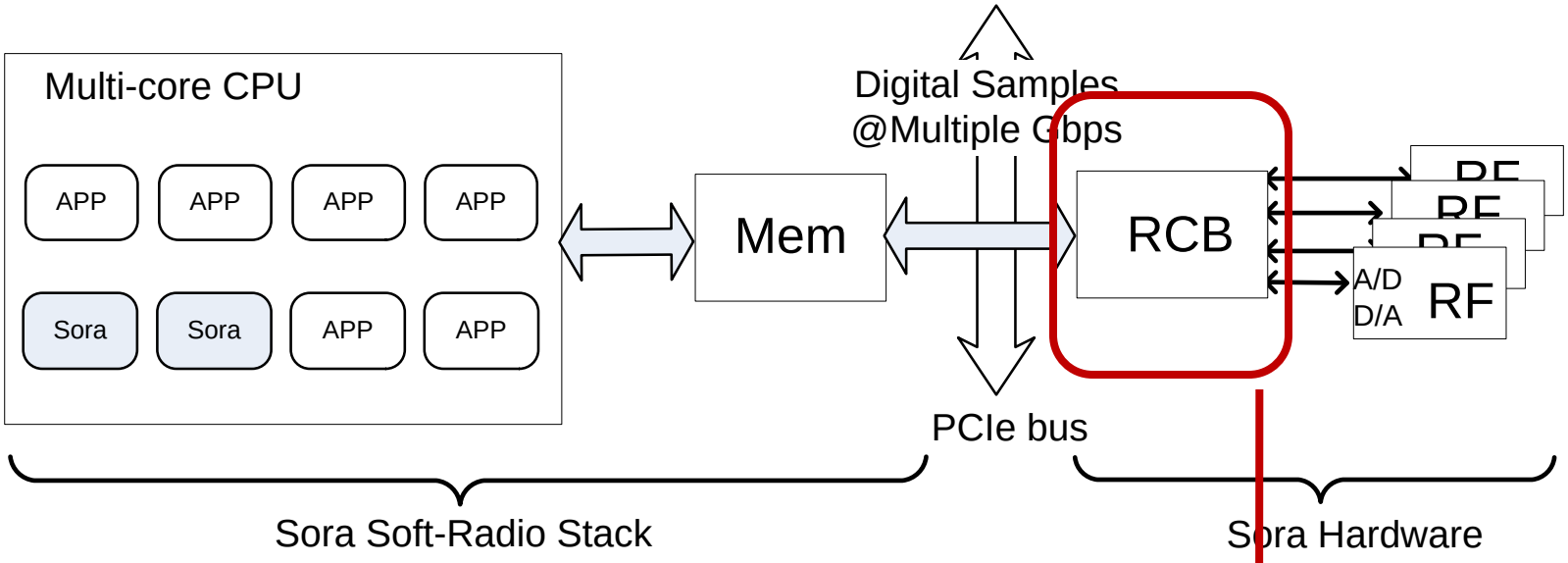
Sora Architecture



General radio front-end: 700M/1.8G/2.4G/5GHz

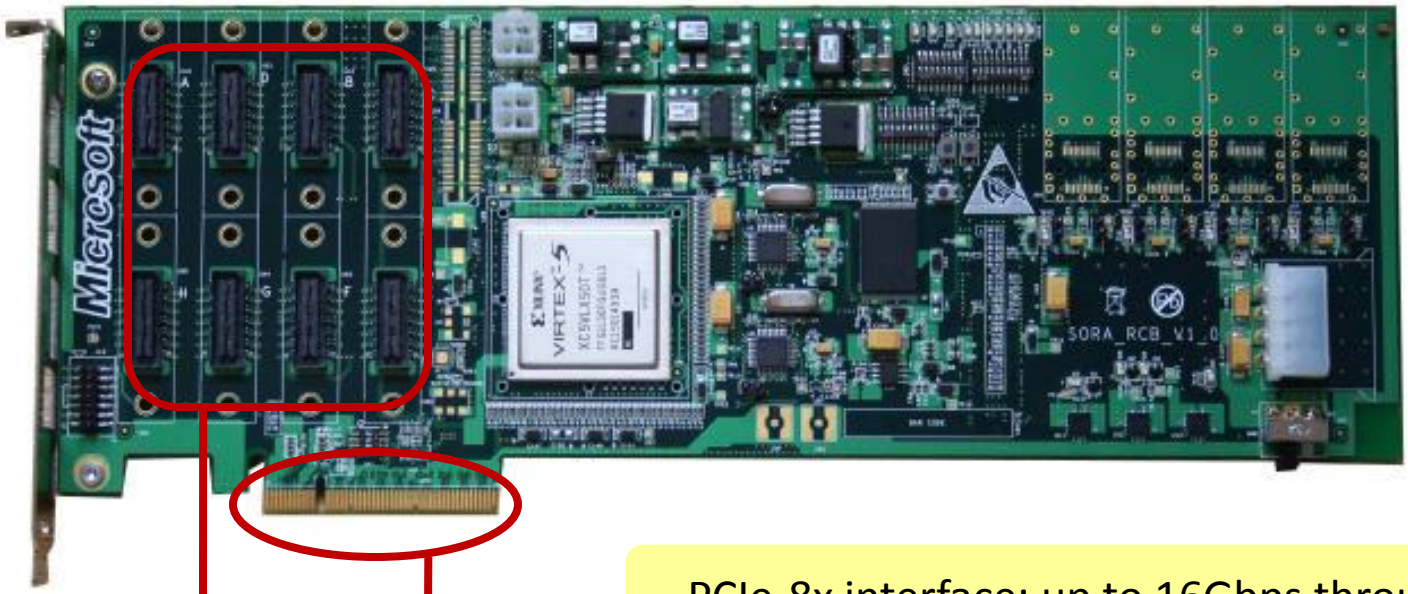
Radio Control Board



PCIe-based High-speed Interface card

- ✓ PCIe is commodity in most modern PCs
- ✓ High throughput: 16Gbps at PCIe-8x
- ✓ Low latency: ~ 1 μs
- ✓ Separated with other I/O devices

