iFrag
Interference-Aware Frame Fragmentation Scheme for Wireless Sensor Networks

Basem Shihada
CEMSE, KAUST

5th Annual IAMCS Spring Symposium **Tuesday May 13th 2013**

Problem Statement

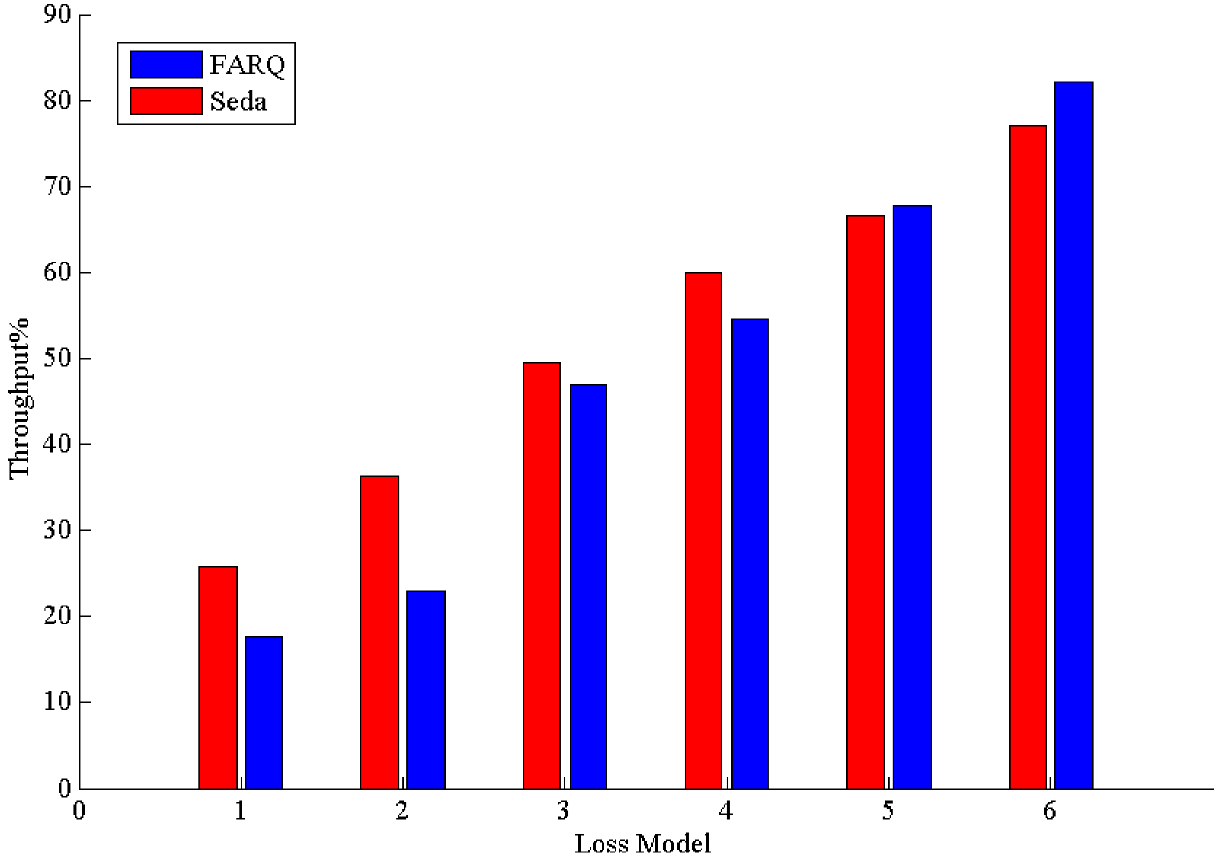
- A single bit error → frame retransmission
- Waste of bandwidth and energy
- Frame fragmentation, but how many fragments?
- More fragments → increased overhead
- Less fragments → bigger chunks for retransmissions

Traditional Solution

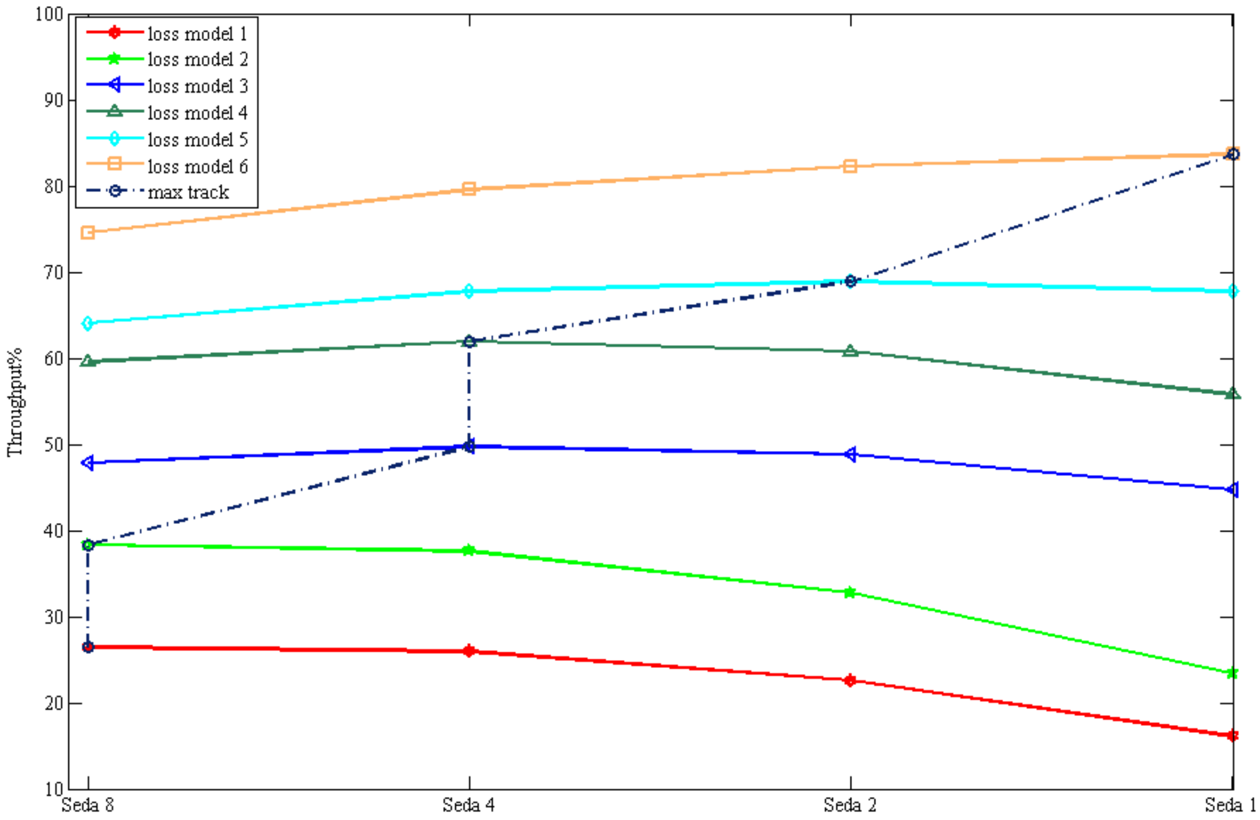
- Divide the frame into several fixed size blocks
- Each block has CRC + block seq#
- Number of fragments is predetermined (static) regardless of the channel condition

Frame fragmentation has to be dynamic and incorporate the wireless channel conditions

Why Not Fixed# of Blocks?



