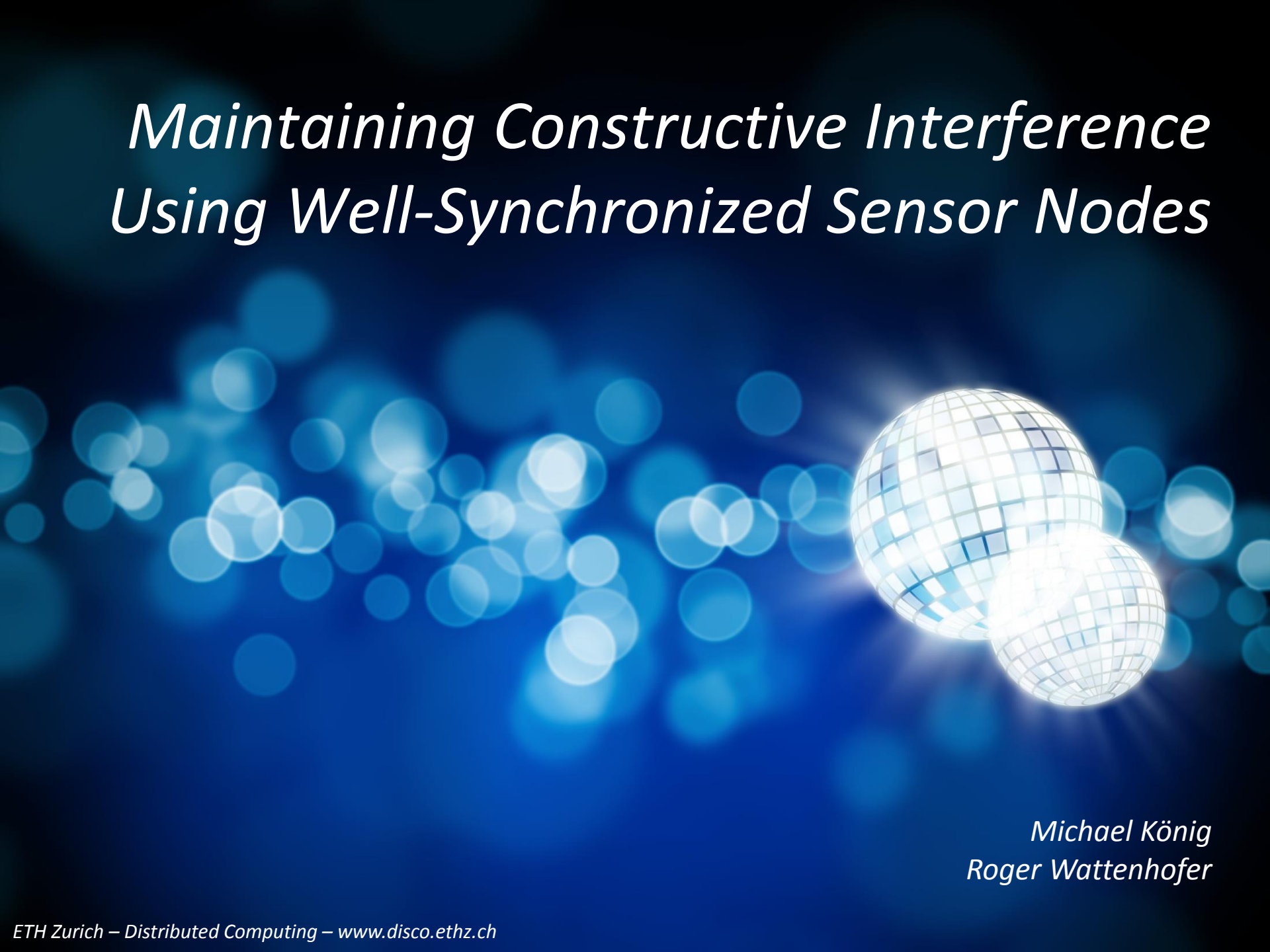
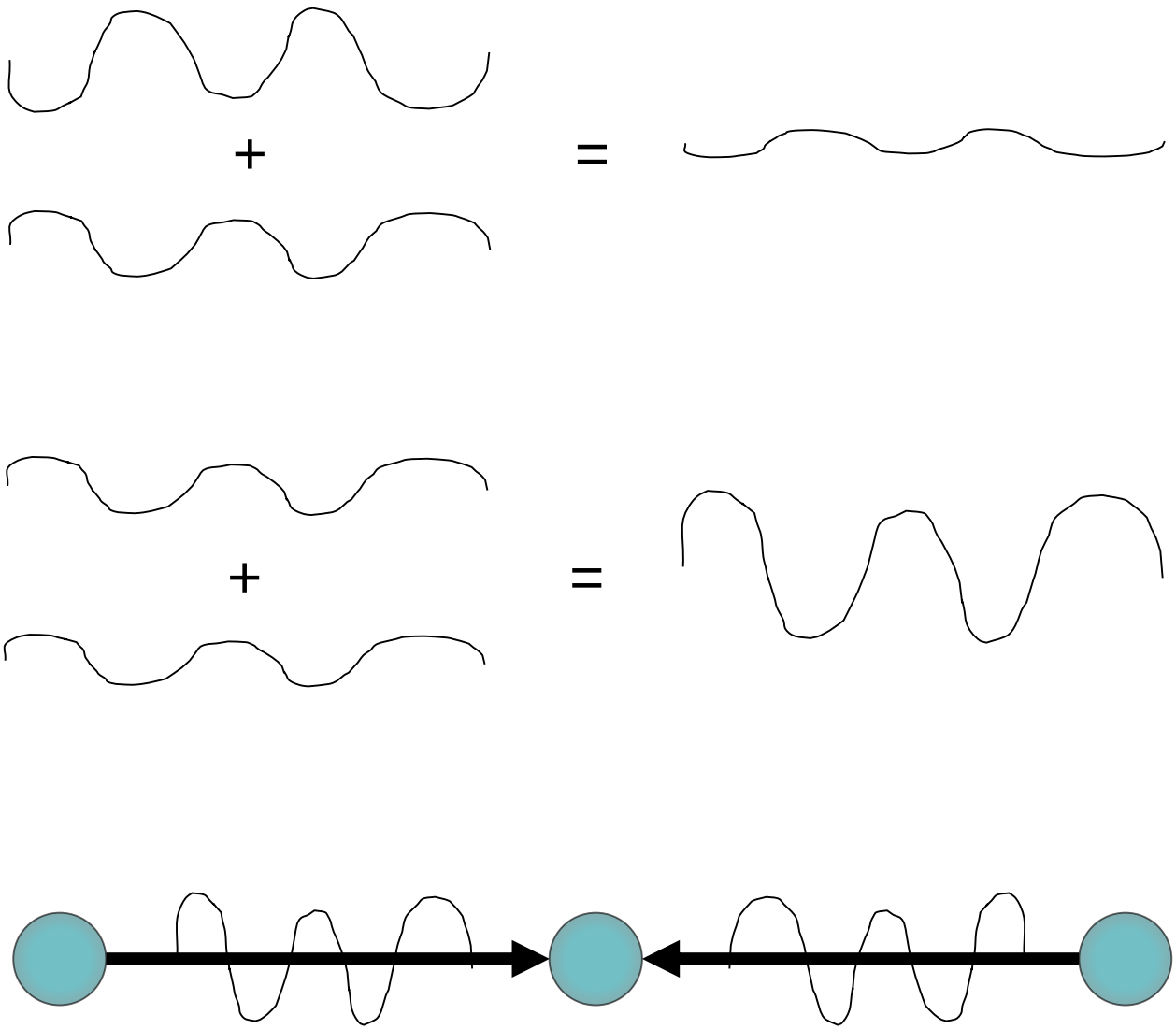
Maintaining Constructive Interference Using Well-Synchronized Sensor Nodes