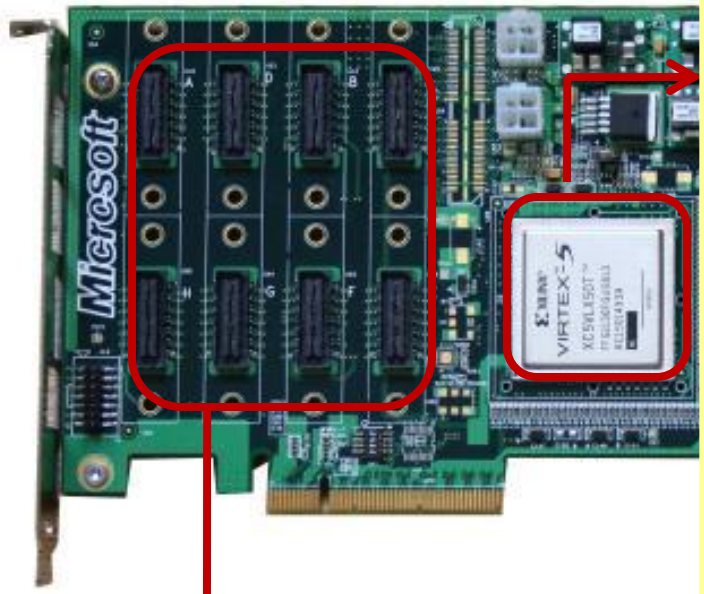
RCB Details

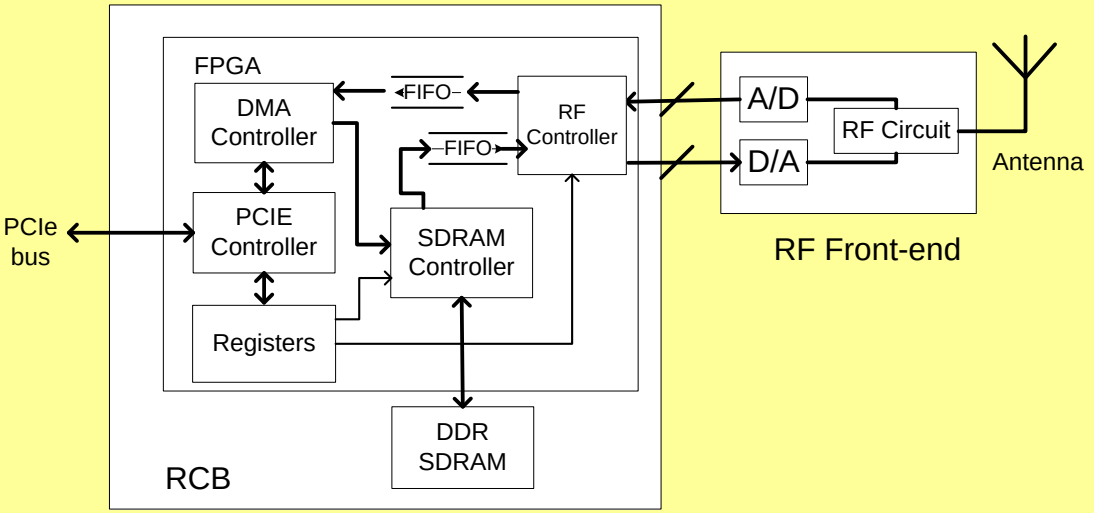


PCIe-8x interface: up to 16Gbps throughput

Versatile RF interface: up to 8 channels (8x8 MIMO)