Why Not Fixed # of Blocks?



iFrag Design

- Dynamically adjusts block size based on channel condition[†]
- Network delay and energy are considered
- Depending on iFrag mode, data frames are composed of one or more block(s)
- 4 implemented iFrag modes, iFrag 1, iFrag 2, iFrag 4 and iFrag 8, where 1, 2, 4 and 8 represent # of blocks in a frame

---

[†] A. Showail, A. El-Rasad, A. Meer, A. Daghistani, K. Jamshaid, and B. Shihada, "iFrag: Dynamic Partial Packet Recovery for Sensor Systems", *ACM Wireless Journal,* Submitted, 2013.

iFrag Frame Structure

Seq# 1 Byte	Block 1 96 Bytes	CRC 1 Byte

iFrag 1

Seq# 1 Byte	Block 1 48 Bytes	CRC 1 Byte	Seq# 1 Byte	Block 2 48 Bytes	CRC 1 Byte

iFrag 2

Seq# 1 Byte	Block 1 24 Bytes	CRC 1 Byte	Seq# 1 Byte	Block 2 24 Bytes	CRC 1 Byte	…	Seq# 1 Byte	Block 4 24 Bytes	CRC 1 Byte

iFrag 4

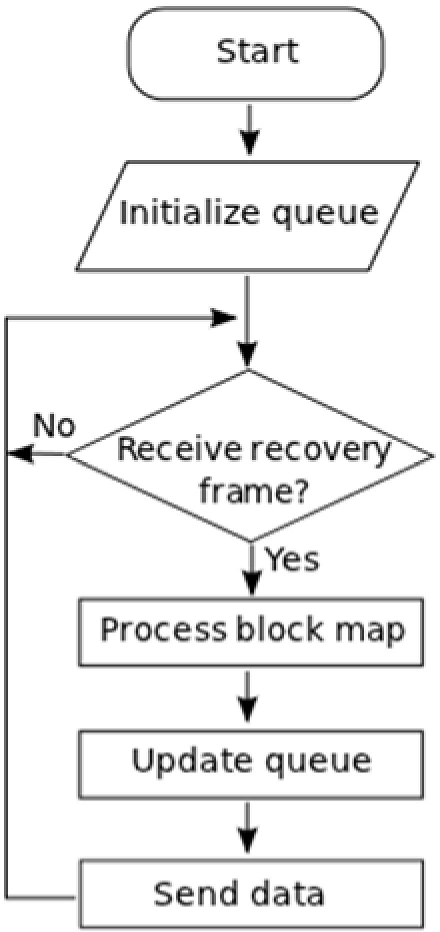
Seq# 1 Byte	Block 1 12 Bytes	CRC 1 Byte	Seq# 1 Byte	Block 2 12 Bytes	CRC 1 Byte	Seq# 1 Byte	Block 3 12 Bytes	CRC 1 Byte	…	Seq# 1 Byte	Block 8 12 Bytes	CRC 1 Byte

iFrag 8

SBN 1 Byte	Block Count 1 Byte	Block Map 4 Bytes	CRC 1 Byte

Recovery Frame

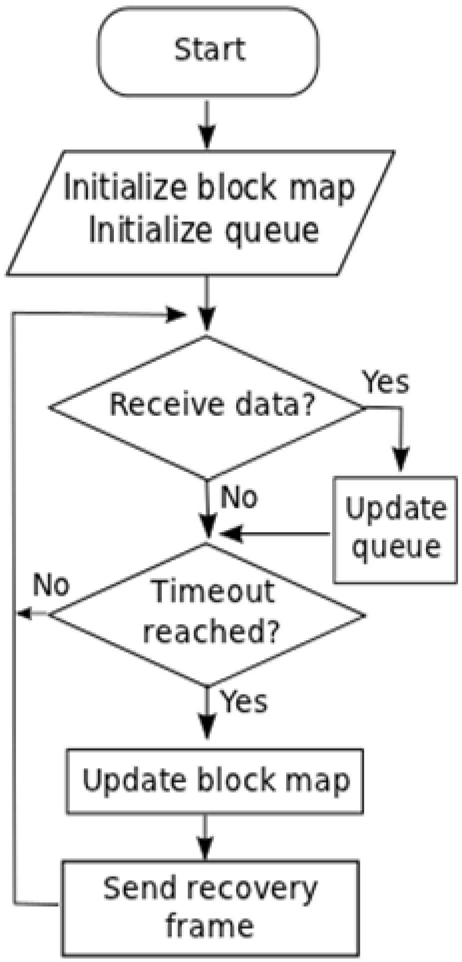
iFrag Operations - Sender



Algorithm 1: iFrag Sender

```
Initiate connection and inform receiver of supported modes
Divide network layer packet into blocks
Add block sequence number and CRC, reframe and handoff packet to MAC layer for transmission
if recovery frame received then
    Update PRR
    if Session is starting then
        Select iFrag mode:
        if PRR > threshold_good then
            Switch to the next mode with bigger block size
        else if PRR < threshold_bad then
            Switch to the next mode with smaller block size
        Reset PRR
    else
        Keep using the same mode
    Retransmit requested blocks (with new blocks if any) as determined from the BlockMap field of the recovery frame
```

iFrag Operations - Receiver



Algorithm 2: iFrag Receiver

```
Connection establishment (know Sender
supported modes)
Send recovery frame that includes BlockMap
and SBN to request for frames from the Sender
if data frame received then
    Identify iFrag mode using frame size
    Identify correct blocks through CRC (each
    block contains consecutive bytes)
    if all blocks are correctly received then
        Re-assemble blocks into a network layer
        packet and hand-off to network layer
    else if some blocks are corrupted then
        reconstruct recovery frame accordingly
        Buffer correctly received blocks
Send recovery frame after each session or after
timeout_recovery period, whichever earlier
Stop sending recovery frame when End
Message received or when no new data has been
received for the period of timeout_end
```

Challenges

1. When to switch between block sizes?
2. How to discover the iFrag mode change without extra signalling?
3. How to ensure data integrity across various recovery modes?

iFrag Switching Approach

Packet Reception Ratio is used to identify the channel error patterns

$$PRR = \frac{\#\ \text{of sent blocks}}{\#\ \text{of correctly received blocks}}$$

over 5 sessions (*i.e.*, 20 frames)

Transitioning from one mode to the next happens in a gradual manner following Threshold policy.

iFrag Switching ThresholdPolicy

- Thresholds for switching from one iFrag mode to the other

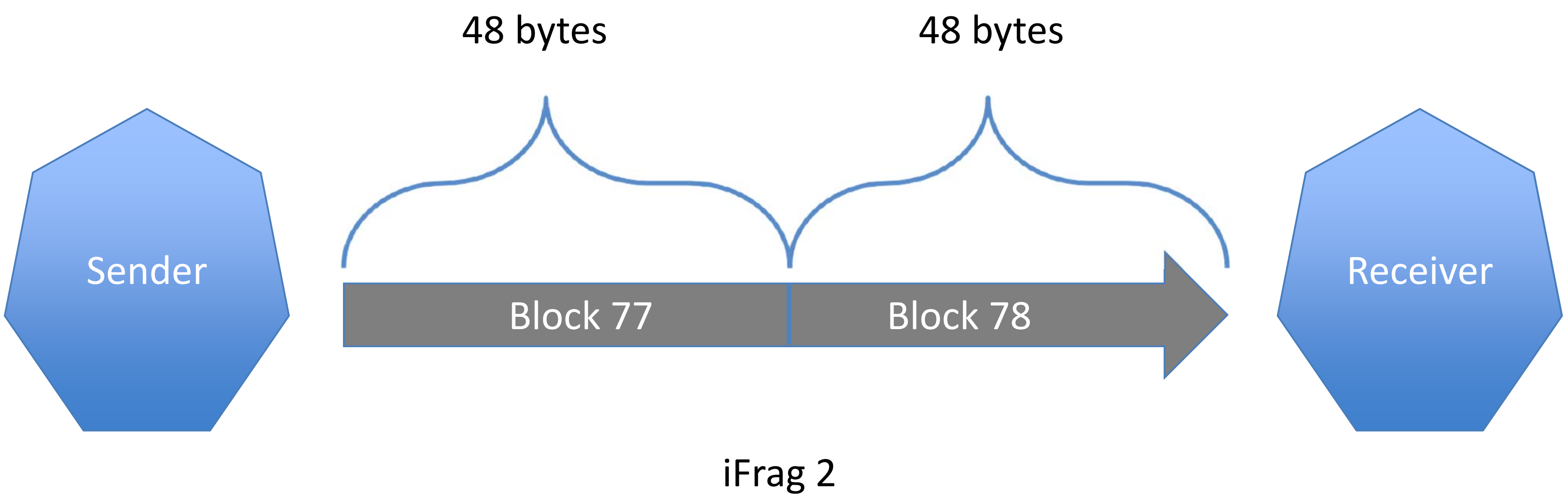
Operational Mode	Channel Condition
iFrag 1	PRR=100%
iFrag 2	80%≤PRR<100%
iFrag 4	50%≤PRR < 80%
iFrag 8	PRR<50%