Michael König
Roger Wattenhofer

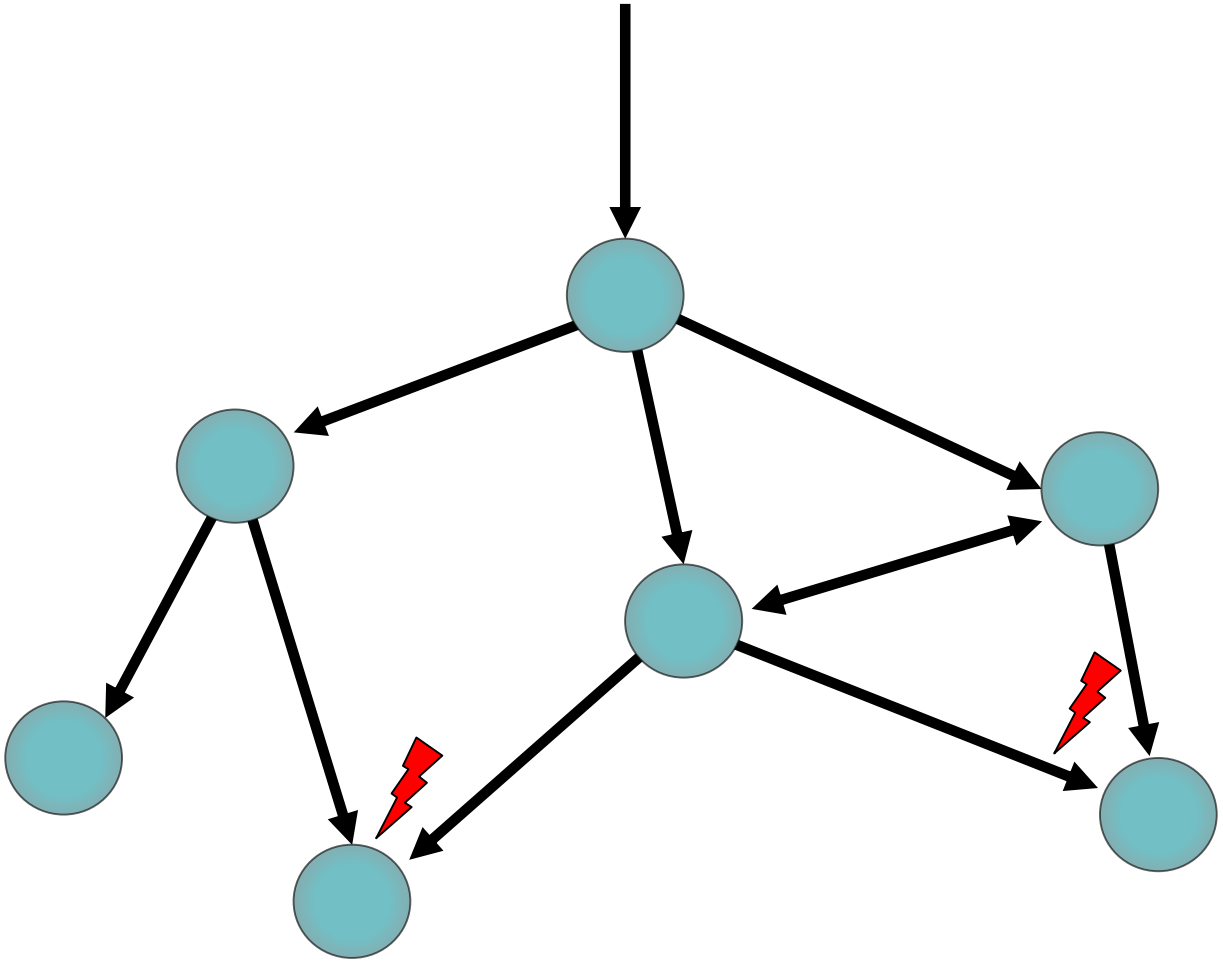
Constructive Interference (CI)



Traditional Approaches to Transmission Synchronization

- Use an external clock.
- Don’t send complicated data.
  - SlotOS [Flury et al., 2010]
  - Black Burst Synchronization [Gotzhein et al., 2011]
- Use triggering packets.
  - Glossy [Ferrari et al., 2011]

CI for Broadcasting: Glossy



Traditional Approaches to Transmission Synchronization

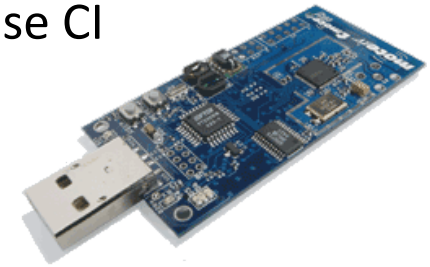
- Use an external clock.
- Don’t send complicated data.
- Use triggering packets.

Can we do it …

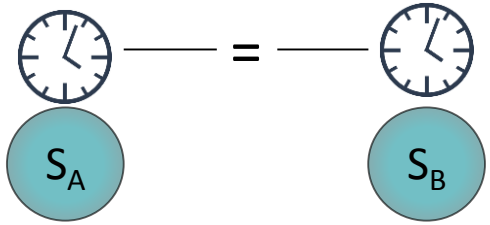
- … without an external clock,
- … with normal data packets,
- … and without triggering packets?

Our Work

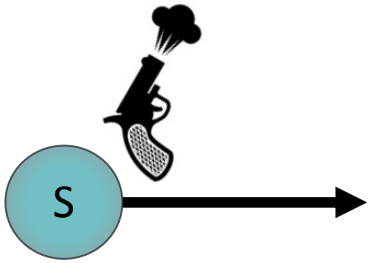
- General case CI
- TelosB



- Minimizing error sources:
  - Clock synchronization



  - Transmission timing

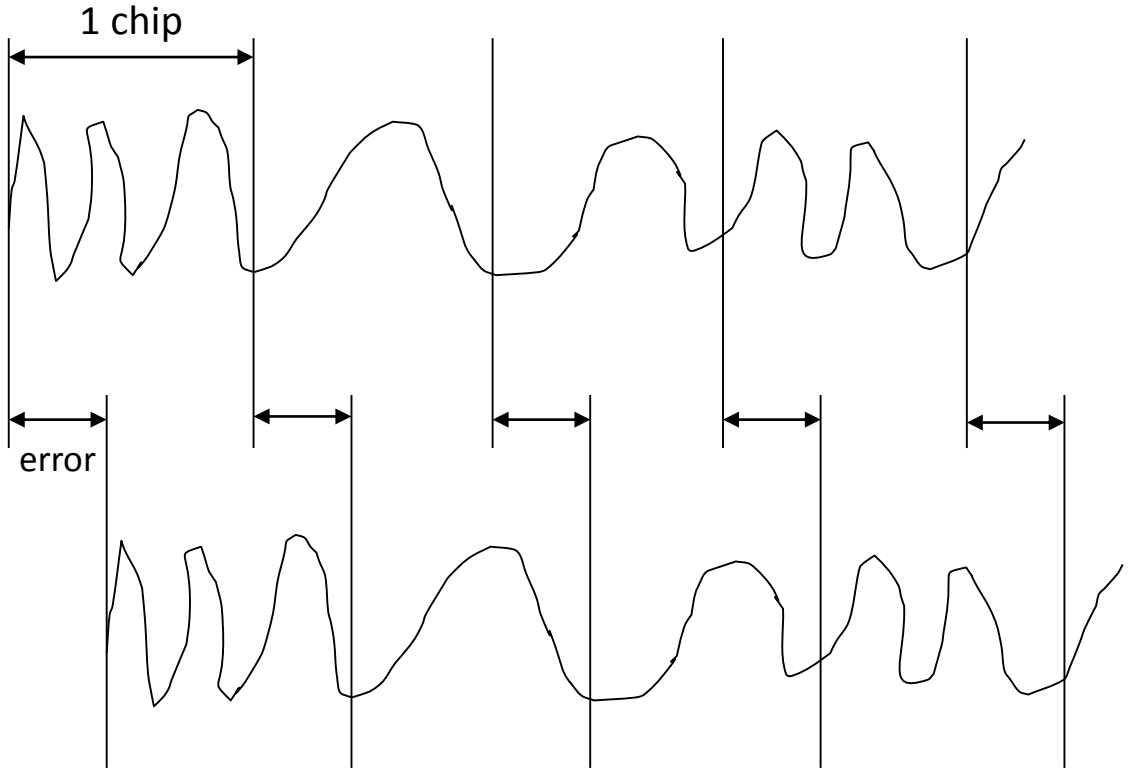


  - Travel time

CI Timing Requirements (IEEE 802.15.4)