RCB Details

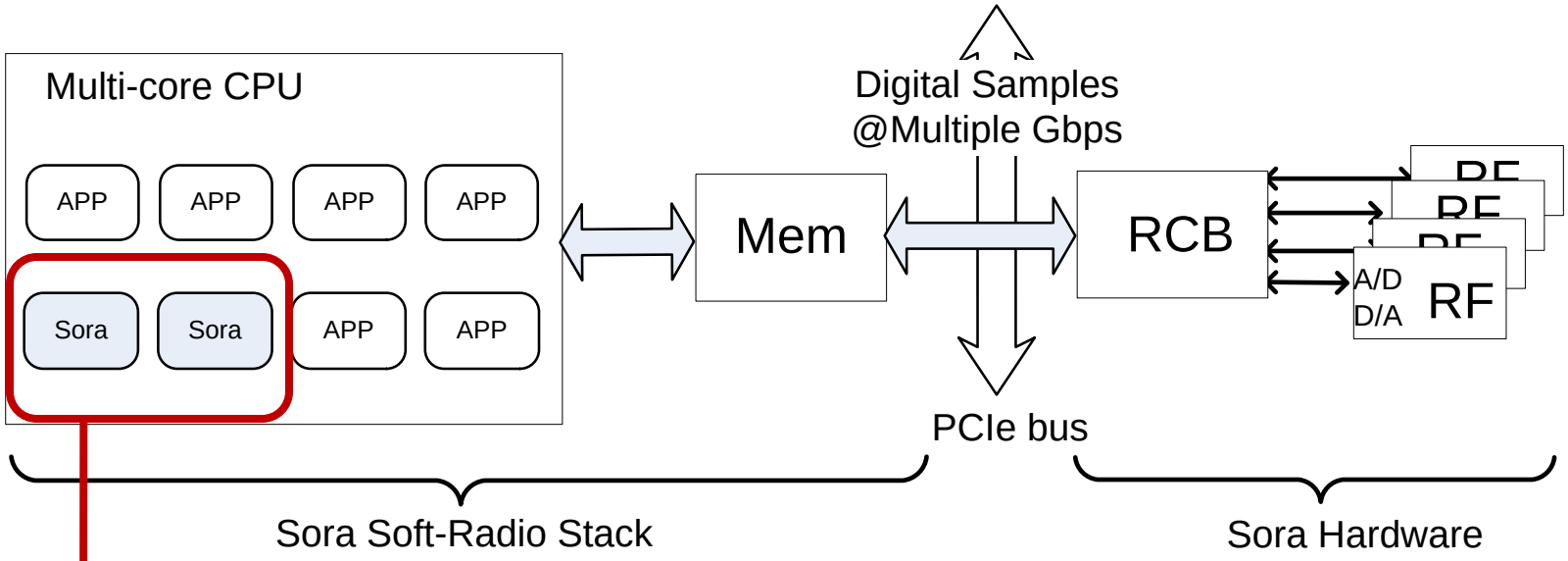




- ✓ Buffered data path: bridging the synchronous ops at RF and asynchronous processing at CPU (12.3Gbps measured)
- ✓ Low latency control path for software (0.36 μs measured)

Versatile RF int

Sora Software

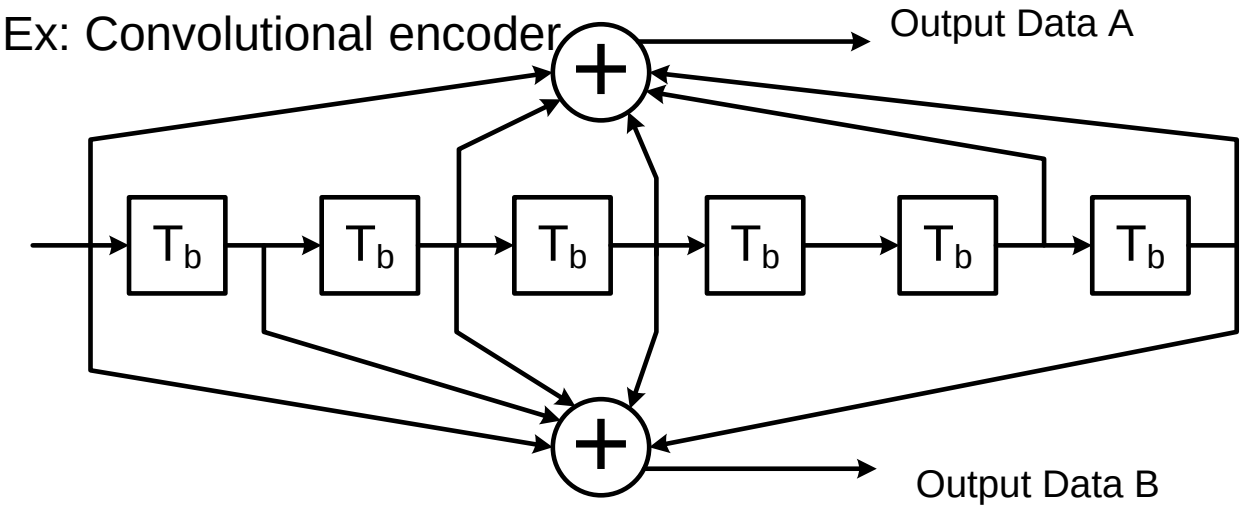


High-performance SDR processing w/ key software techniques

- ✓ Efficient PHY implementation using SIMD and LUTs
- ✓ Speed up PHY using multi-core streamline processing
- ✓ Core dedication for real-time support

Efficient PHY Implementation

- Exploit large high-speed cache memory
  - Extensive use of lookup tables (LUT): trade memory for calculation; still well fit into L2 cache
  - Applicable for more than half of the common algorithms; speedup ranges from 1.5x to 22x



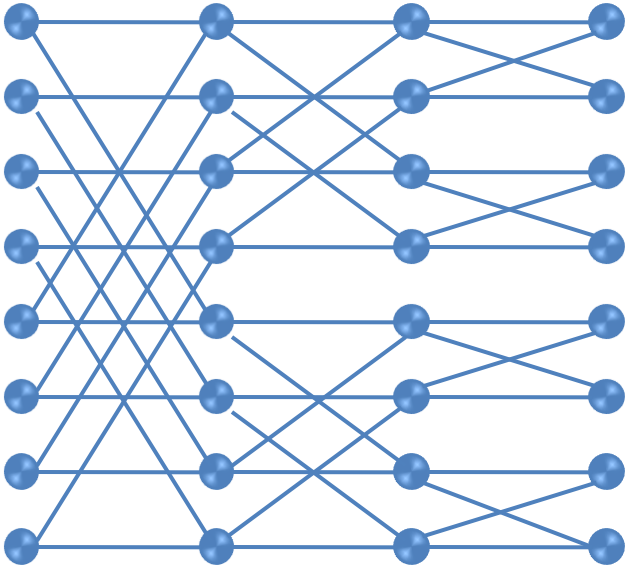
Direct impl. 8 ops per bit

LUT impl. 2 Look-up op for 8 bits! (size 32KB)