- The choice of these thresholds was experimentally verified

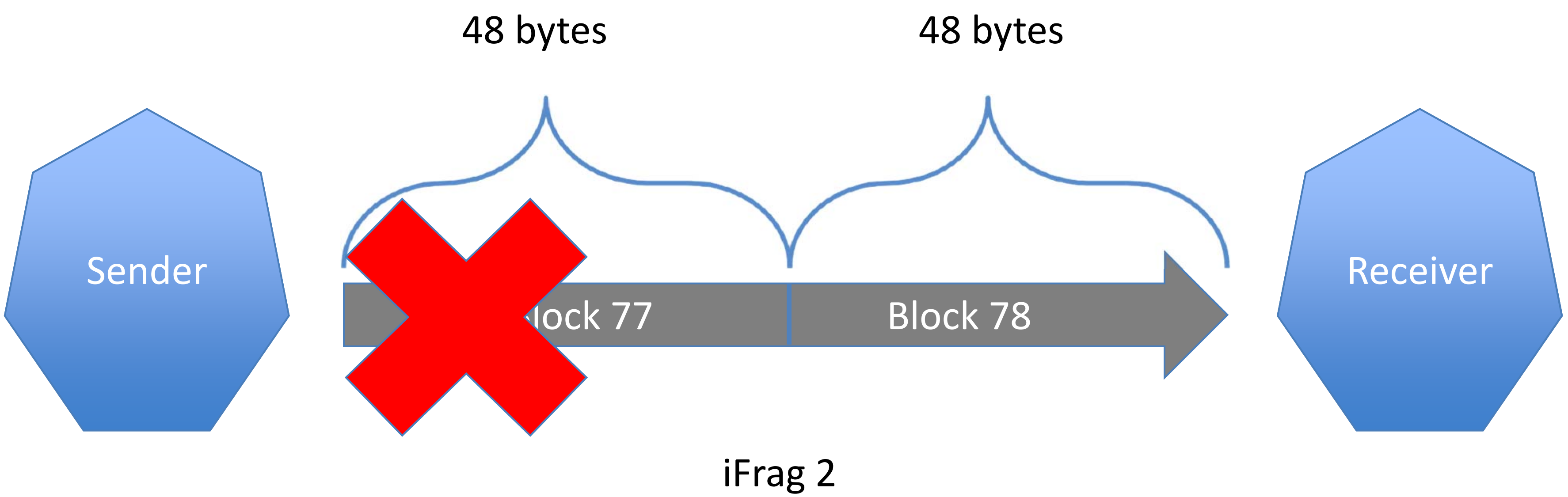
iFrag Mode Discovery Approach

- iFrag allows switching between modes on the fly
- Each mode has different # of blocks, hence different frame sizes.
- The receiver determines iFrag mode based on the received frame size
- No need for a control frame (extra signalling)