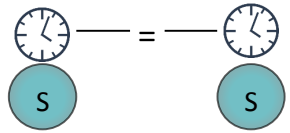
- 250,000 bit/s
- 4-bit symbols
- 32 chips/symbol, on I and Q phases in parallel

→ 1 µs per chip

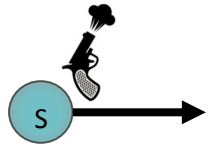


Error Sources

- Clock synchronization $e_{clock} = clock_A - clock_B$



- Transmission Timing $e_{transmit} = delay_B - delay_A$



- Travel Time $e_{travel} = traveltime_B - traveltime_A$

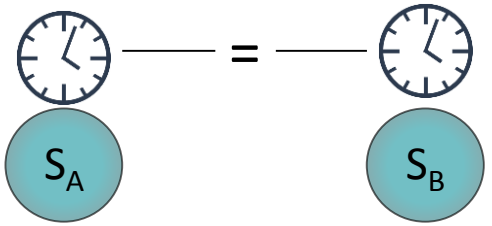
- Total Error $e_{total} = e_{clock} + e_{transmit} + e_{travel}$
- Goal $|e_{total}| << 0.5\ \mu s$

Travel Time (e_{travel})

- 30 meters = 0.1 µs travel time
- Workaround: negligible for senders of similar distance
  - $|e_{travel}| < 0.033$ µs

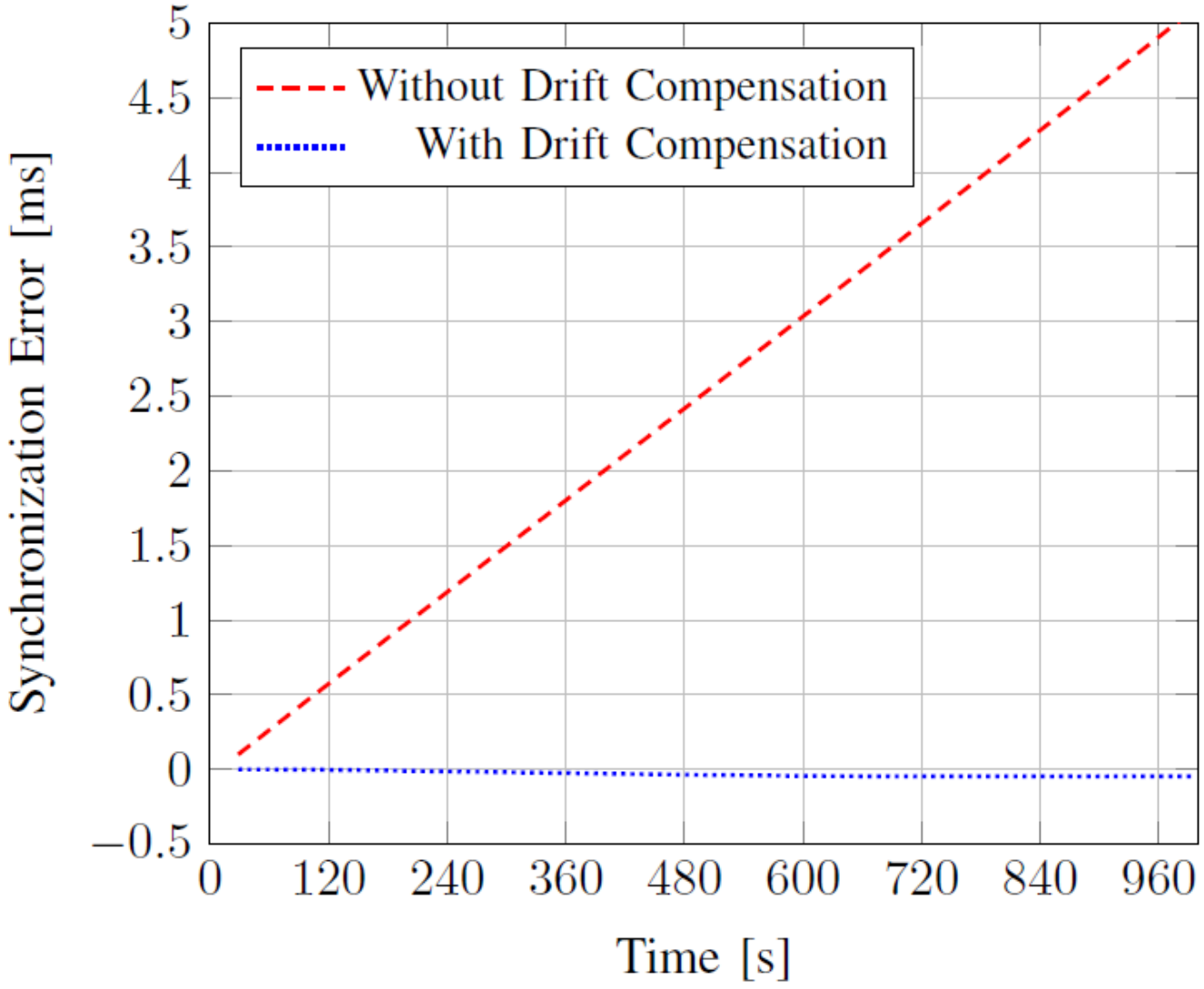
- Recent work
  - "Time-of-Flight Aware Time Synchronization" (TATS) [Lim et al., 2016]

Clock Synchronization (e_{clock})



- MAC layer timestamping
- Averaging out errors
- Drift compensation
- Combining clocks
- Extended precision timestamps