Efficient PHY Implementation

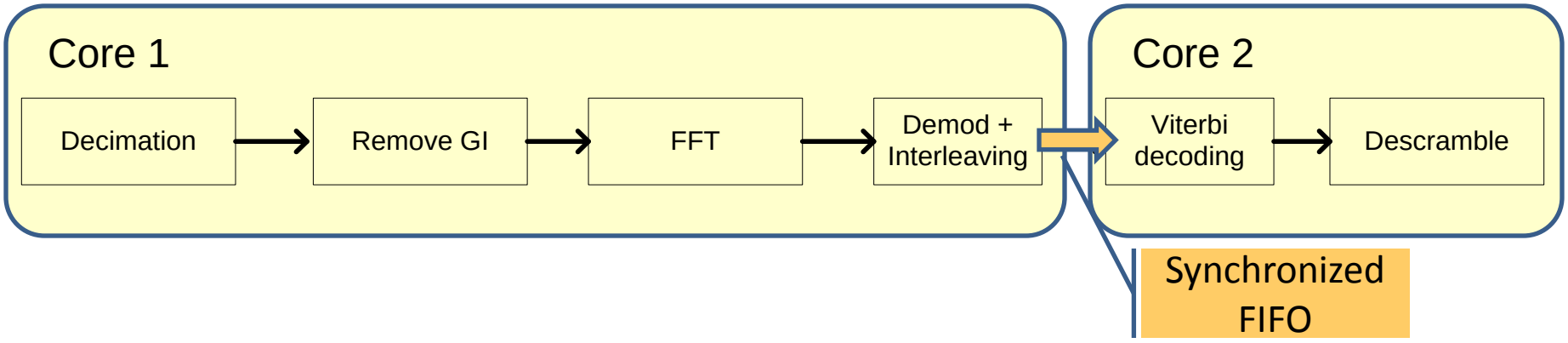
- Exploit data parallelism in PHY
  - Utilize wide-vector SIMD extension in CPU
  - Applicable to many PHY algorithms with significant speedups (1.6x ~ 50x)

Ex. (I)FFT



Speed up PHY using multi-core streamline processing

- Efficiently partition and schedule the PHY processing across cores
  - Interconnecting sub-pipeline with light-weight, synchronized FIFOs
  - Static scheduling of processing modules in PHY pipeline



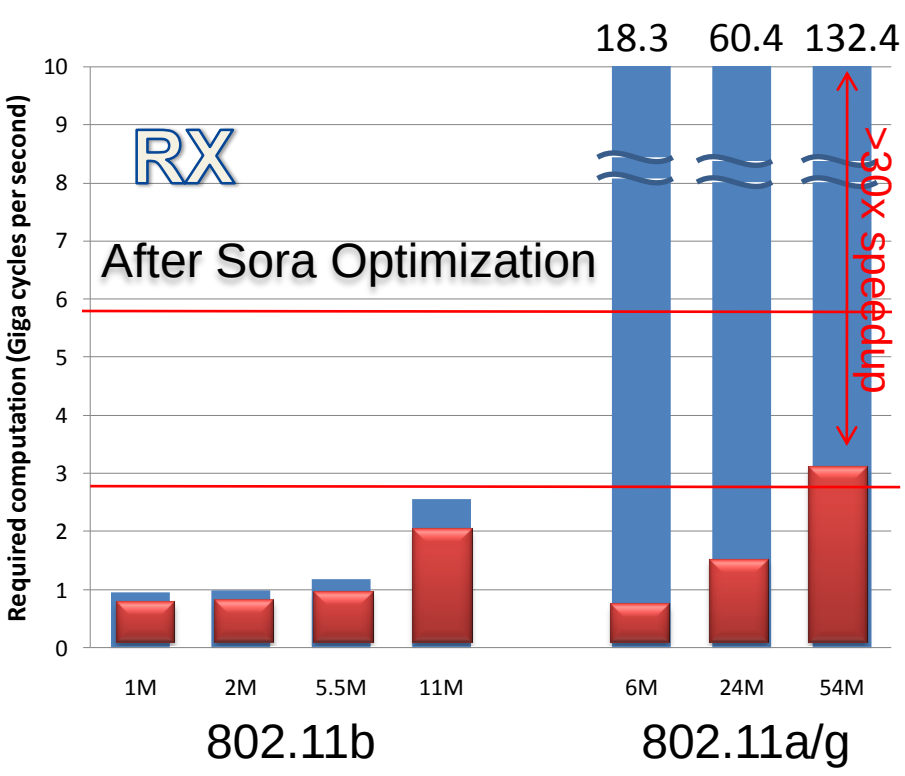
Core Dedication for Real-time Support

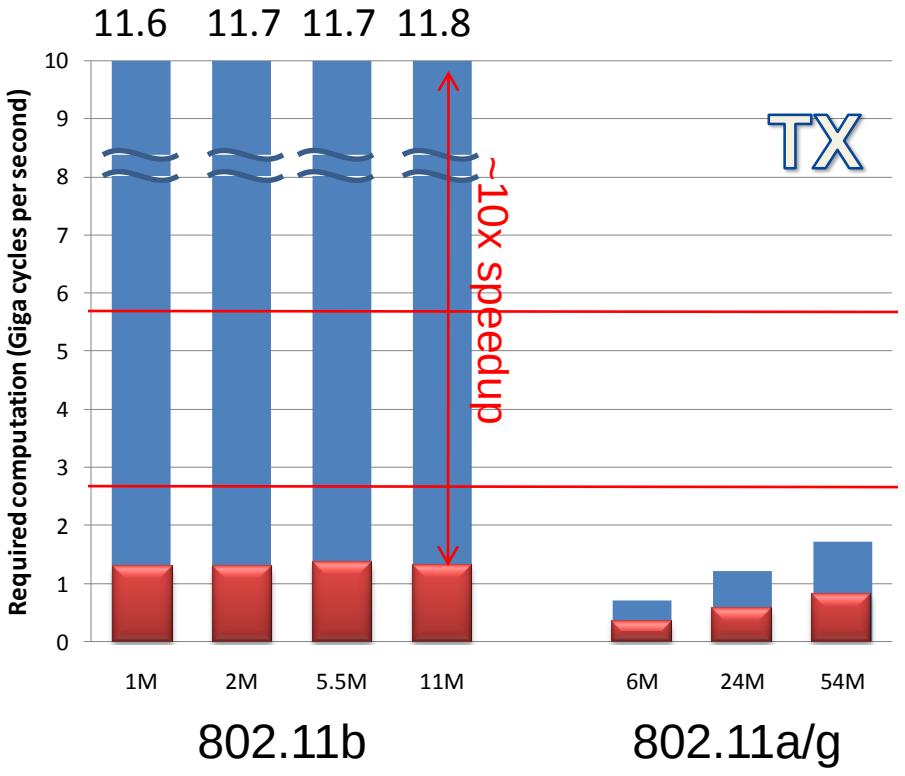
- Exclusively allocate enough cores for SDR processing in multi-core systems
  - Guarantee the CPU, cache and memory bandwidth resources for predictable performance
  - Achieve μs-level timing control
  - Simple abstraction, and easier to implement in standard OSes than RT-scheduler
    - Implemented in WinXP without modifications to Kernel