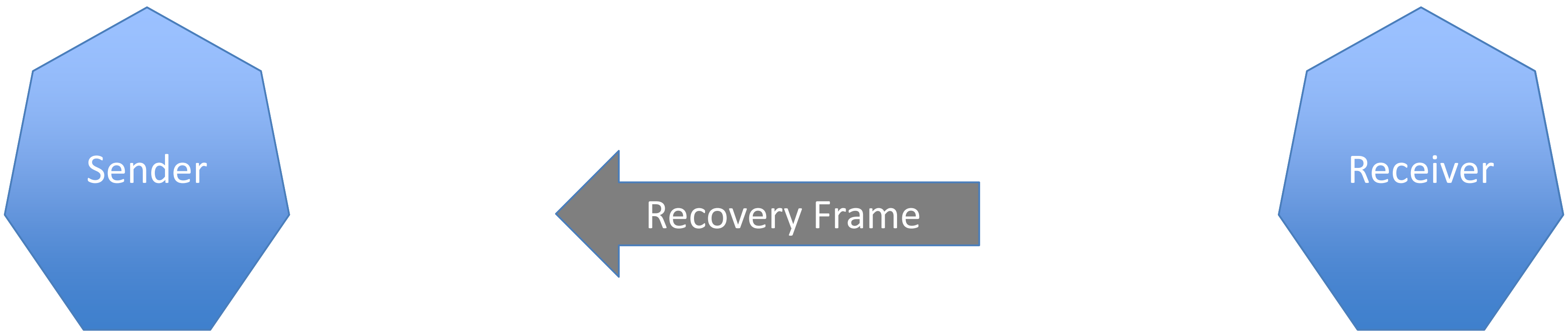
iFrag Data Integrity Problem



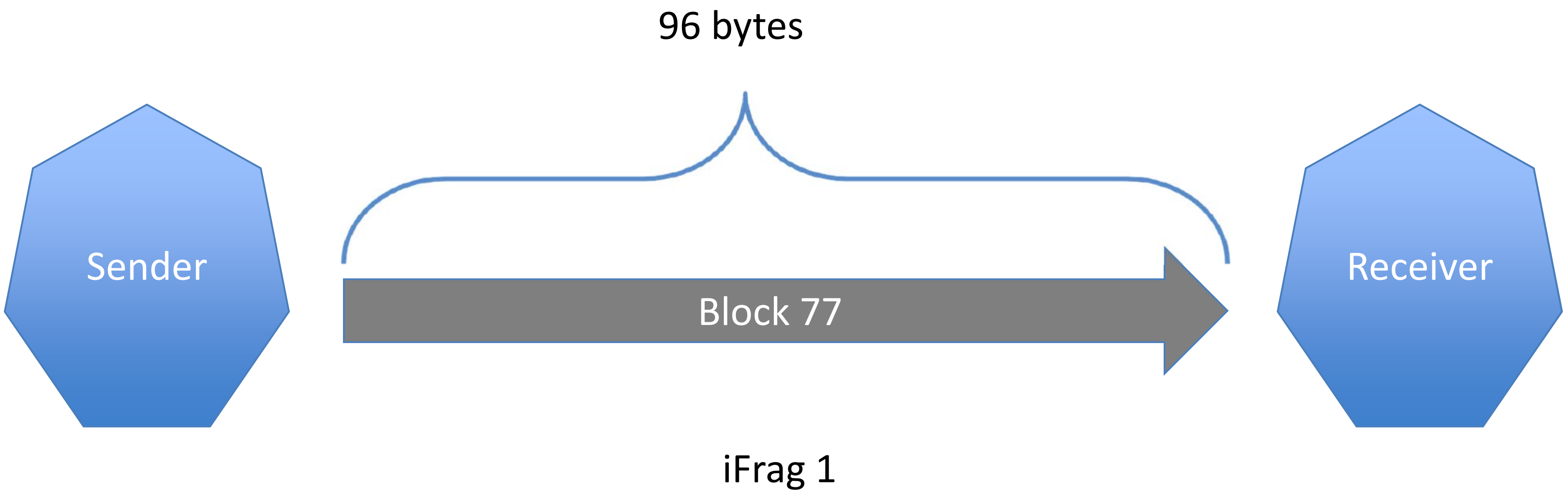
iFrag Data Integrity Problem



iFrag Data Integrity Problem



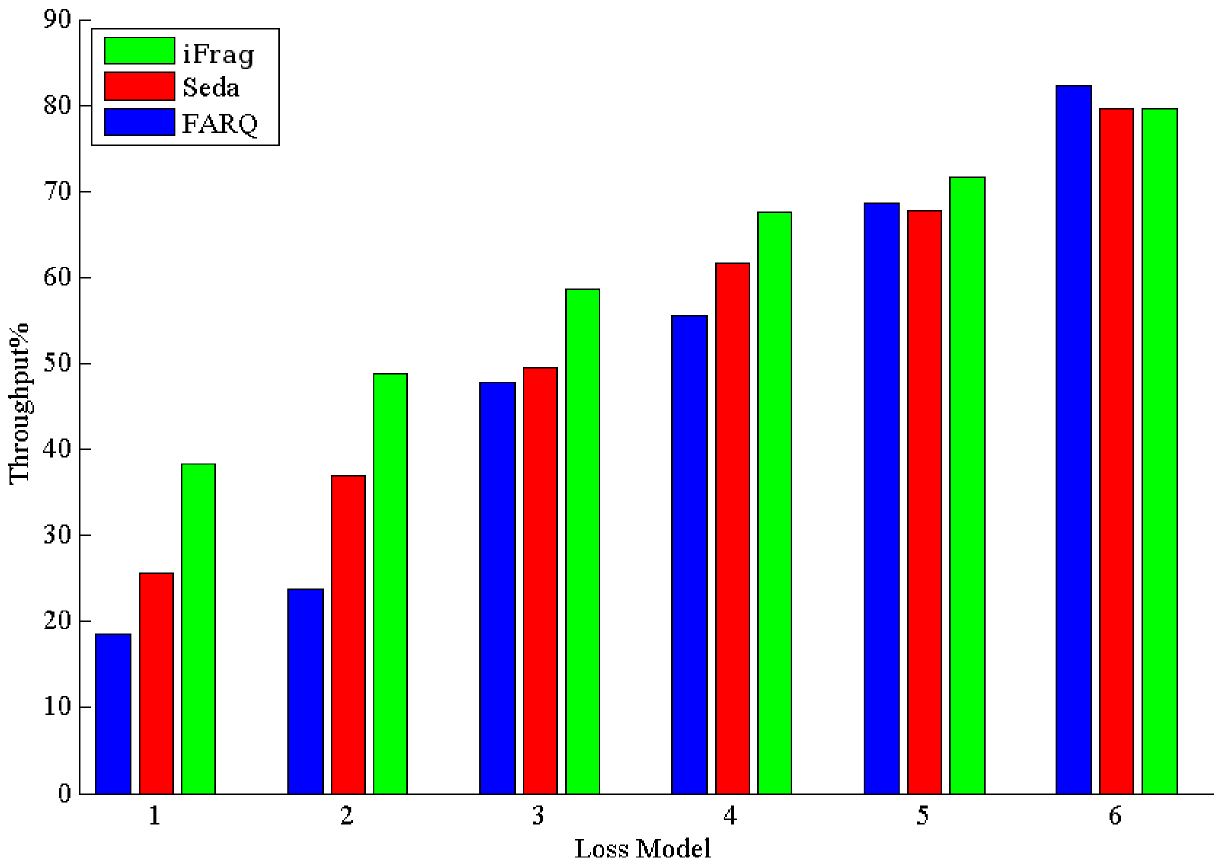
iFrag Data Integrity Problem



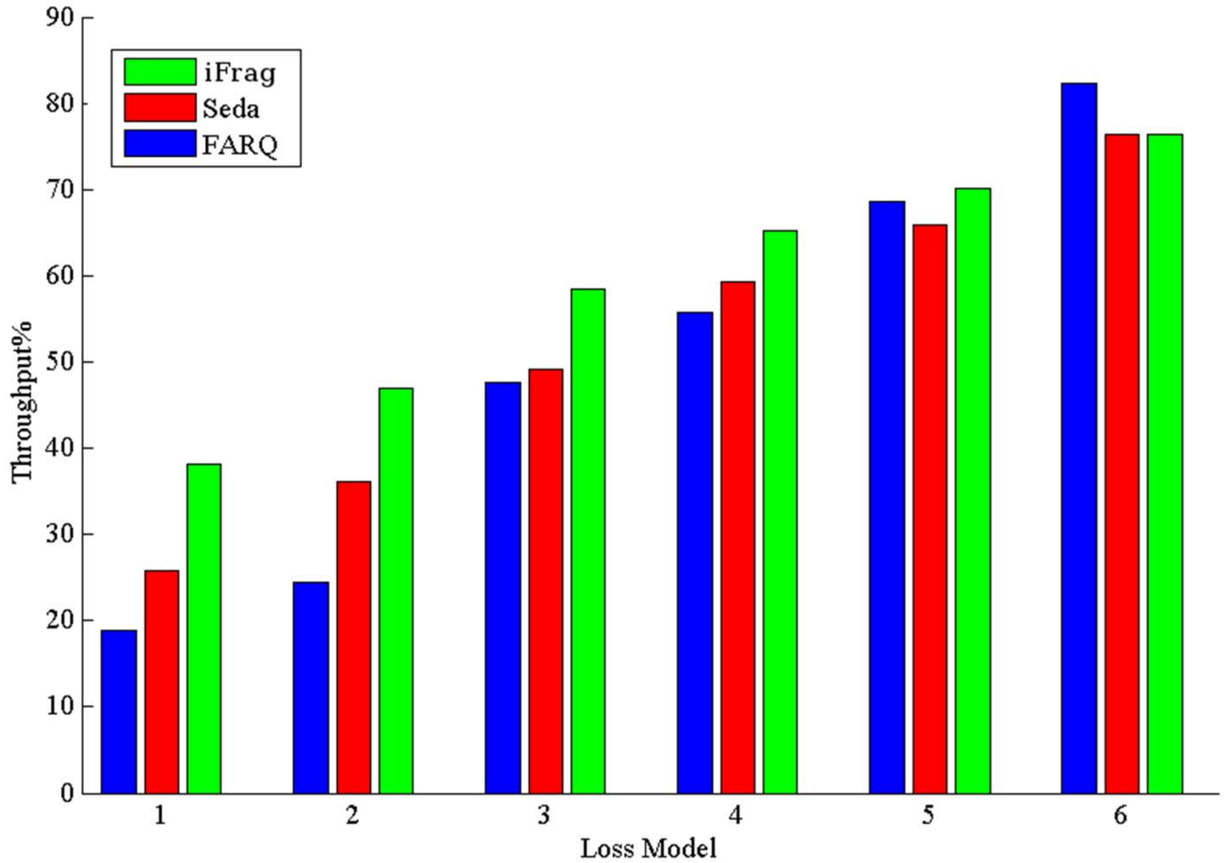
iFrag Data Integrity Approach

- Standardizing the block sizes and the block numbering convention
- All block numbers refer to the first small iFrag 8 block
- Data are represented in the smallest granularity (chunks of 12 bytes where each has its own number).

