Maranello: Practical Partial Packet Recovery for 802.11

Bo Han
Aaron Schulman
Neil Spring
Bobby Bhattacharjee
University of Maryland

Francesco Gringoli
Lorenzo Nava
University of Brescia

Lusheng Ji
Seungjoon Lee
Robert Miller
AT&T Research

Block-based partial packet recovery

corrupt packet

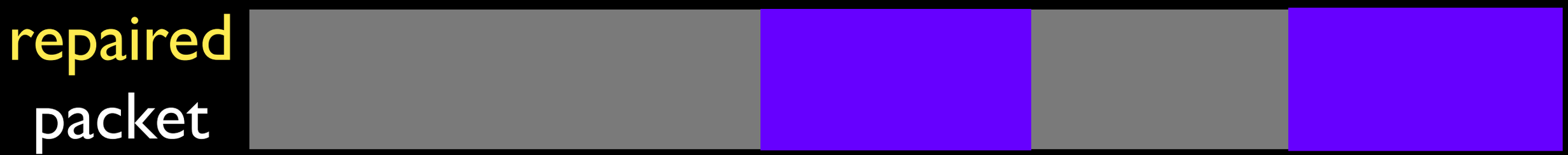
Block-based partial packet recovery

Block-based partial packet recovery

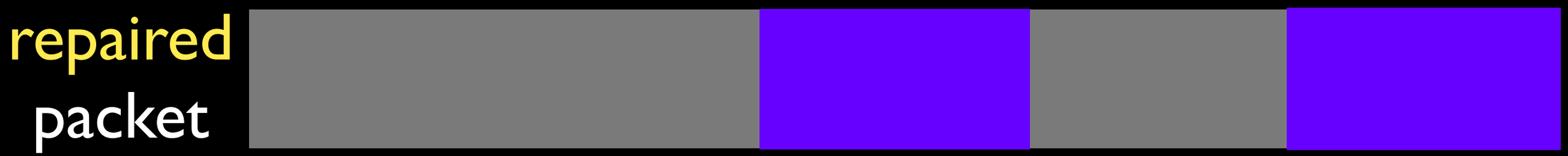
corrupt packet

Block-based partial packet recovery

Block-based partial packet recovery

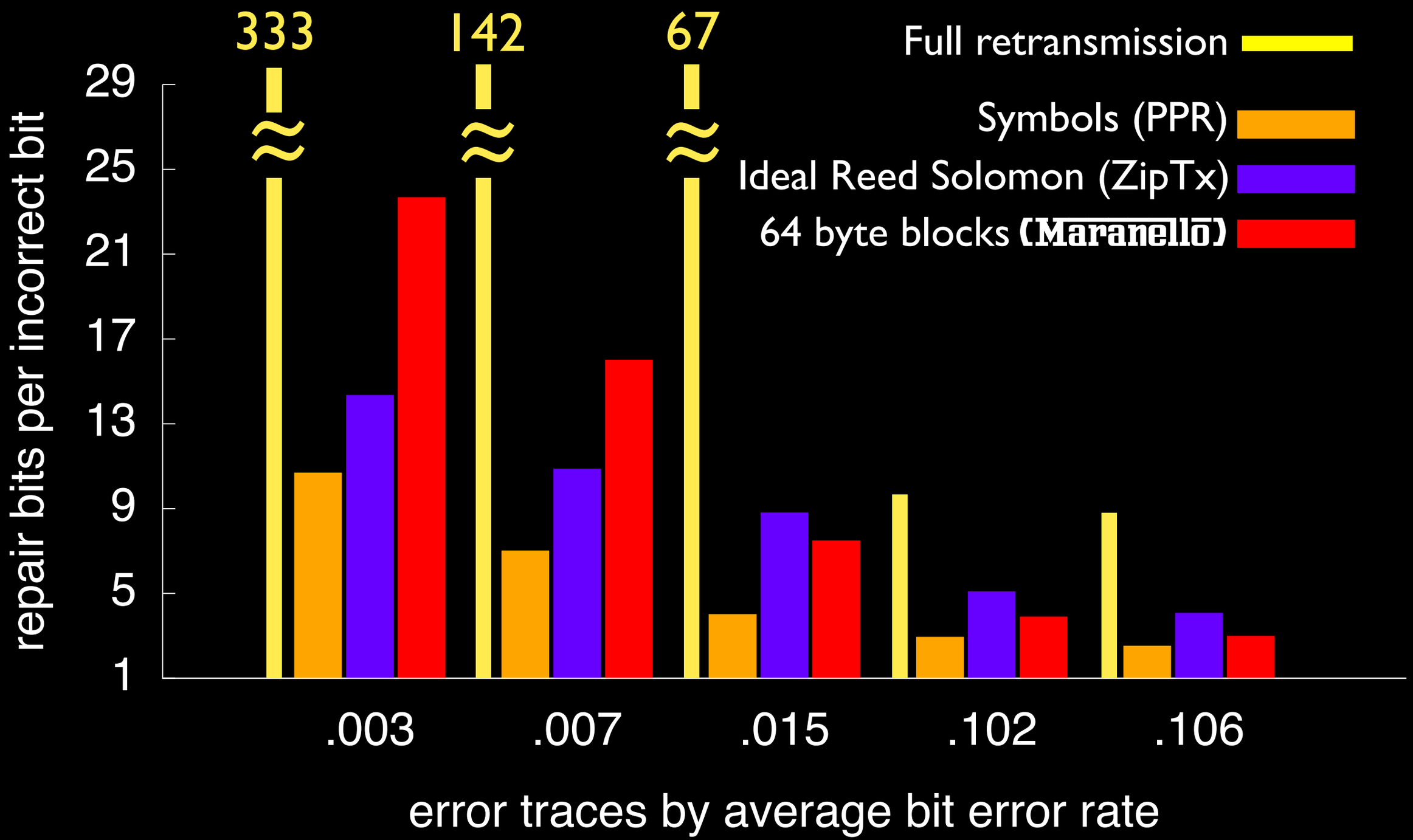


Block-based partial packet recovery

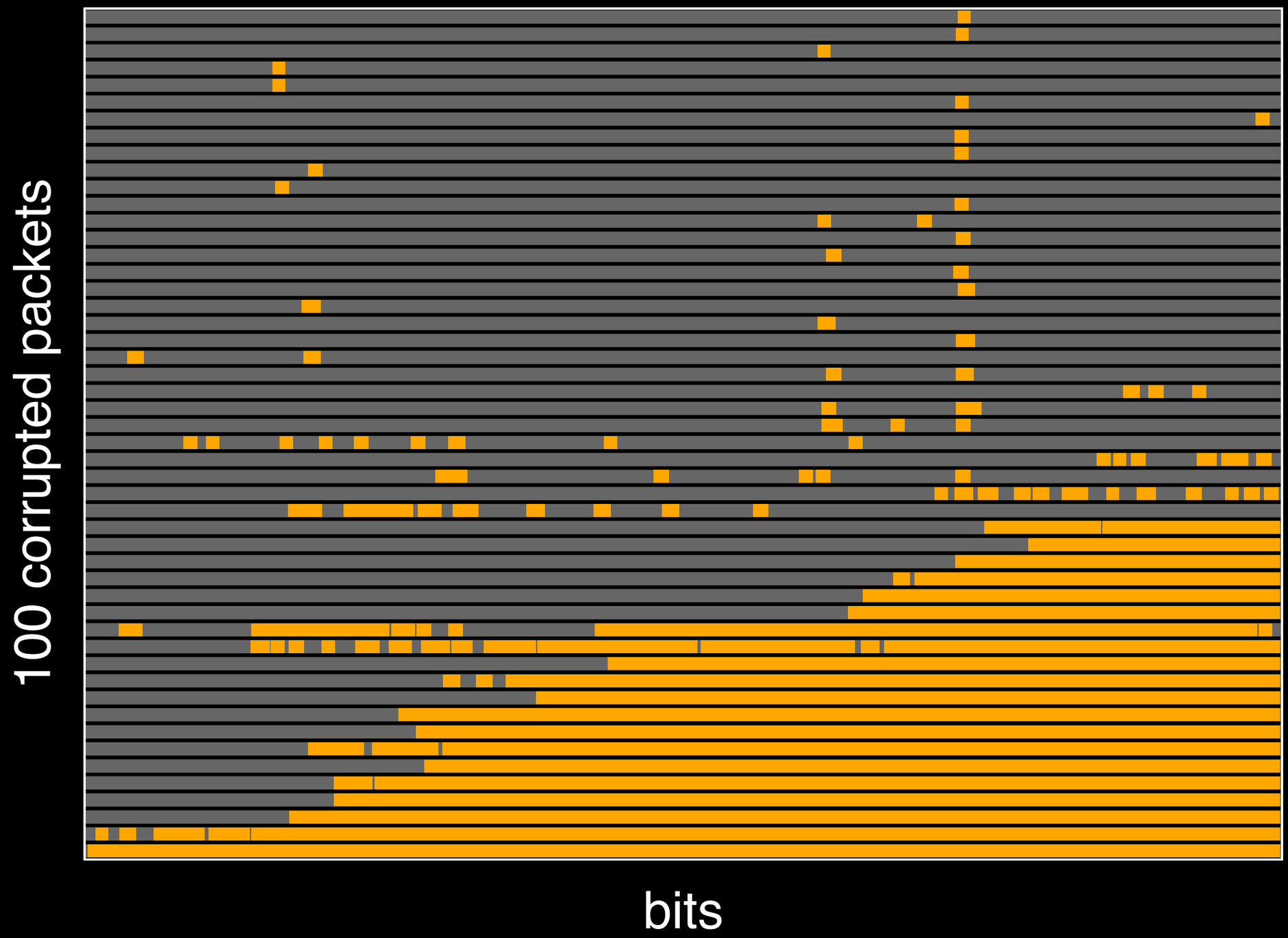


- Increases throughput because:
- Repairs are shorter than retransmissions
- Short transmissions have higher delivery probability
- Early delivery avoids backoff and low rate retransmission