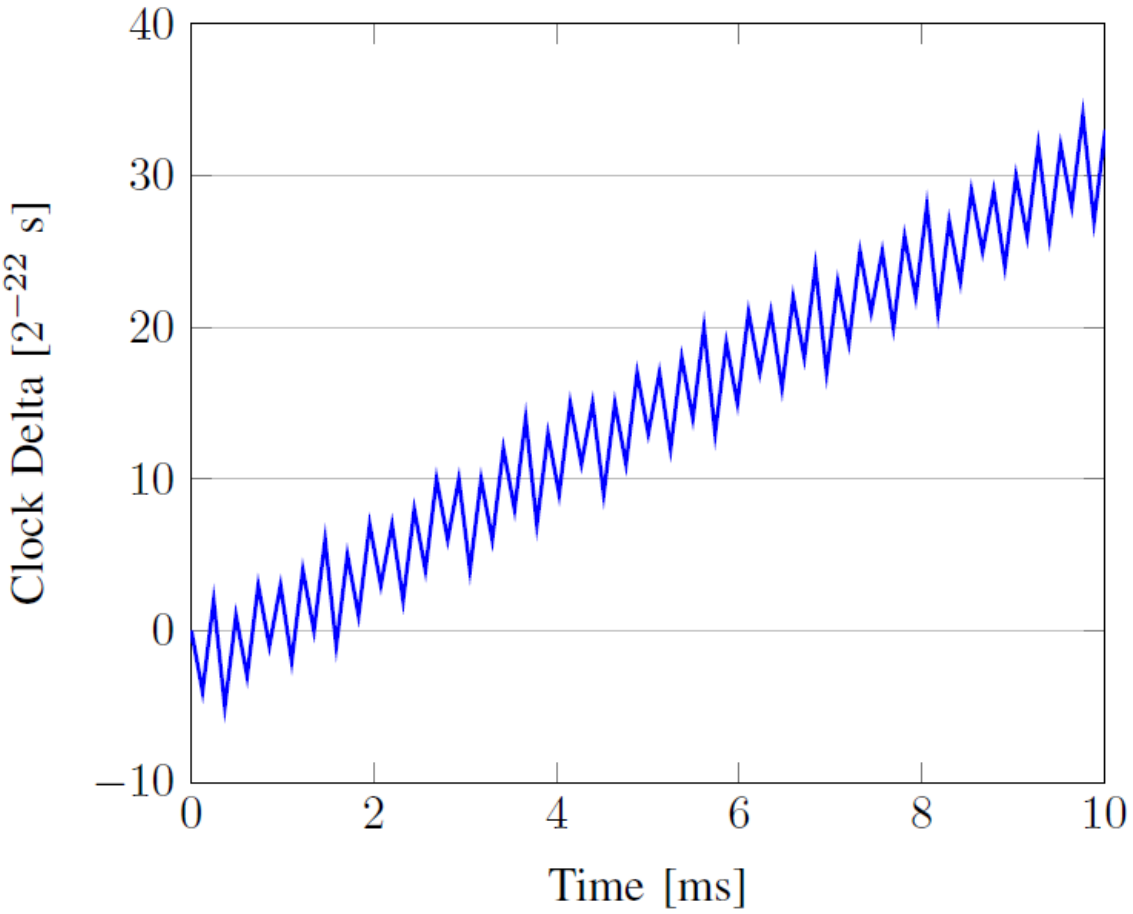
Drift Compensation



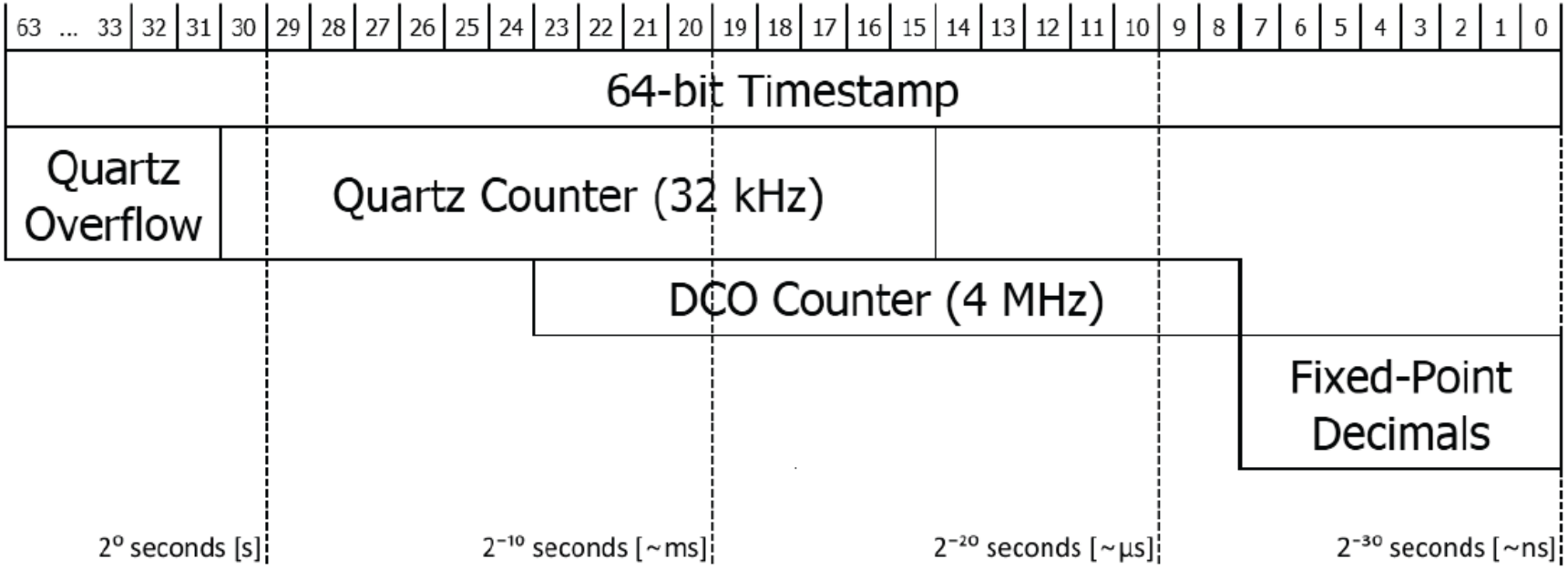
Combining Clocks

2 clocks:

- 32 kHz quartz
- 4 MHz DCO (digitally controlled oscillator)

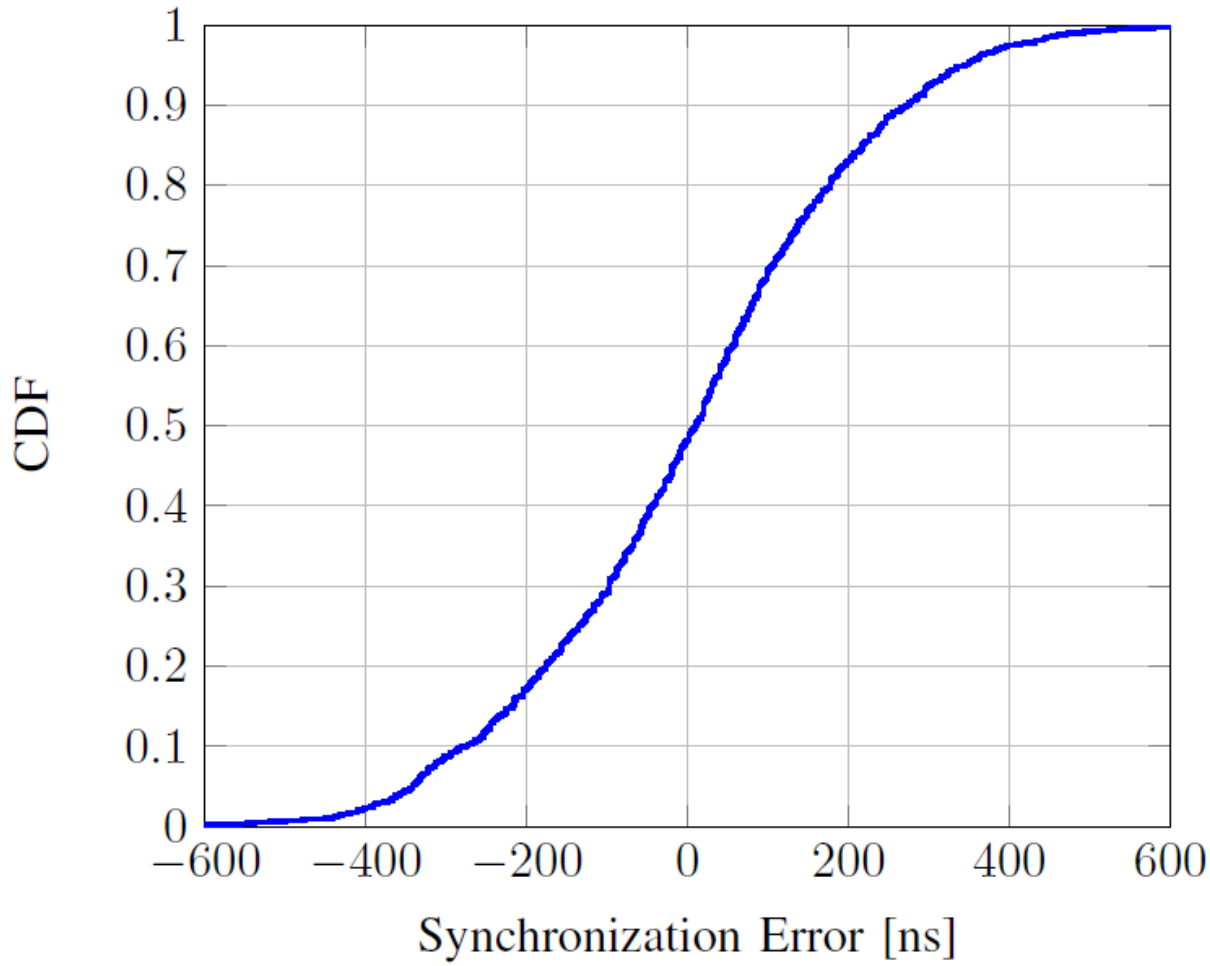


Combined Timestamp



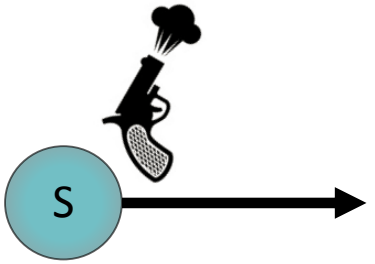
Virtual High-Resolution Time [Schmid et al., 2010]