Implementation

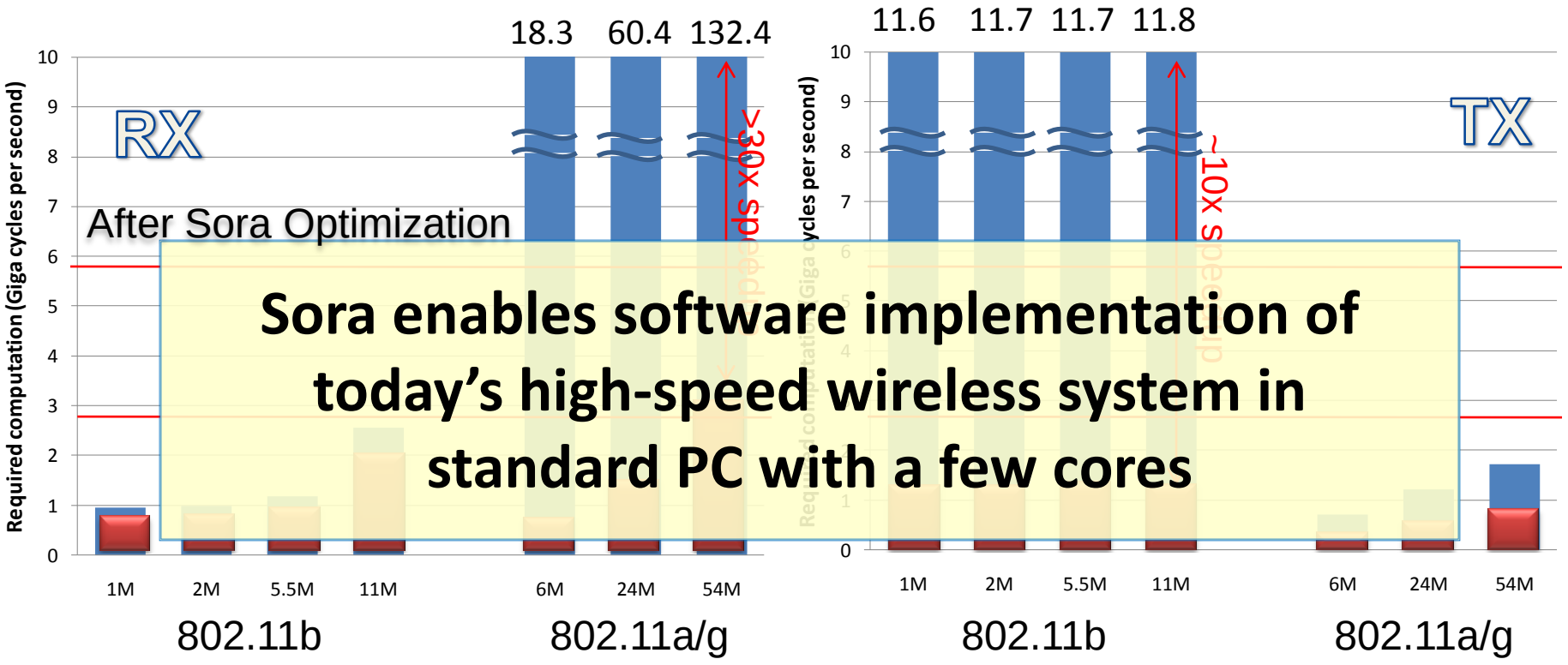
- Sora software platform on Win XP
  - 14K lines of C code, including PCIe driver framework, memory management, FIFO management, etc
- SoftWiFi: full implementation of IEEE 802.11a/b/g PHY and DCF MAC
  - 9K lines of C code; 4 man-month for dev & test
  - DSSS 1, 2, 5.5, 11Mbps for 11b; OFDM 6, 9, 12, 18, 24, 36, 48, 54Mbps for 11a/g