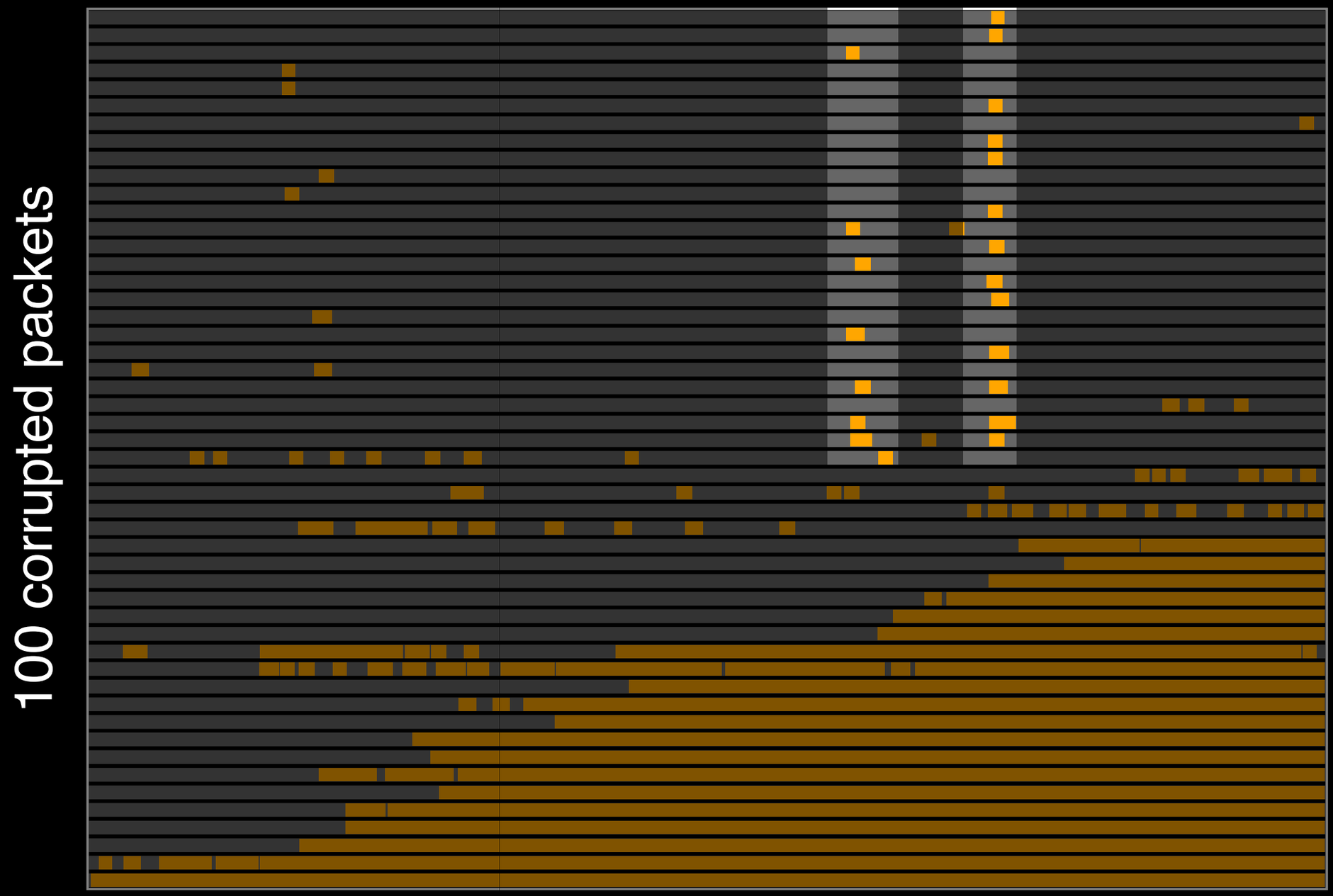
Deployable partial packet recovery



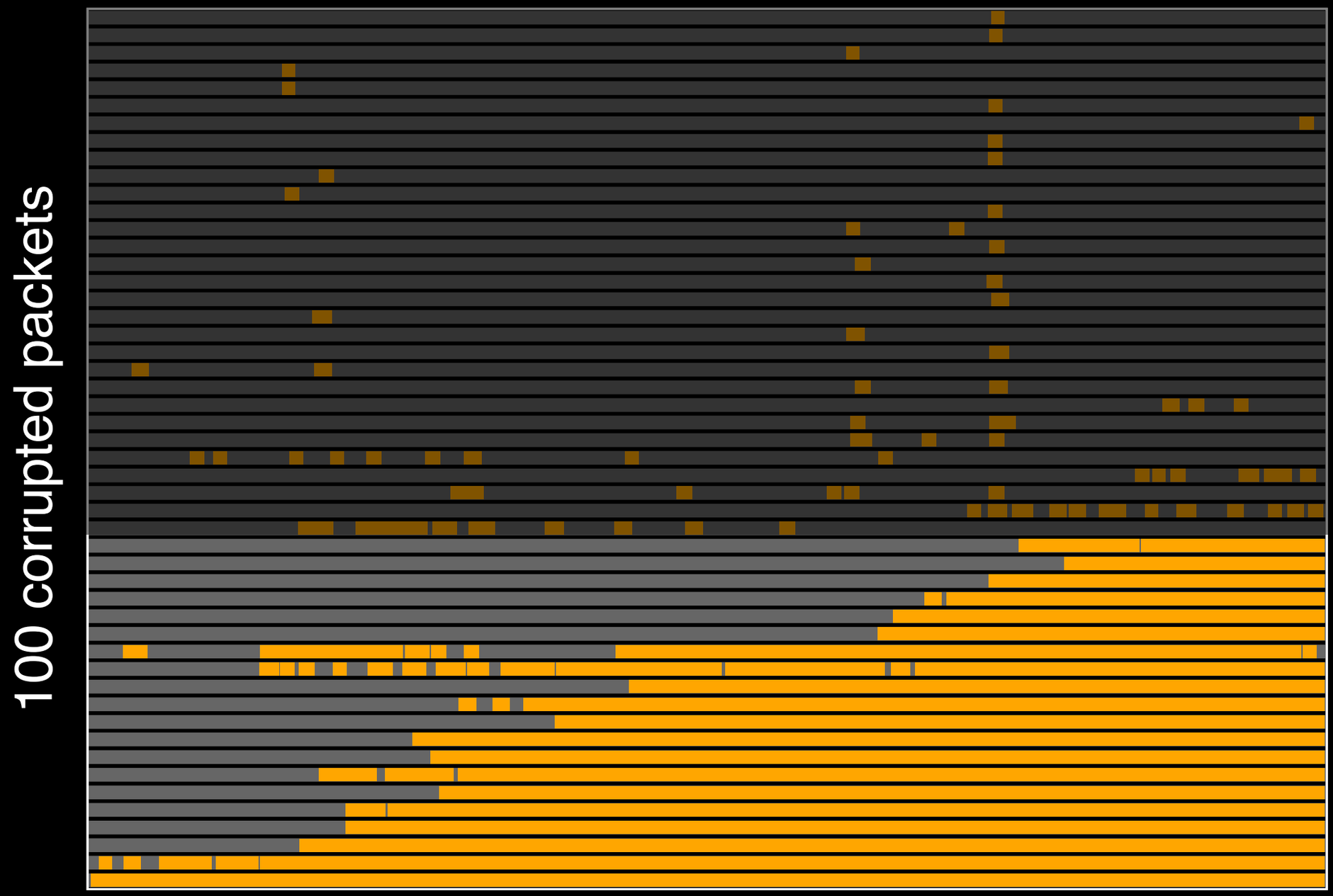
802.11 errors are clustered



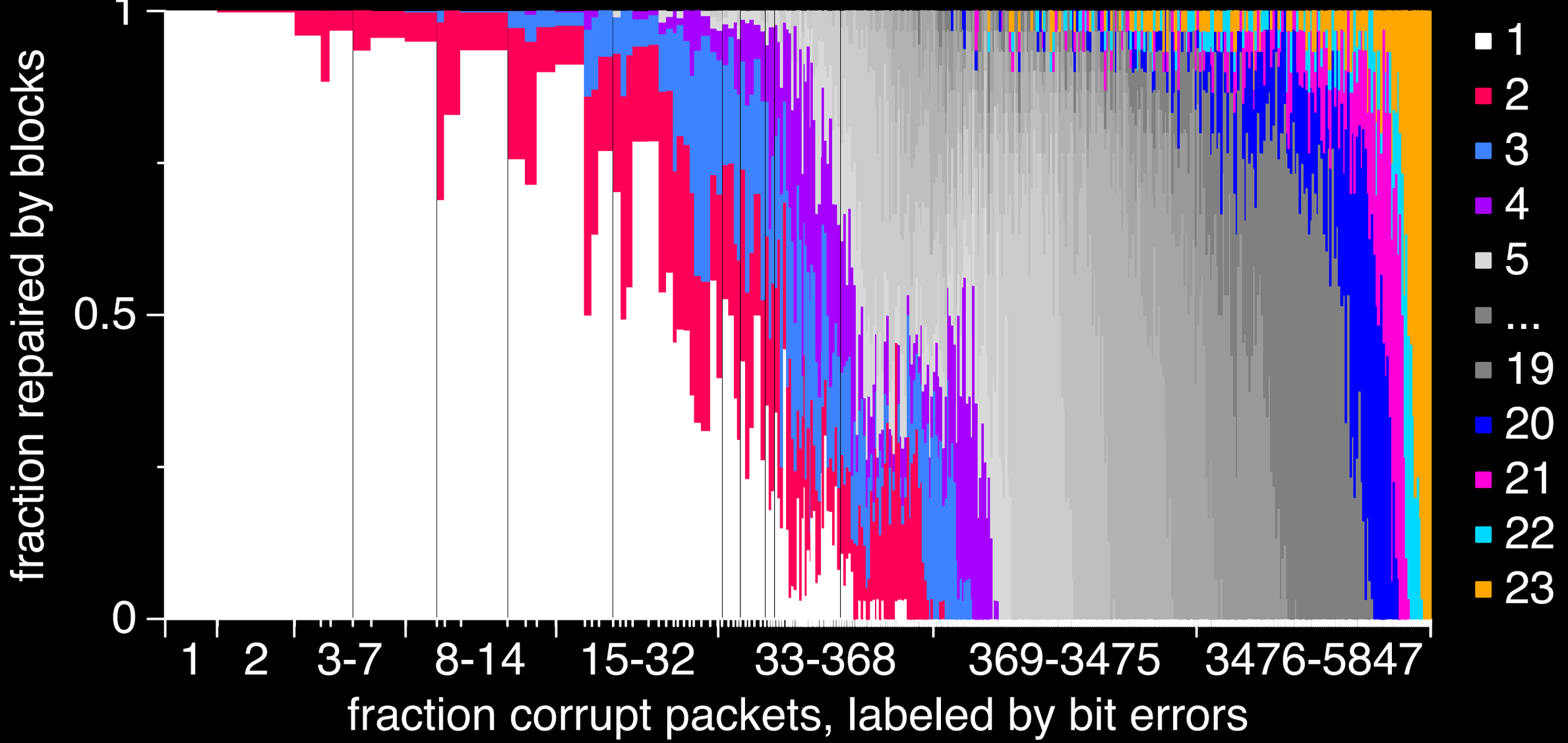