Clock Synchronization Result



- $|e_{clock}| < 0.05\ \mu s$ in 20% of cases
- $|e_{clock}| < 0.25\ \mu s$ in 75% of cases

Transmission Timing ($e_{transmit}$)



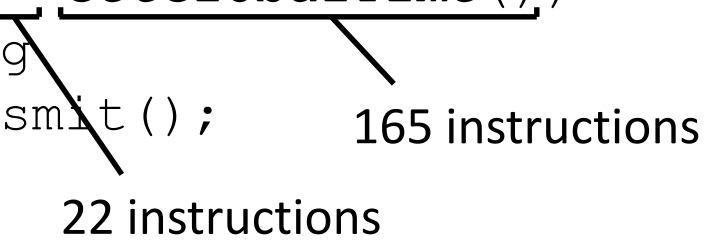
- Measure: TXON command → SFD pin ("start of frame delimiter")

$t_{SFD} - t_{STXON}$	Instance Count	Fraction of Total
1593	15	0.26%
1594	893	15.71%
1595	3726	65.53%
1596	1049	18.45%
1597	3	0.05%
Σ	5686	100.00%

(DCO ticks)

Naïve Transmission Timing

```
if (TargetTime - GetGlobalTime() < 10 ms) {
    while (TargetTime > GetGlobalTime())
        ; // do nothing
    cc2420_driver.transmit();
}
```



Split Clock Transmission Timing

```
void await(uint64_t local_target) {
    uint16_t target_tarof = (local_target >> 23) & 0xFFFF;
    uint16_t target_tar   = (local_target >>  7) & 0xFFFF;

    while (TAR_overflows < target_tarof)
        ;
    while (TAR < target_tar)
        ;

    uint16_t target_tbr = (local_target & 0x007F) + TBCCR6;
    while (TBR < target_tbr)
        ;
}


await(GetLocalTime(TargetTime));
```

New Loop Size

```
.L24:
    mov &__TBR, r15  ; 3 cycles
    cmp r12, r15     ; 1 cycle
    jlo .L24         ; 2 cycles
```

New maximum error: 6 cycles

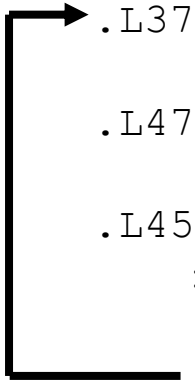
Loop Alignment

```
switch ((target_tbr - TBR) % 8) {
case 7:
    _NOP();
case 6:
    _NOP();
case 5:
    _NOP();
case 4:
    _NOP();
case 3:
    _NOP();
case 2:
    _NOP();
case 1:
    _NOP();
case 0:
default:
    ;
}
while (TBR < target_tbr) {
    _NOP();
    _NOP();
}
```

the jump table (omitted)

```
    and #7, r15
    rla r15
    br .L34(r15)

.L33:
    nop                 ; 1 cycle
.L32:
    nop                 ; 1 cycle
.L31:
    nop                 ; 1 cycle
.L30:
    nop                 ; 1 cycle
.L29:
    nop                 ; 1 cycle
.L37:
    nop                 ; 1 cycle
.L47:
    nop                 ; 1 cycle
.L45:
    mov &__TBR, r15     ; 3 cycles
    cmp r12, r15        ; 1 cycle
    jlo .L37            ; 2 cycles
```



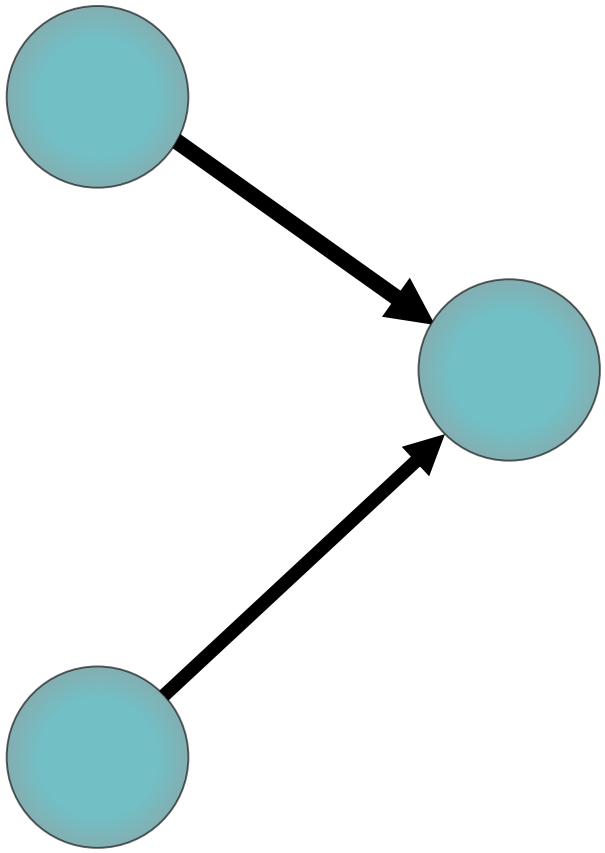
New maximum error: 1 cycle!

Summary of Error Sources

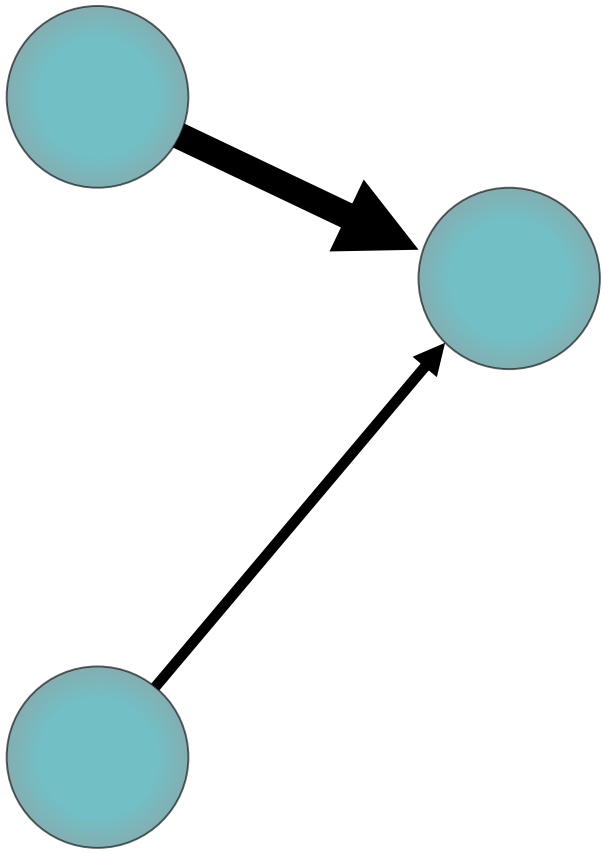
- $|e_{clock}|$ < 0.25 µs (in 75% of cases)
- $|e_{transmit}|$ < 0.25 µs = 1 cycle
- $|e_{travel}|$ < 0.033 µs = 10 m/c