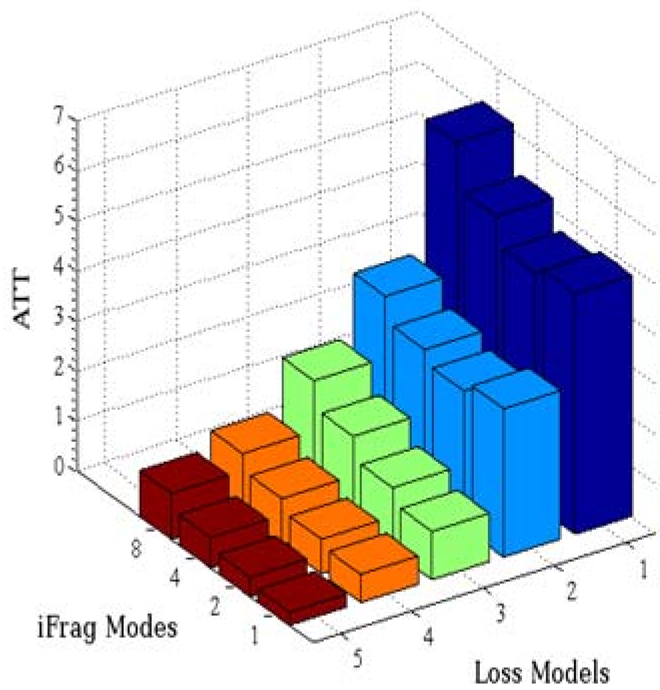
iFrag Analytical Results



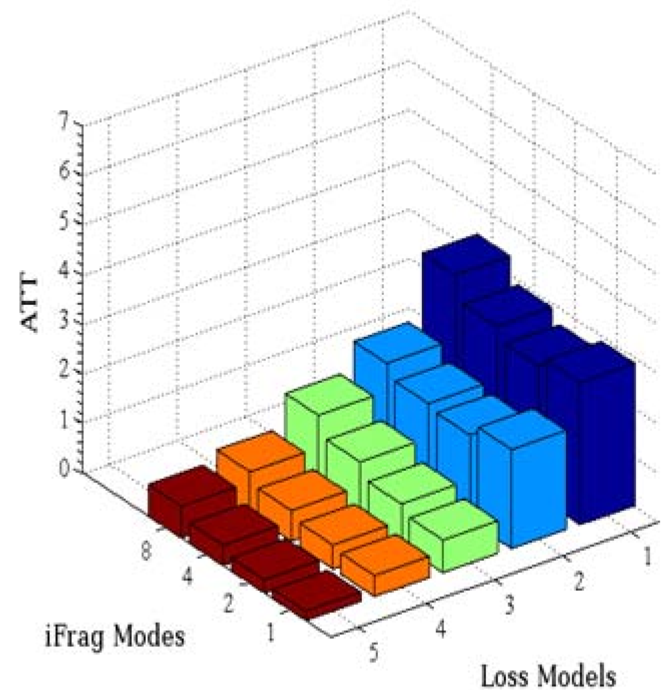
iFrag Simulation Results



iFrag Average Transmission Trials



Seda

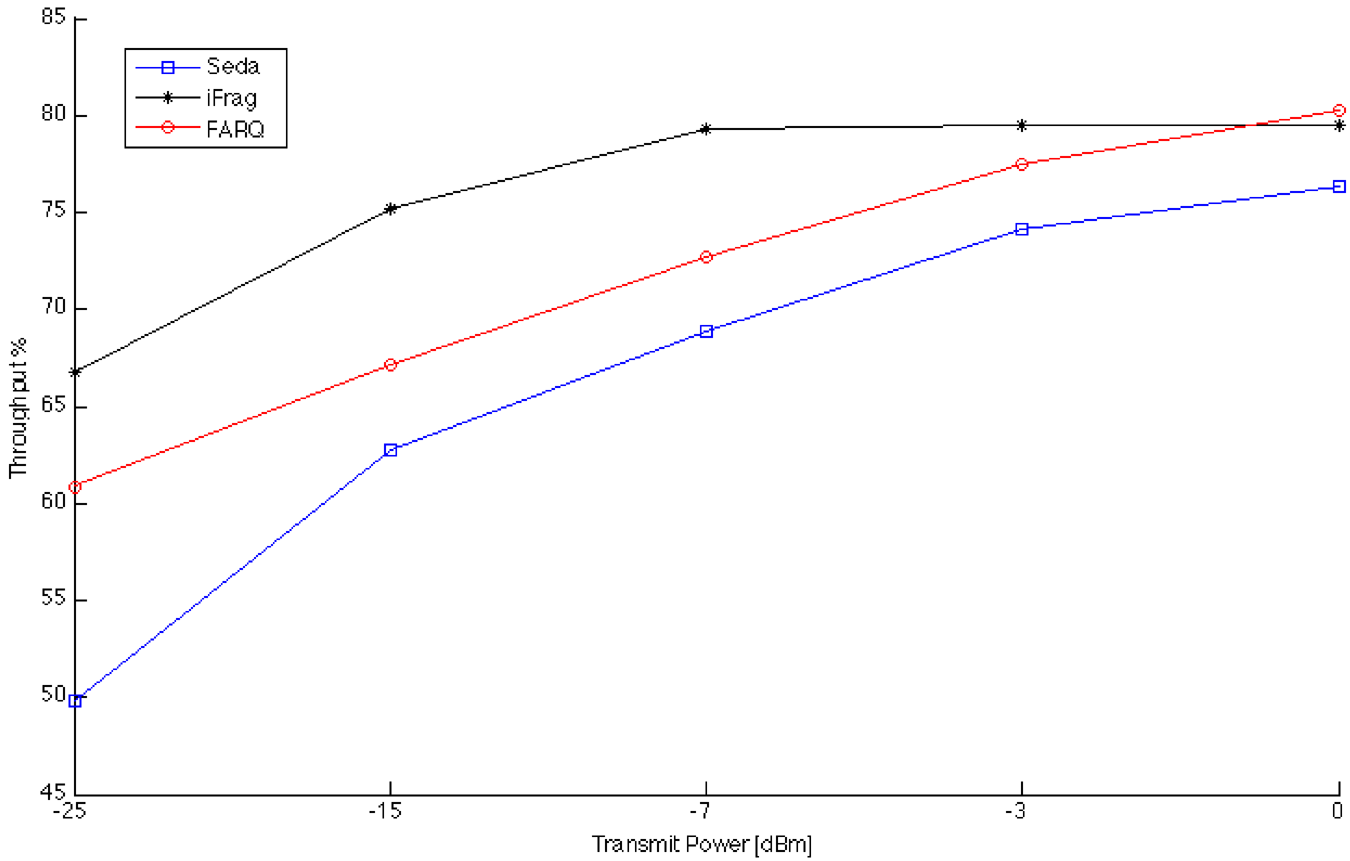


iFrag

Experimental Setup

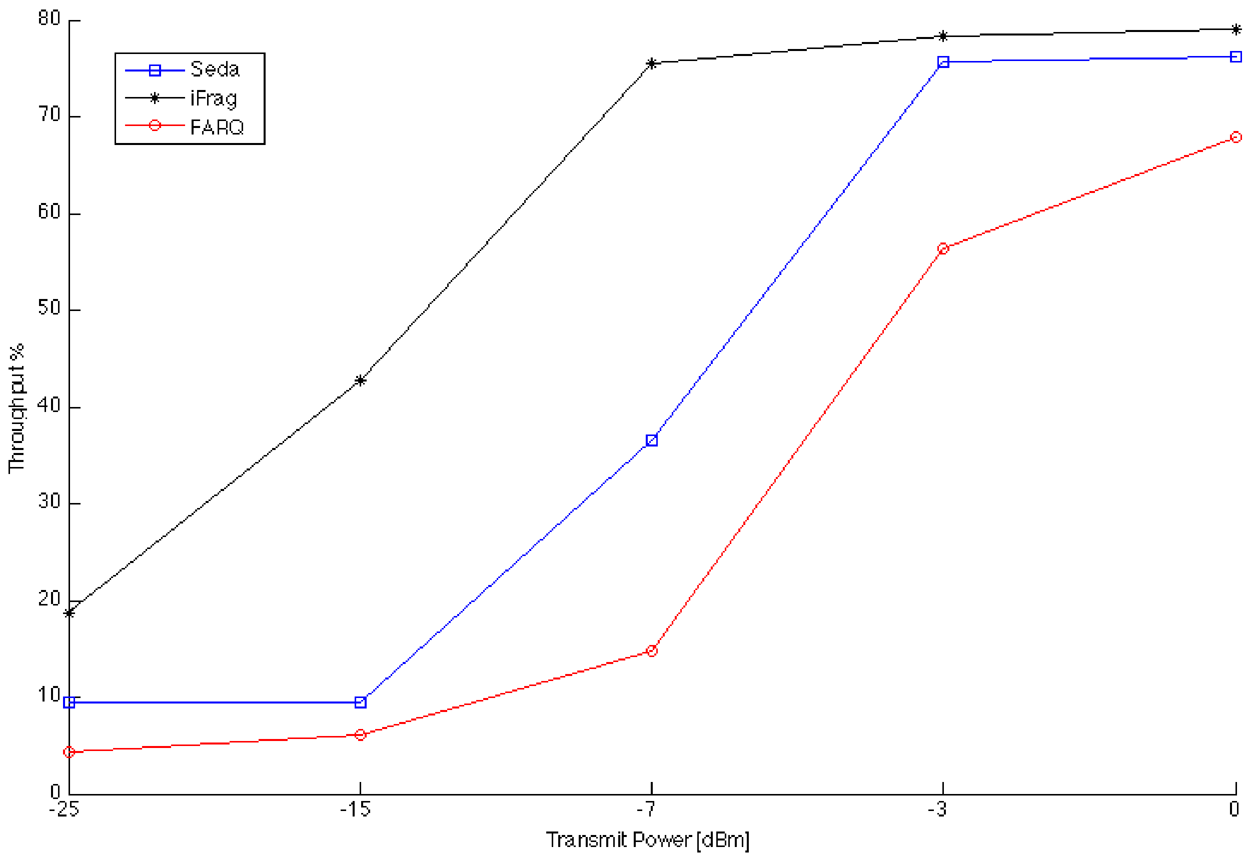
- TelosB with Chipcon- CC2420 (ZigBee)
- TinyOS 2.1.1
- Experiments were performed at night
- Place the motes 1*m* apart from each other
- Motes were powered through USB
- MAC-layer automatic CRC is disabled
- Introduce channel noise through large file transfer over WiFi (20 dBm)
- Sender mote sends 1000 frames to the receiver mote in each run
- Results are then averaged over five runs
- Each frame may have one, two, four, or eight block(s)