802.11 errors are clustered



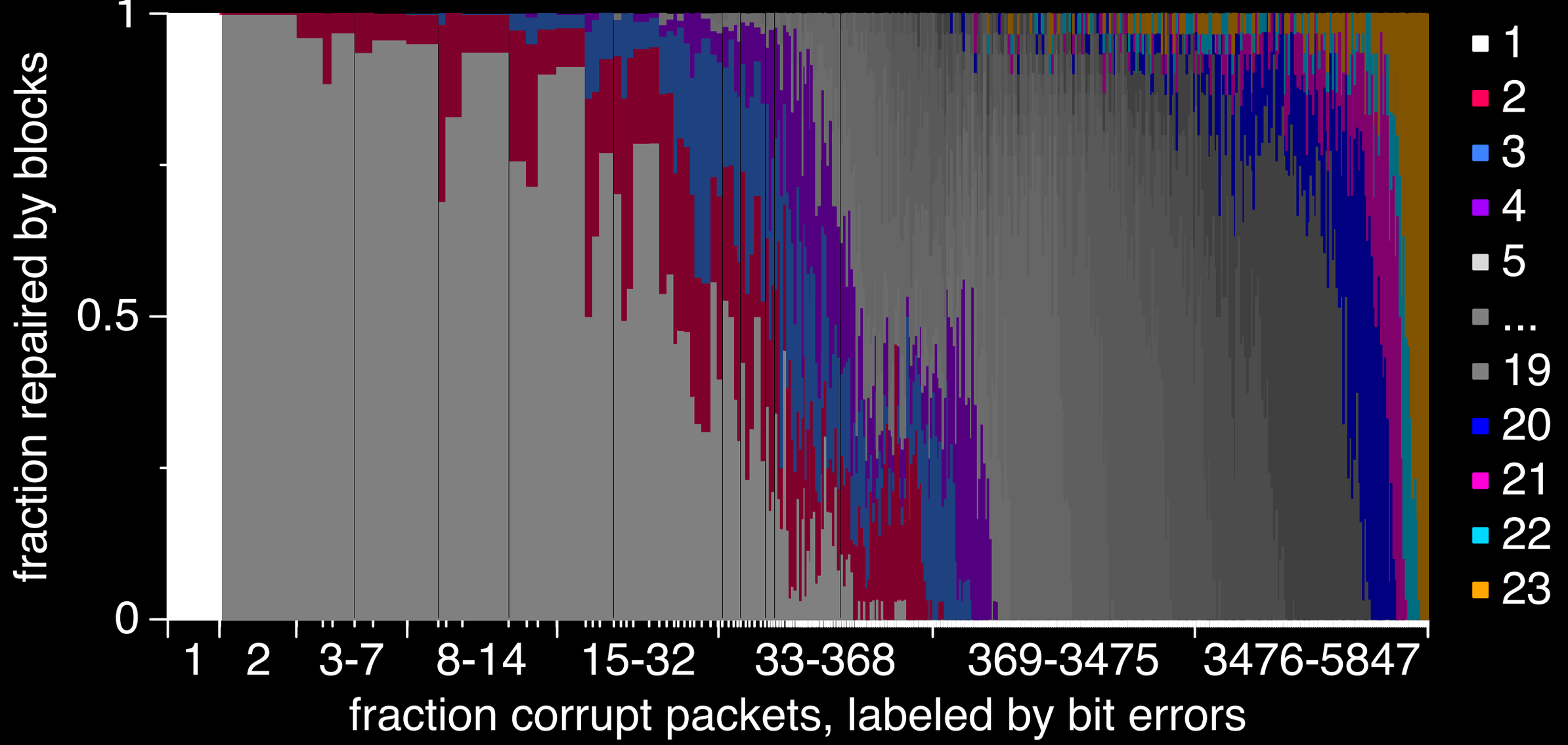
802.11 errors are clustered



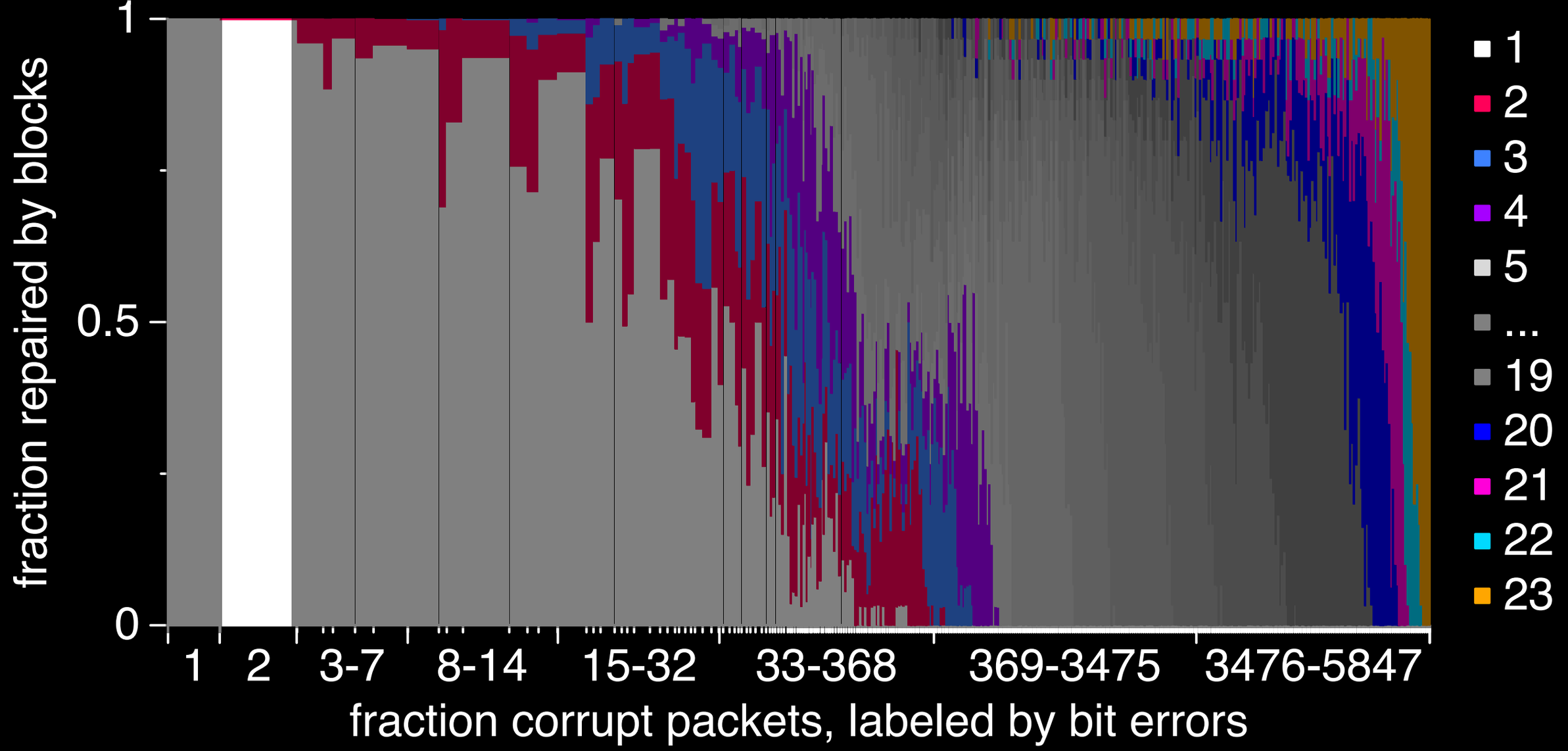
Need few blocks for repairs