- $|e_{total}| = |e_{clock} + e_{transmit} + e_{travel}| \overset{!}{<<} 0.5$ µs

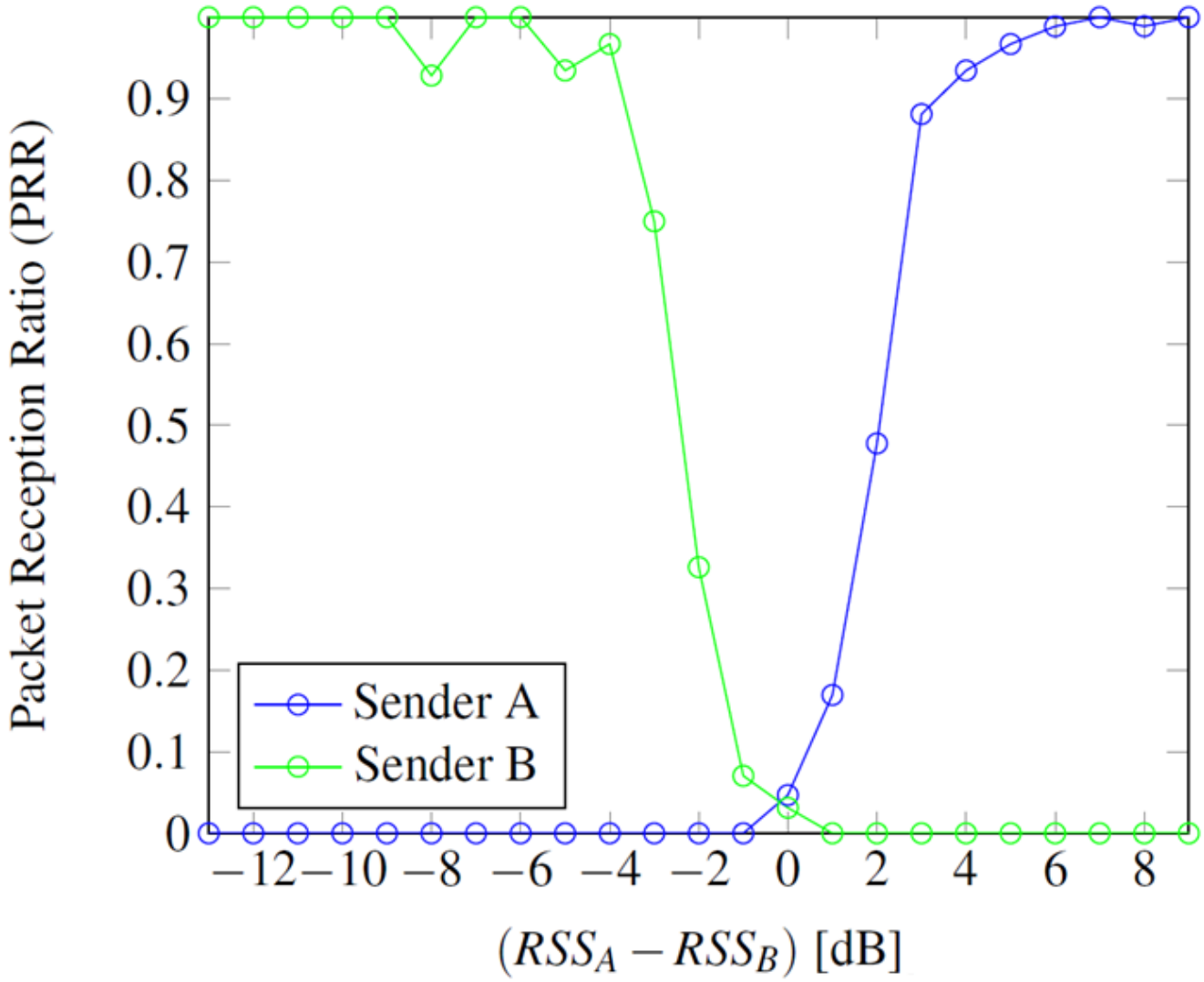
Excursion: The Capture Effect



Excursion: The Capture Effect

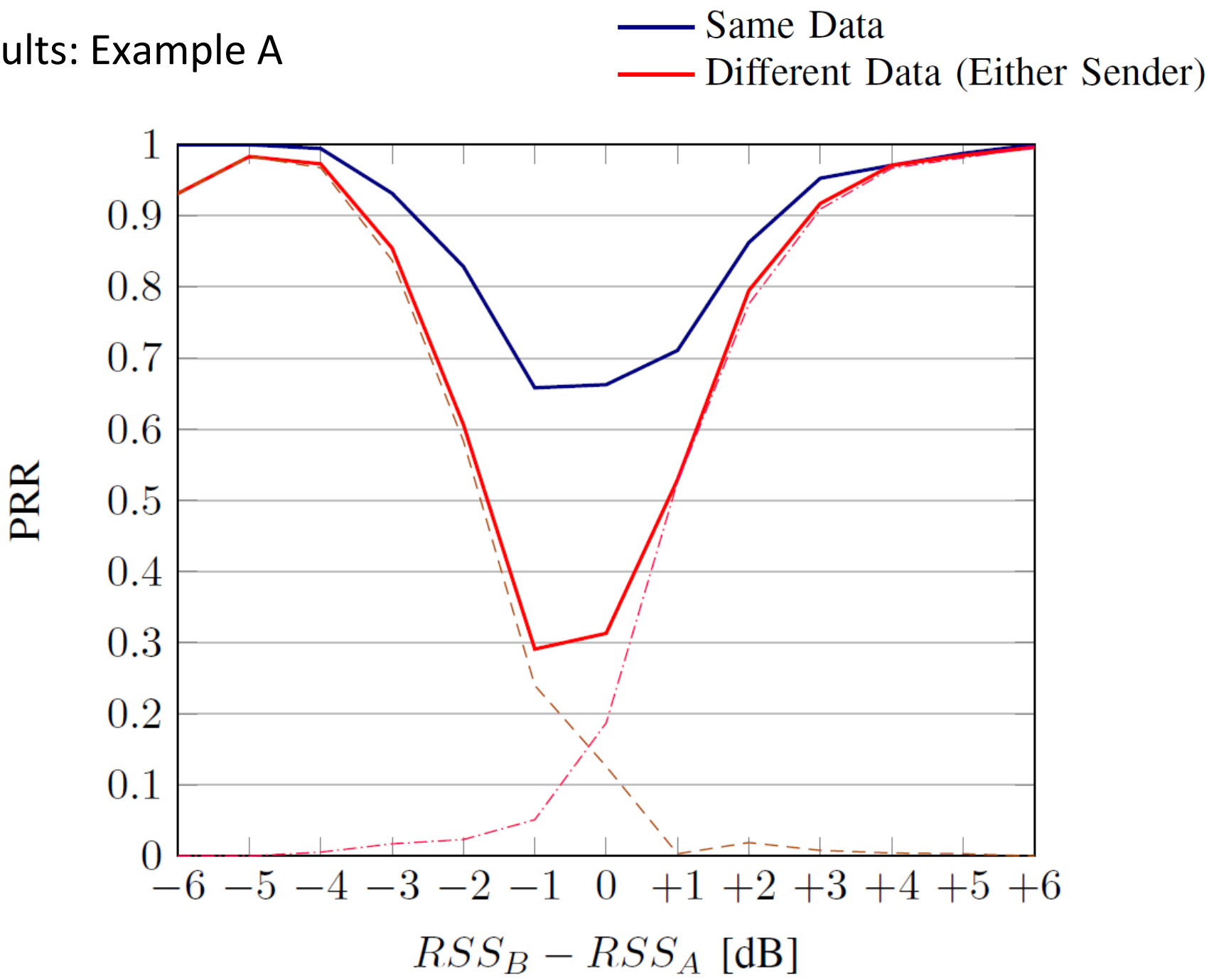


The Capture Effect – Power Difference

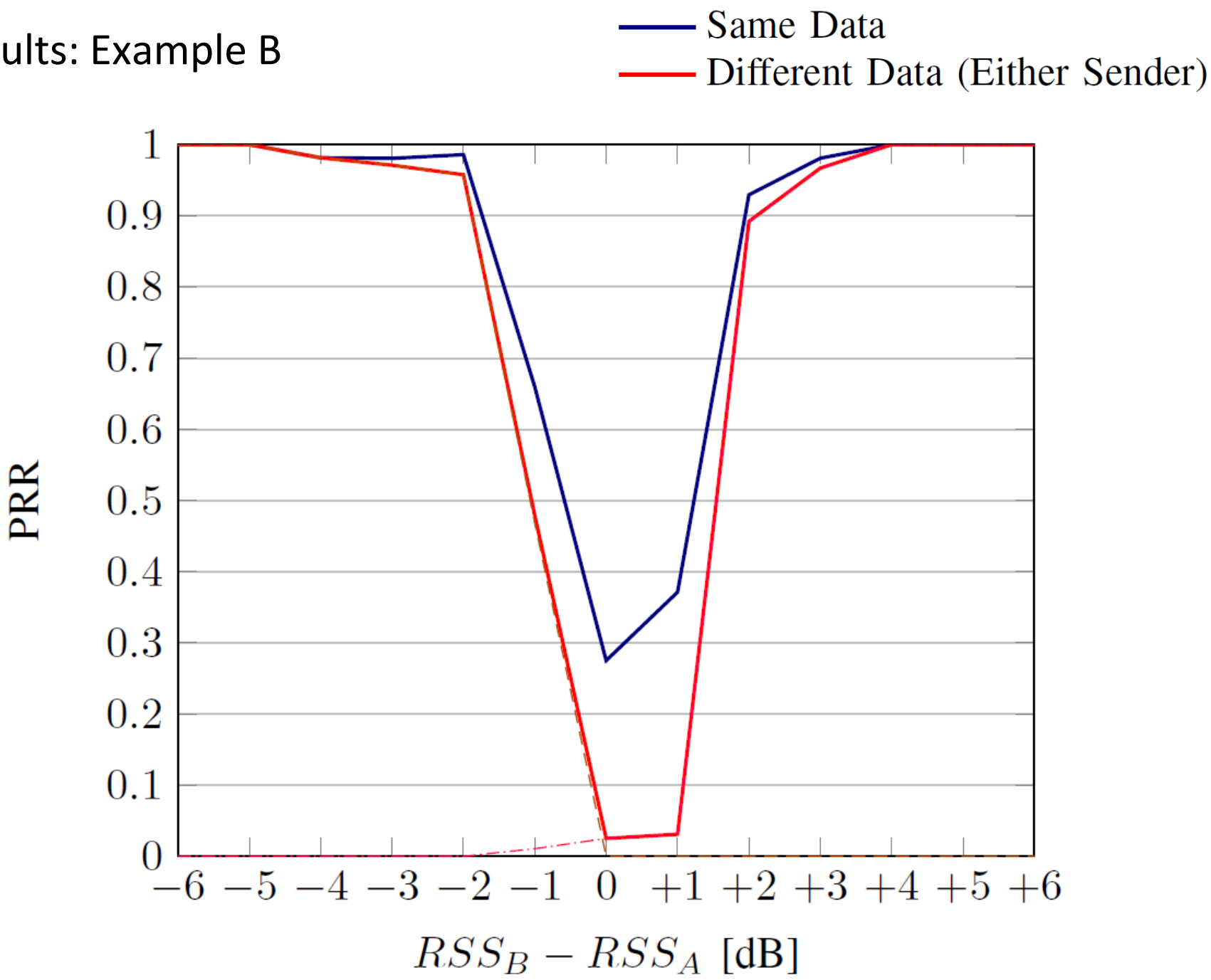


["Sharing a Medium Between Concurrent Protocols Without Overhead Using the Capture Effect", 2016]

Results: Example A



Results: Example B



Results with 3 Senders

$RSS_2 - RSS_3$ [dB]

$RSS_1 - RSS_3$ [dB]

Same Data	0	1	2	3	4	5
0						
1	32%	35%				
