### 3. FRAME COMBINING

Errors are bursty and independent

![Graph](image)

- Conventional link-layer ACKs do not work
  - Final status known only to MRDC
- Two levels of ACKs are high overhead
- Cannot disable link-layer ACKs

#### Flawed retransmission schemes

- How to correct simultaneous frame errors?
  - Frame combining
- How to handle retransmissions in MRD?
  - Request-for-acknowledgment protocol
- How to adapt bit rates in MRD?
  - MRD-aware rate adaptation

#### Challenges in developing MRD

1. Synchronous ACK
2. Soft selection
3. Synchronous ACK loss
4. Frame combining

![Diagram](image)

#### Bit-by-bit frame combining

- Recall: bit errors occur in bursts
- Divide frame into $N_b$ blocks (e.g., $N_b = 6$)
- Attempt recombination with all possible block patterns until CRC passes
  - # of checks upper bounded by $2^{N_b}$
  - Failure rate increases with $N_b$

#### Block-based frame combining

- Correlation and bursty characteristics
- Bit error rates range from $0.1$ to $10^{-5}$
- Minimal correlation is achieved with $L_b = 150$

![Diagram](image)

**Problem:** Exponential # of CRC checks in # of mismatched bits.
### Block based frame combining

- Transmit:
- Checksum:
- R1:
- R2:

$2^\Delta$ combinations of $\Delta$ differing blocks

$\Delta < N_b$

Main issue: what is probability of failure: $p_f$?

### Balls and bins analogy

- Throw $d_r$ identical red, $d_g$ identical green balls in $N_b$ (distinguishable) bins at random
- Place red ball in bin $b$: bit error in block $b$ of $AP_1$
- Place green ball in bin $b$: bit error in block $b$ of $AP_2$
- Frame combining error $c$: ???

### Stars and bars analogy

- How many ways to throw $d$ identical balls into $N_b$ (distinguishable) bins?

- As many as there are star-bar strings!

$$
\binom{d + N_b - 1}{N_b - 1} = \binom{d + N_b - 1}{d}
$$

### Conditional probability of combining error

- Suppose we have $d_r$ errors in received transmission at $AP_1$.
- $d_g$ errors in received transmission at $AP_2$.

- Choose a bin to have one green, one red

- Probability of failure given $d_r$ errors at $AP_1$, $d_g$ errors at $AP_2$:

$$
p_f(d_r, d_g) = \frac{N_b + (d_r - 1) + 1}{N_b} \cdot \frac{N_b + (d_g - 1) + 1}{N_b}
$$

### Probability of frame combining failure

- $p_f = \sum_{d_r + d_g = d} p_f(d_r, d_g) Pr(d_r \text{ errors at } AP_1, d_g \text{ errors at } AP_2)$

- Graph: A. Alu, MBD 103
MRD-aware rate adaptation

- Standard rate adaptation does not work
  - Reacts only to link-layer losses from 1 receiver
  - Uses sub-optimal bit-rates
- MRD-aware rate adaptation
  - Reacts to losses at the MRD-layer

**Implication:** First use multiple paths, then adapt bit rates.

---

**Experimental setup (“HIVAR”)**

- 802.11a/b/g implementation in Linux (MADWiFi)
- L transmits 100,000 1,500 byte UDP packets w/7 retries
- 802.11a @ auto bit rate [6, 9, 12, 18, 24, 36, 48, 54]
- L is in motion at walking speed, > 1 minute per trial
- Variants: R1, R2, MRD (5 trials each)

---

**MRD throughput improvement**

- 18.7 Mbps, 2.3x Improvement, 8.25 Mbps

---

**MRD maintains high bit rate**

---

**Closer look at frame recovery method**

- Frame recovery rate = 49%
Would increasing $N_B$ help?

Best case for frame combining

Frame recovery rate = 59%

Latency distribution

ExOR: a new approach to routing in multi-hop wireless networks

The big wireless picture

Initial approach: Traditional routing

- Influence on later work
- Real implementation
- Evaluation

- Identify a route, forward over links
- Abstract radio to look like a wired link
Radios aren’t wires

- Every packet is broadcast
- Reception is probabilistic

ExOR: exploiting probabilistic broadcast

- Decide who forwards after reception
- Goal: only closest receiver should forward
- Challenge: agree efficiently and avoid duplicate xmits

Outline

- Introduction
  - Why ExOR might increase throughput
  - ExOR protocol
  - Measurements
  - Related Work

Why ExOR might increase throughput (1)

- Best traditional route over 50% hops: 3(1/0.63) = 6 tx
- Throughput = 1/\text{transmissions}
- ExOR exploits lucky long receptions
- ExOR recovers unlucky short receptions

Why ExOR might increase throughput (2)

- Traditional routing: 1/0.25 + 1 = 5 tx
- ExOR: 1/(1 - (1 - 0.25)/4) + 1 = 2.5 transmissions
- Assumes independent losses

Outline

- Introduction
  - Why ExOR might increase throughput
  - ExOR protocol
  - Measurements
  - Related Work
ExOR batching

- Challenge: finding the closest node to have rx'd
- Send batches of packets for efficiency
- Node closest to the dst sends first
  - Other nodes listen, send remaining packets in turn
- Repeat schedule until dst has whole batch