iFrag vs. Seda vs. FARQ Throughput **w/o interference**



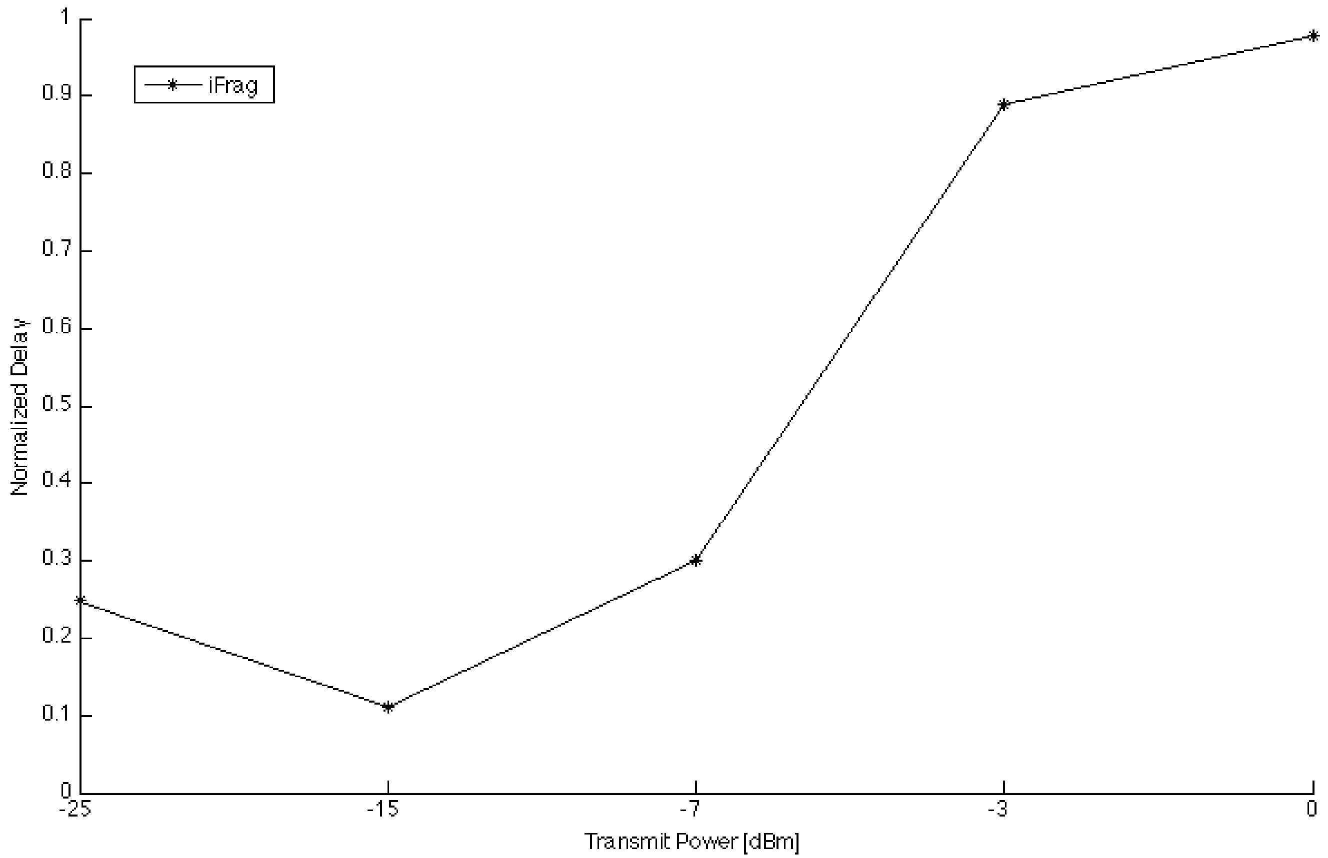
An average of 13% throughput improvement across all channel conditions

iFrag vs. Seda vs. FARQ Throughput **with interference**



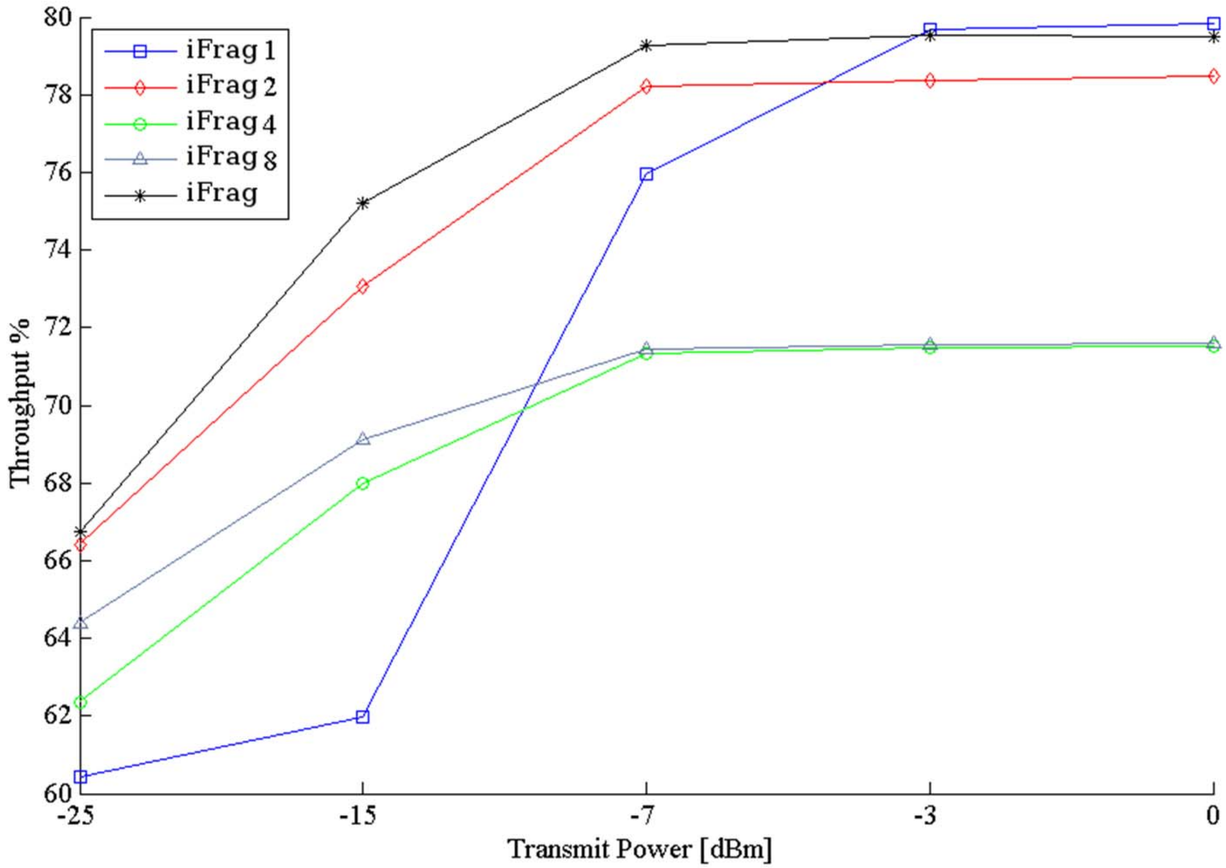
3× increase in throughput compared to Seda in bad channel condition