Results: PHY Processing





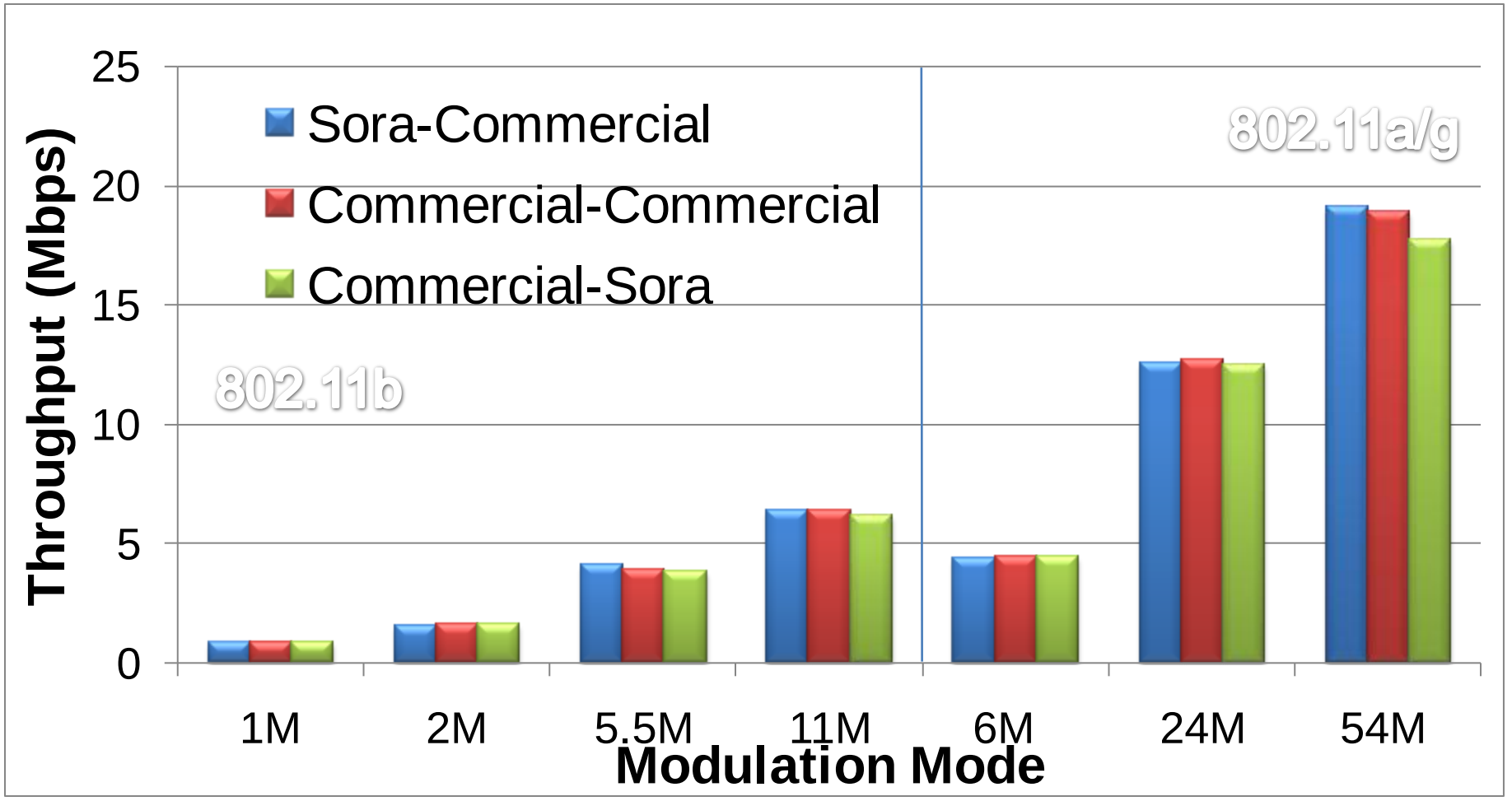
Results: PHY Processing



Sora enables software implementation of today's high-speed wireless system in standard PC with a few cores