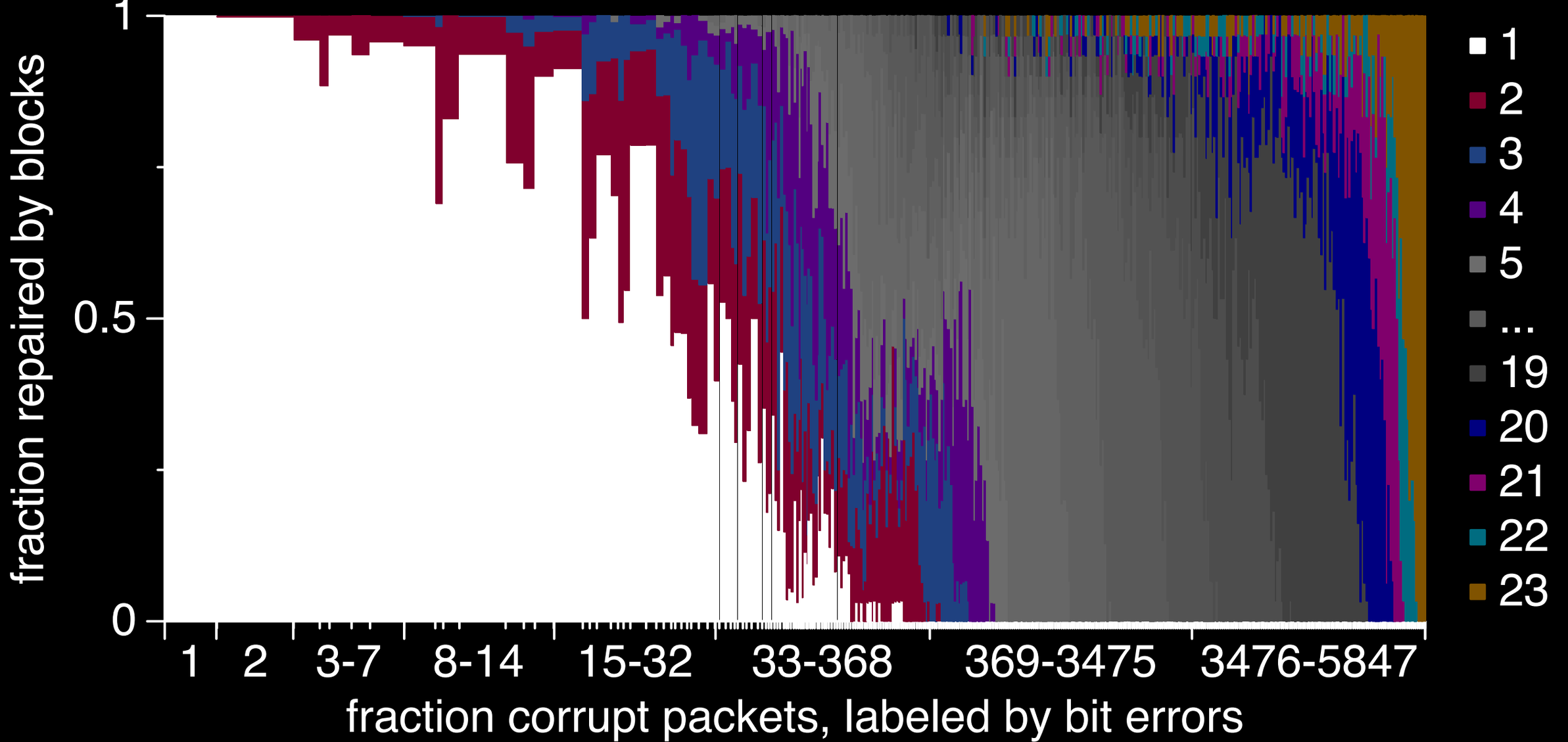
Need few blocks for repairs



Need few blocks for repairs

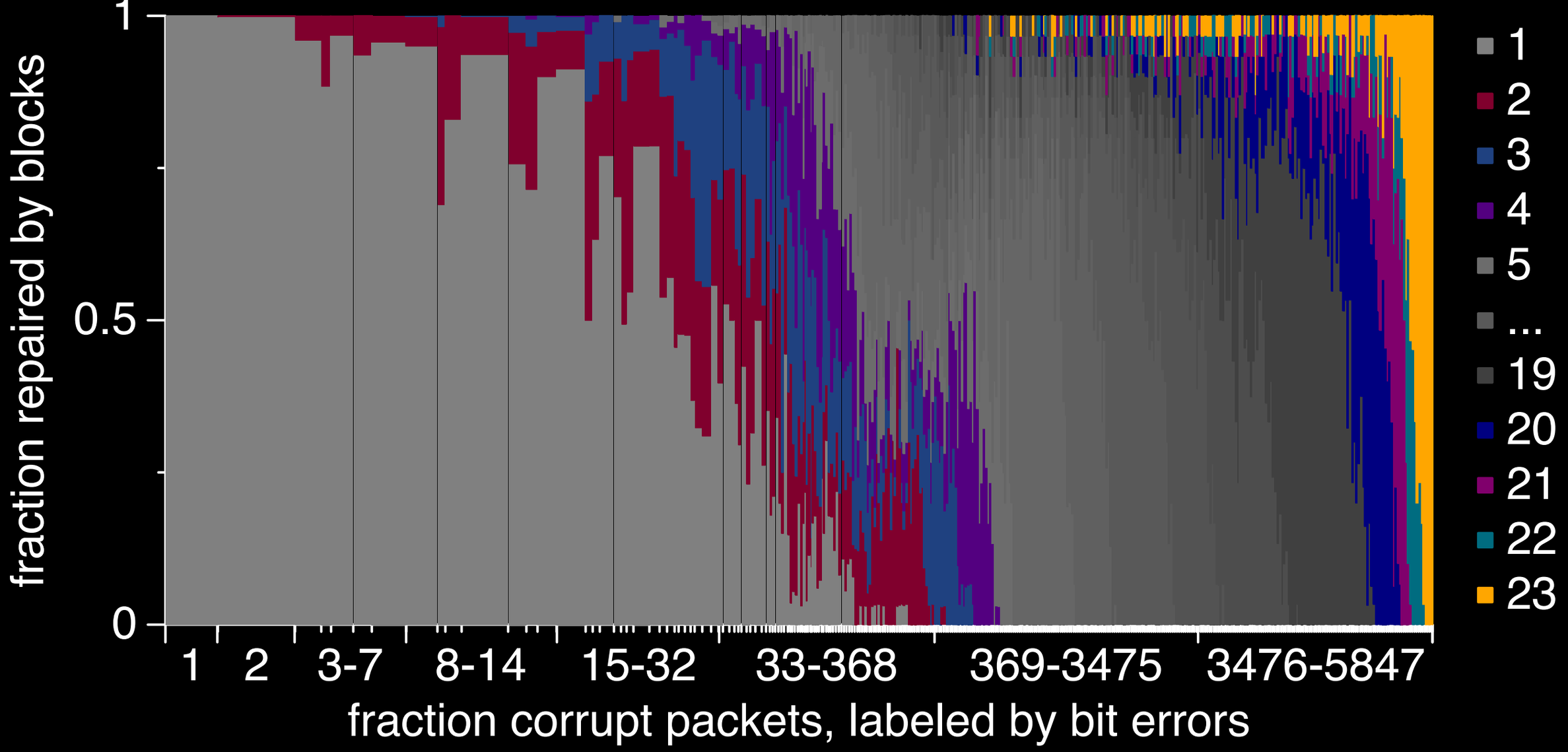


Need few blocks for repairs



≤32 bit errors,
repaired by one block