iFrag Network Delay



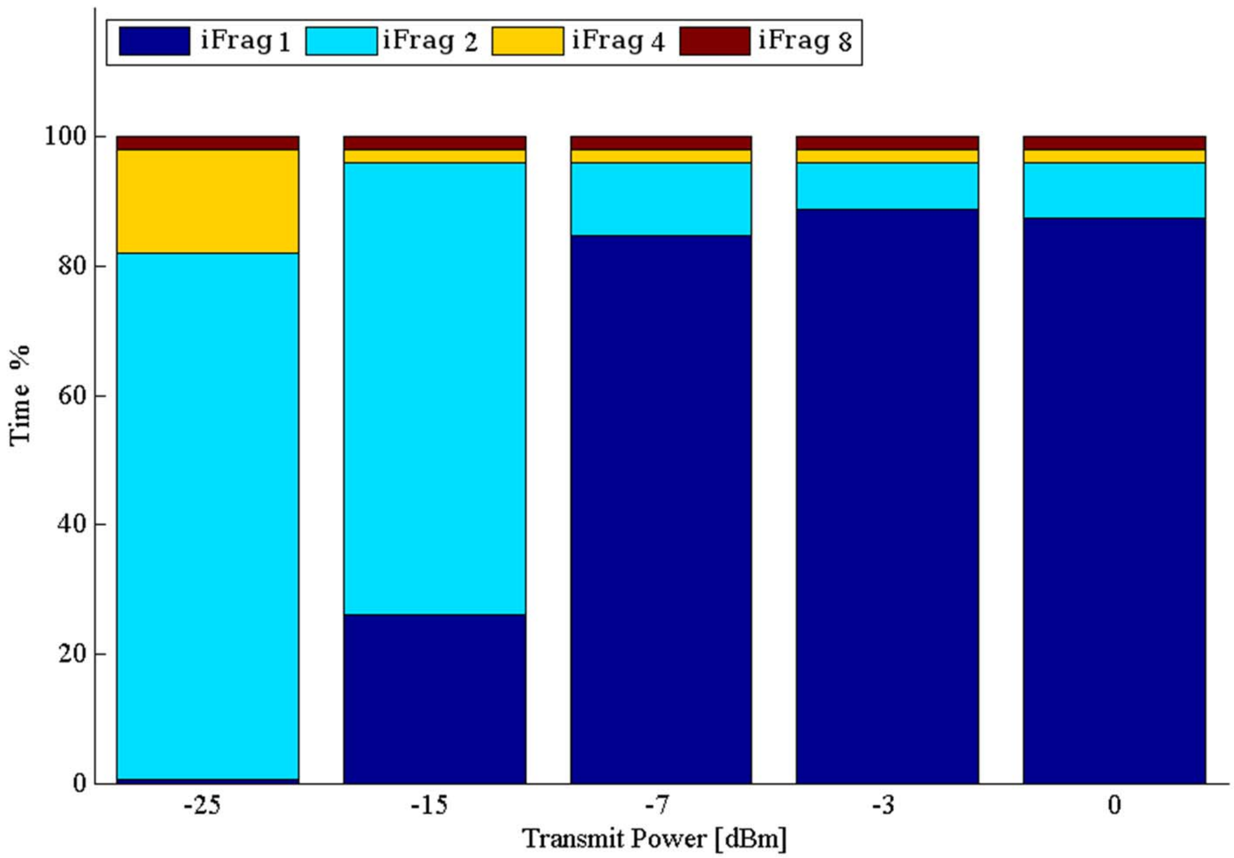
Reduce the network delay by 12%

iFrag vs. Static iFrag Throughput **w/o interference**



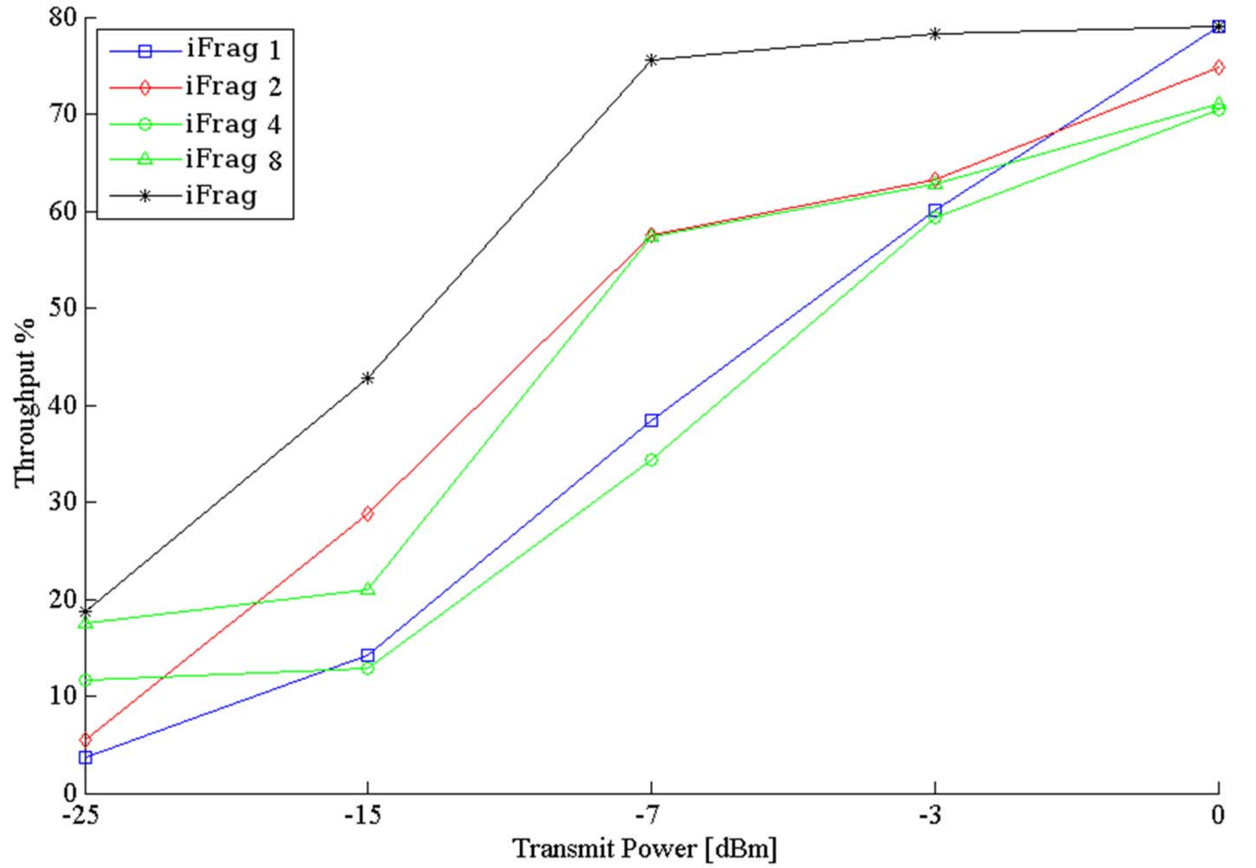
iFrag outperforms all other static ones

Time iFrag is spending in each mode **w/o interference**



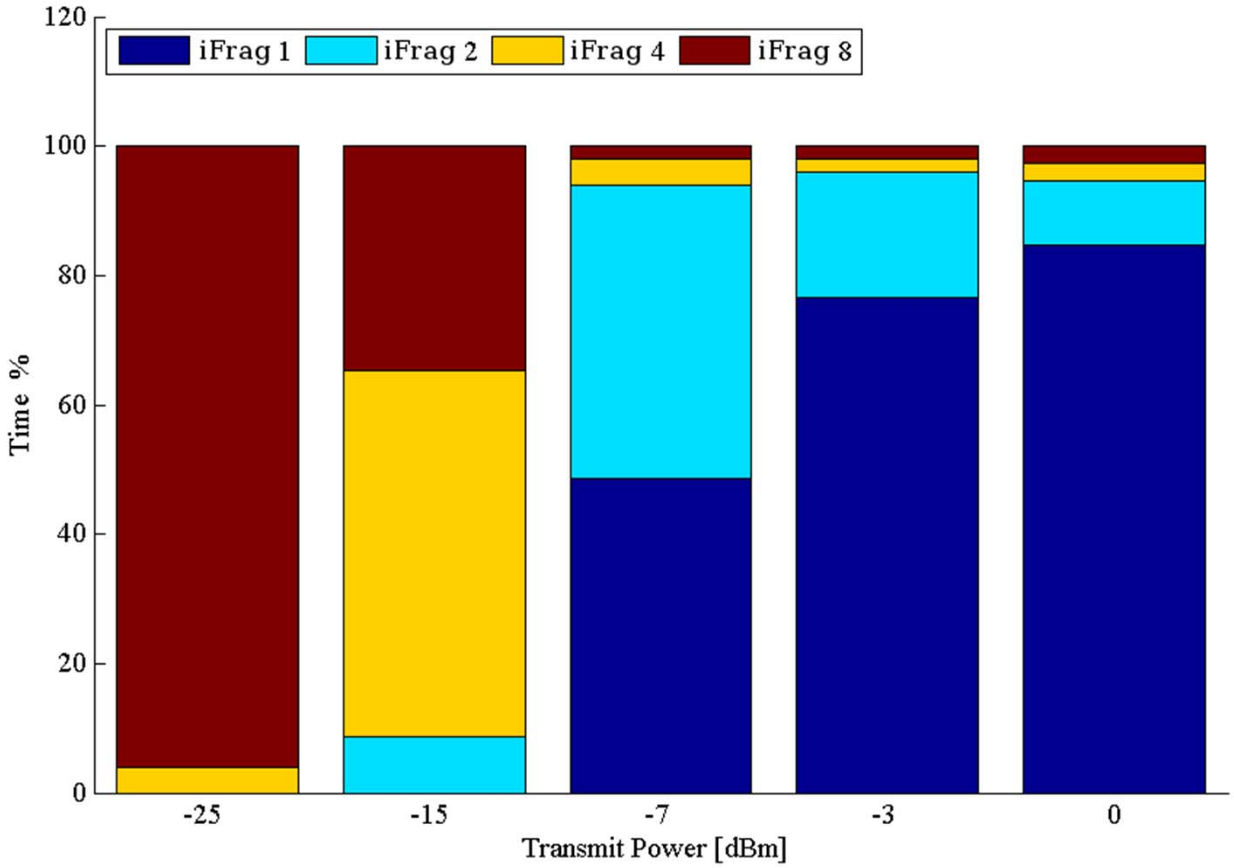
iFrag still spends around 10% of its time in iFrag 2 due to PRR Threshold value

iFrag vs. Static iFrag Throughput **with interference**



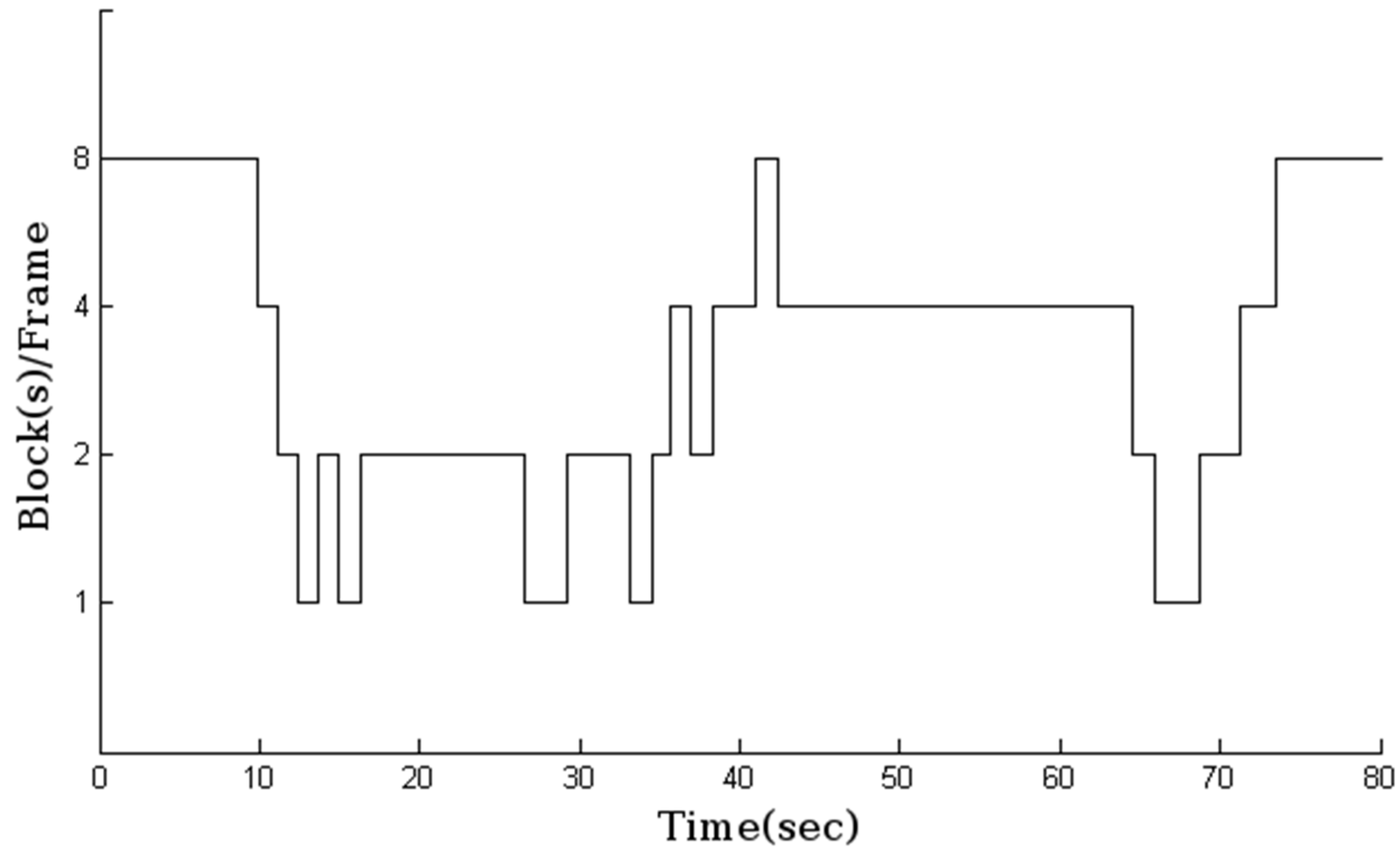
iFrag achieves higher throughput

Time iFrag is spending in each mode **with interference**



iFrag is spending more than 80% of the time in iFrag 1 when the channel is good

Mode Transitions Over Time



Protocol stabilizes for a minimum of one session, where transitions occur gradually.

iFrag, what's next?

- Hybrid iFrag (Hi-Frag[1])
  - Heterogeneous block sizes due to different error patterns
  - Less power consumption
- Green-Frag[2]
  - Energy efficient scheme that combines frame fragmentation techniques with adaptive power mechanism
  - Selects the best transmit power according to the channel conditions and error patterns

---

[1] A. Meer, A. Daghistani, and B. Shihada, "Hi-Frag: Hybrid Interference-Resilient Frame Fragmentation for Wireless Sensor Networks", *IEEE Globecom,* Submitted, 2013.

[2] A. Daghistani and B. Shihada, "Green-Frag: Energy Efficient Frame Fragmentation Scheme for Wireless Sensor Networks ", *IEEE WiMob,* In Progress, 2013.

Conclusion

- Static frame fragmentation is not suitable for channels with varying quality.
- iFrag dynamically selects # of blocks in each frame based on the channel condition
- iFrag limits unnecessary data retransmissions by sending periodic recovery frames
- iFrag increases throughput by 13% on average while reducing delay by 12% compared to static fragmentation approach

KAUST NETLAB

Thanks!

http://netlab.kaust.edu.sa