2	39%	35%	34%			
3	67%	36%	40%	41%		
4		43%	48%	41%		
5	58%	59%	51%	46%	42%	56%

Different Data	0	1	2	3	4	5
0						
1	0%	2%				
2	0%	0%	1%			
3	7%	3%	2%	0%		
4		14%	9%	8%		
5	83%	50%	21%	3%	3%	0%

Results with 3 Senders

Var($RSS_{>1}$) [dB]

$RSS_1 - RSS_{avg}$ [dB]

Same Data	0	2	6	10
0	34%			
2	36%	38%	56%	
4	51%	48%	52%	50%
6	70%	91%	78%	62%
8				100%

Different Data	0	2	6	10
0	2%			
2	1%	2%	0%	
4	23%	10%	12%	0%
6	80%	78%	50%	24%
8				67%

Results with 4 Senders

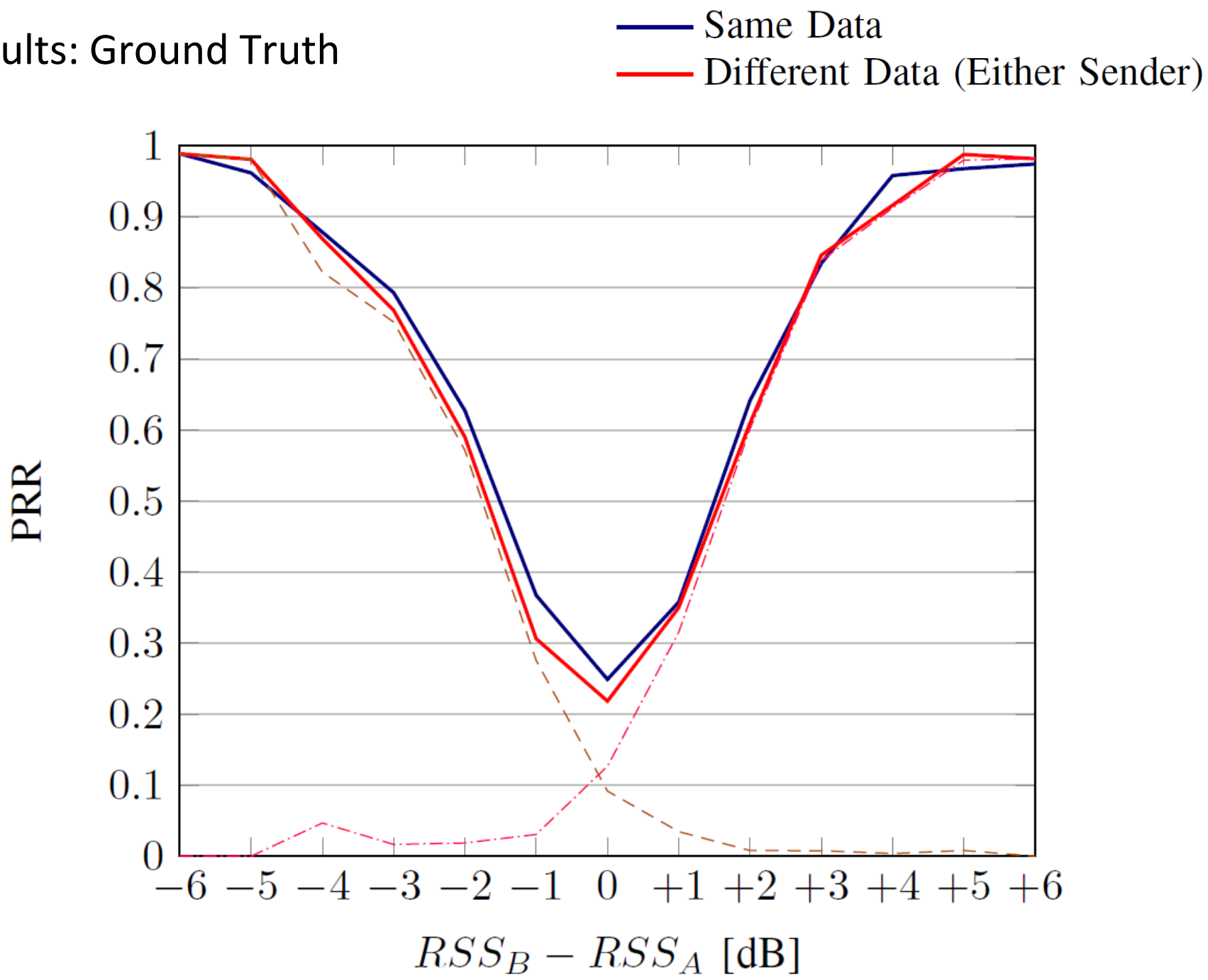
$Var(RSS_{>1})$ [dB]

$RSS_1 - RSS_{avg}$ [dB]