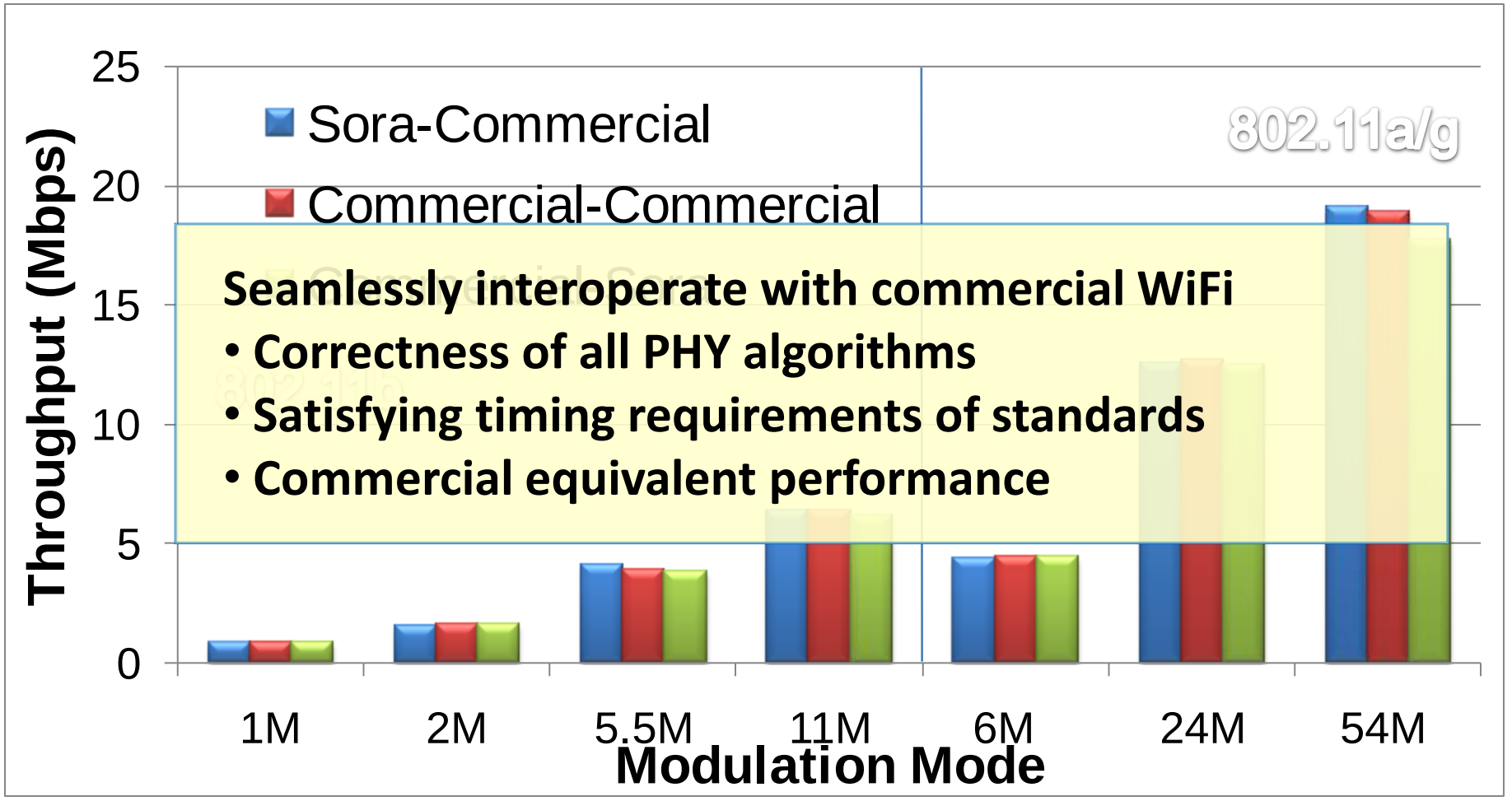
Results: End-to-end Throughput

Communicating with commercial 802.11a/b/g card



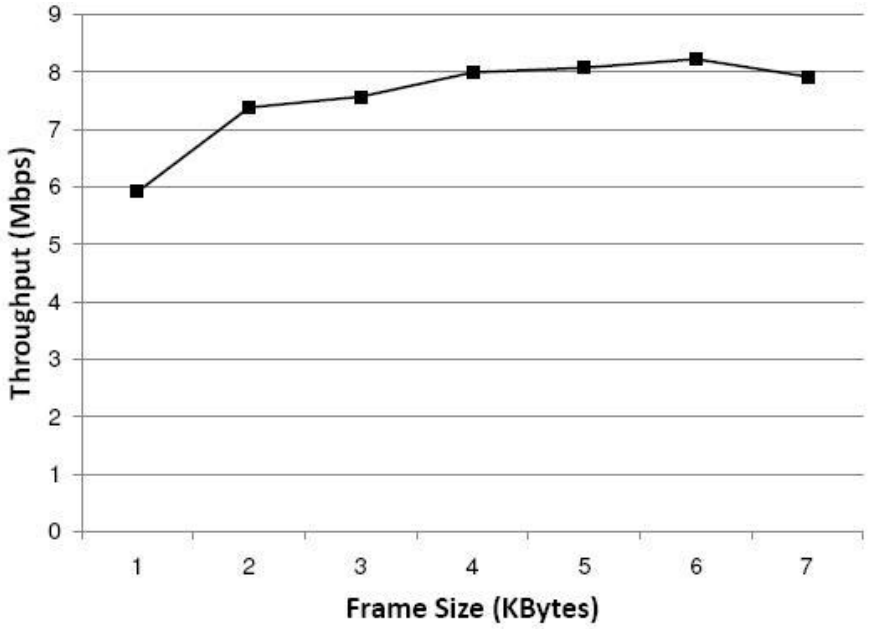
Results: End-to-end Throughput

Communicating with commercial 802.11a/b/g card



Extensions

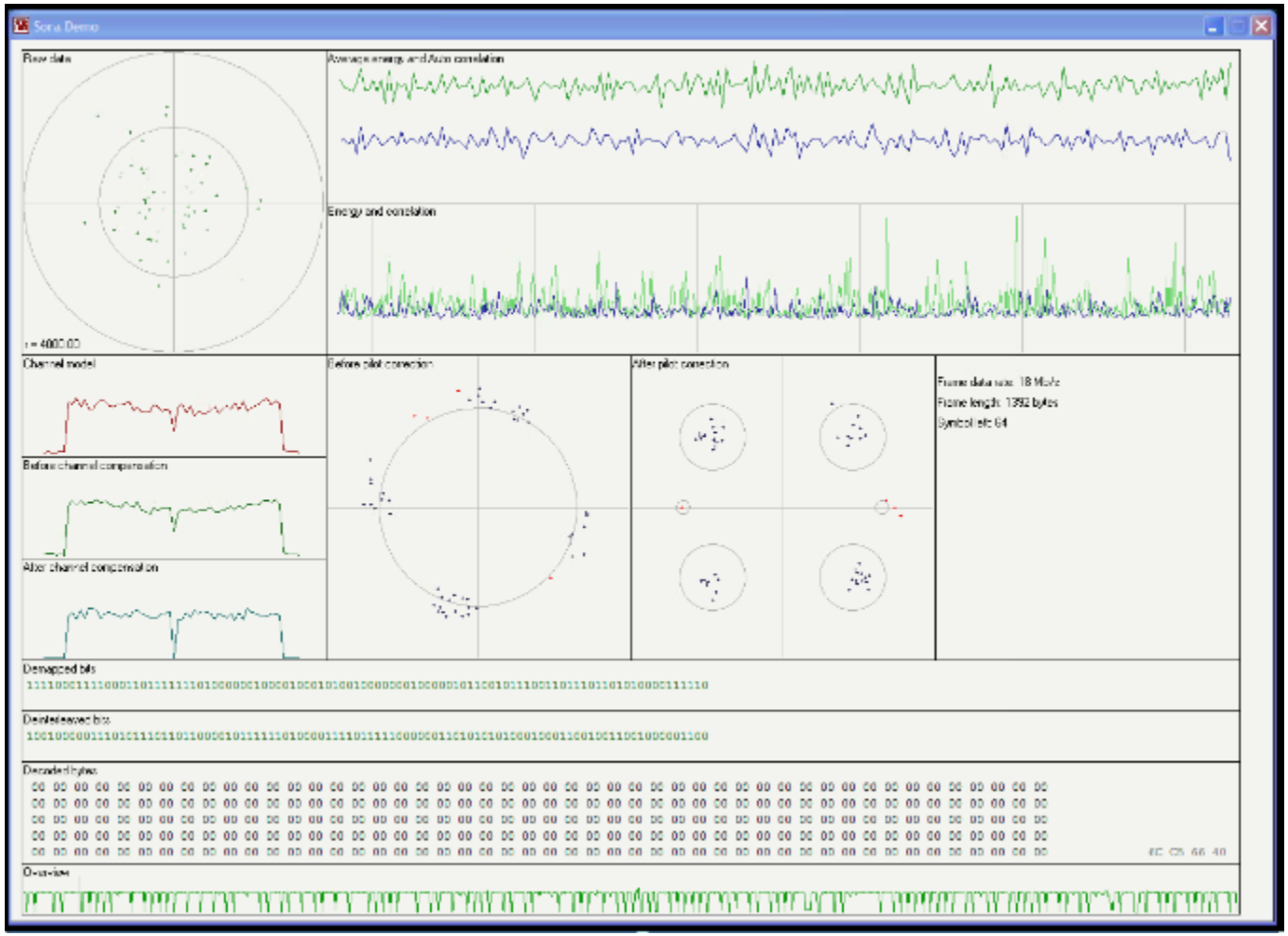
Jumbo frames in 802.11



TDMA MAC

	10ms	50ms	100ms
$\epsilon/\sigma(\mu s)$	0.85/0.5	0.96/0.54	0.98/0.46
Outlier	0.5%	0.4%	0.4%

Extensions: New Applications

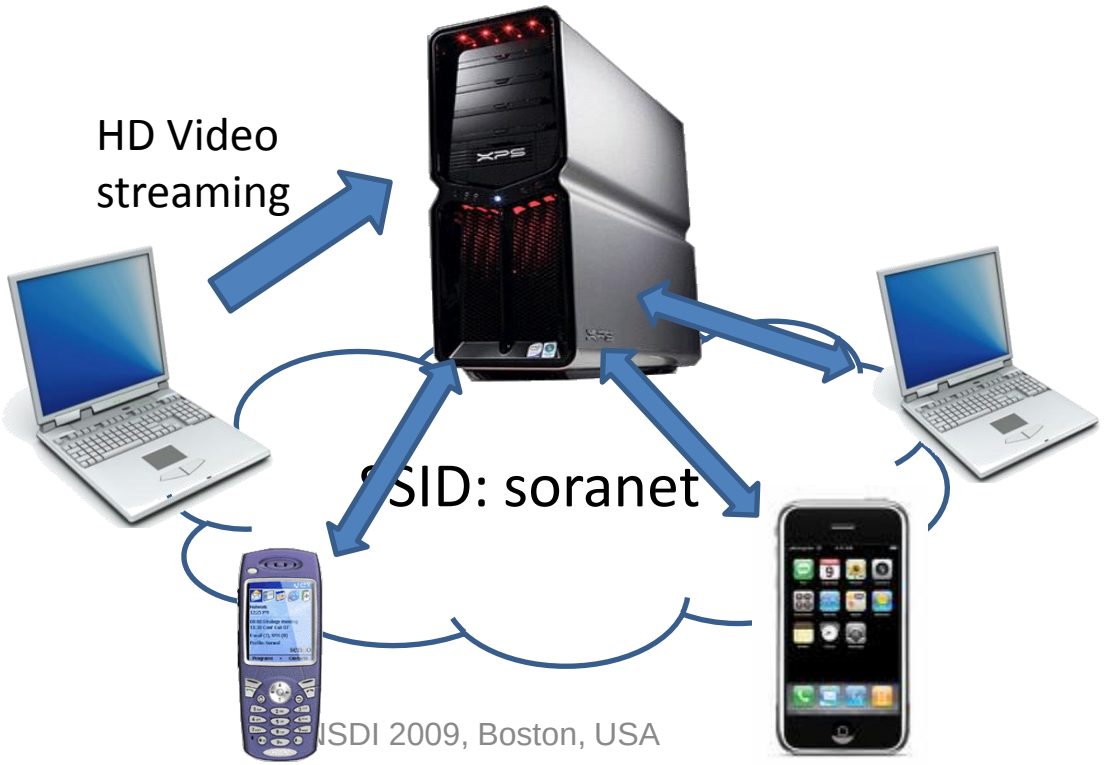


Conclusion

- Sora is a fully programmable software radio platform on commodity PC architecture
  - Easy C programming on multi-core CPU
  - High performance: high processing speed, low latency, and performance guarantee
    - *Confirmed by SoftWiFi, the first fully interoperable IEEE 802.11 (PHY and MAC) on general purpose processors*
- Plan to release Sora SDK to research community
  - H/W: RCB + 2.4G RF front-end set (~$2K USD)

DEMO

- Sora demo in the demo session this evening
- You can interact with Sora with your own laptop, iPhone, or other smart phones



Q&A