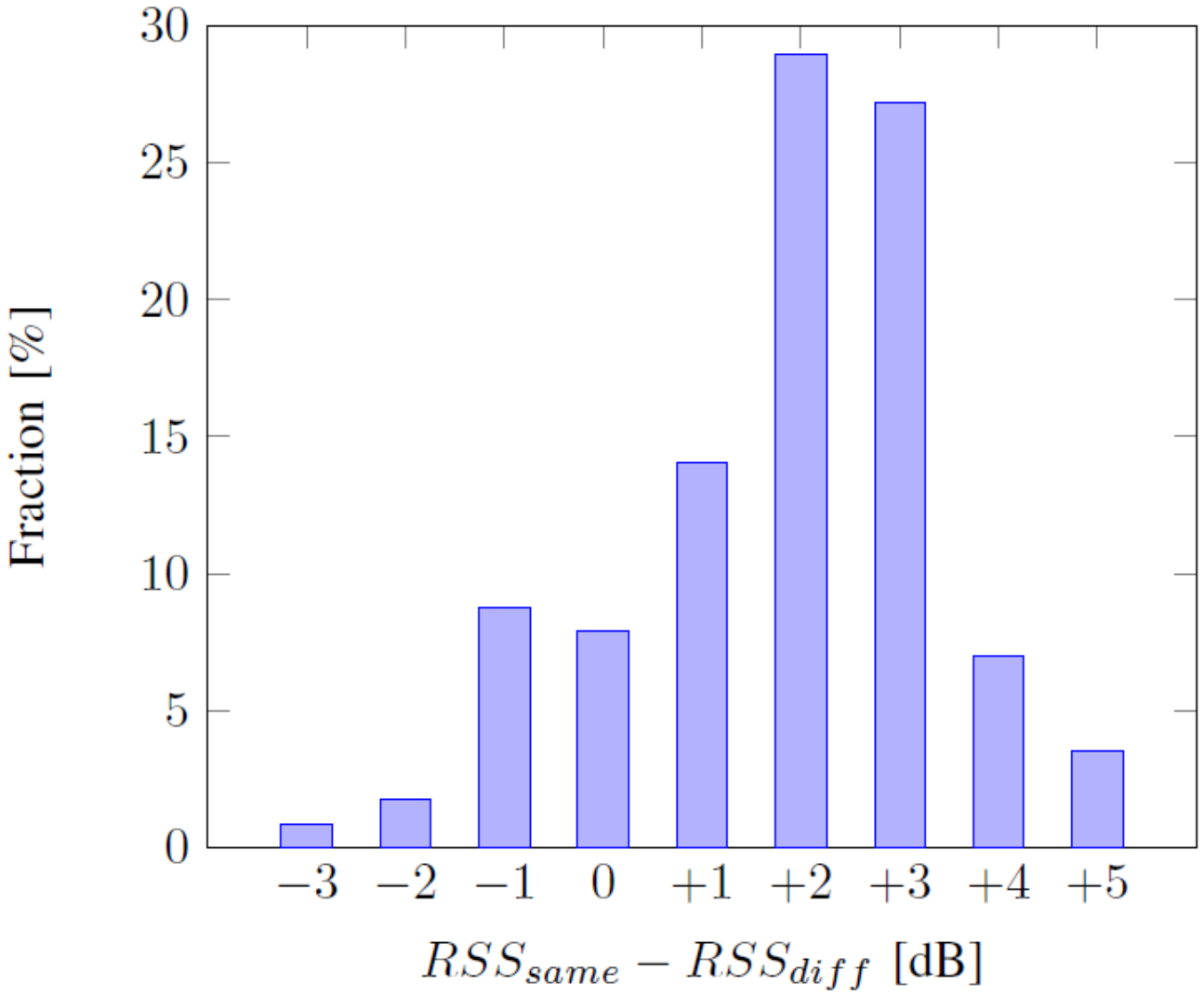
Same Data	0	2	4	6	8	10
0	10%					
2	23%	24%	36%	21%		17%
4	30%	32%	30%	18%	9%	56%
6	25%	34%	36%	33%	22%	
8	50%	50%	50%			

Different Data	0	2	4	6	8	10
0	0%					
2	1%	0%	0%	0%		0%
4	1%	1%	2%	0%	0%	0%
6	4%	6%	9%	6%	6%	
8	30%	28%	29%			

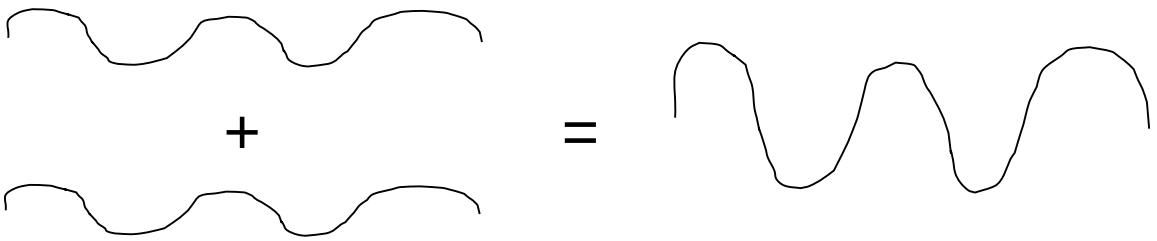
Results: Ground Truth



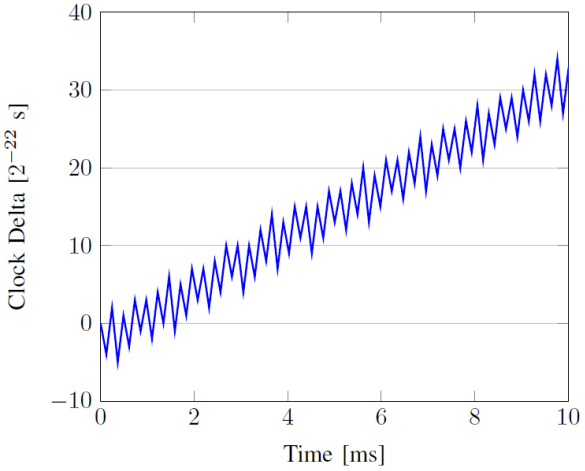
RSS Gain



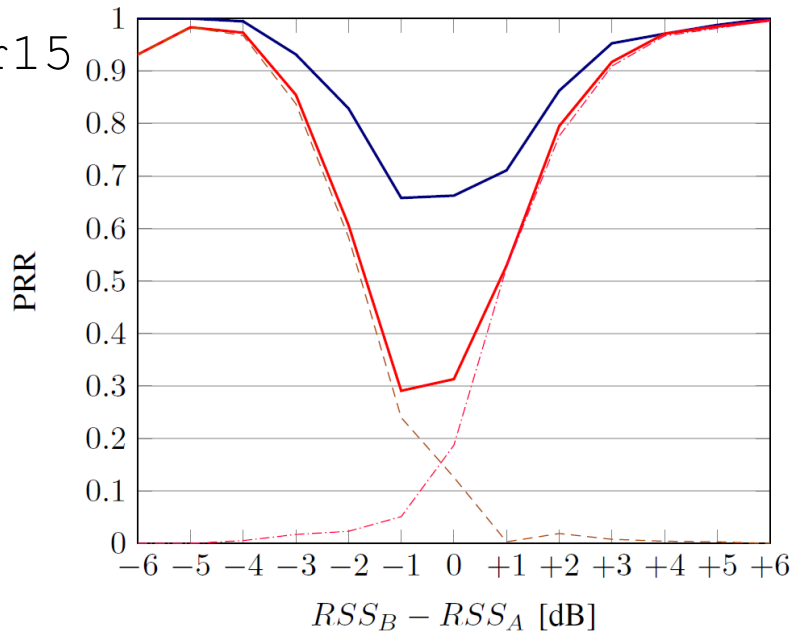
Summary



$$|e_{total}| = |e_{clock} + e_{transmit} + e_{travel}| \overset{!}{\ll} 0.5\ \mu s$$




```
.L24:
mov &__TBR, r15
cmp r12, r15
jlo .L24
```



Questions

Michael König