Need few blocks for repairs



≤32 bit errors,
repaired by one block

many bit errors,
still correct blocks

Maranello design goals

- Compatible with 802.11 (Maintain link latency)
- Incremental deployment
- Runs on existing hardware
- No extra bits for correct packets

Firmware suits block repair

GnuRadio
e.g. PPR

long delay
\+ expensive
\+ low rate

deploy completely
not compatible

Firmware
Maranello

short delay
\+ cheap
\+ high rate

deploy incrementally
802.11 compatible

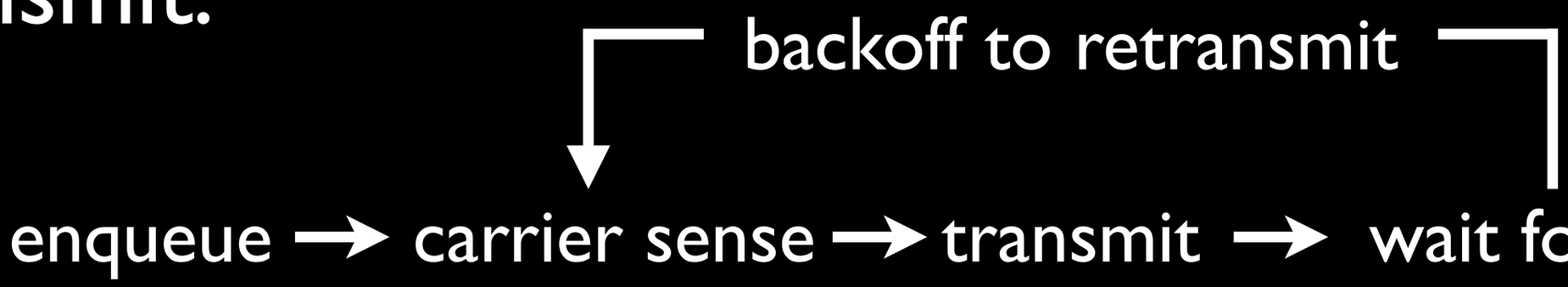
Driver
e.g. ZipTx

long delay
\+ cheap
\+ high rate

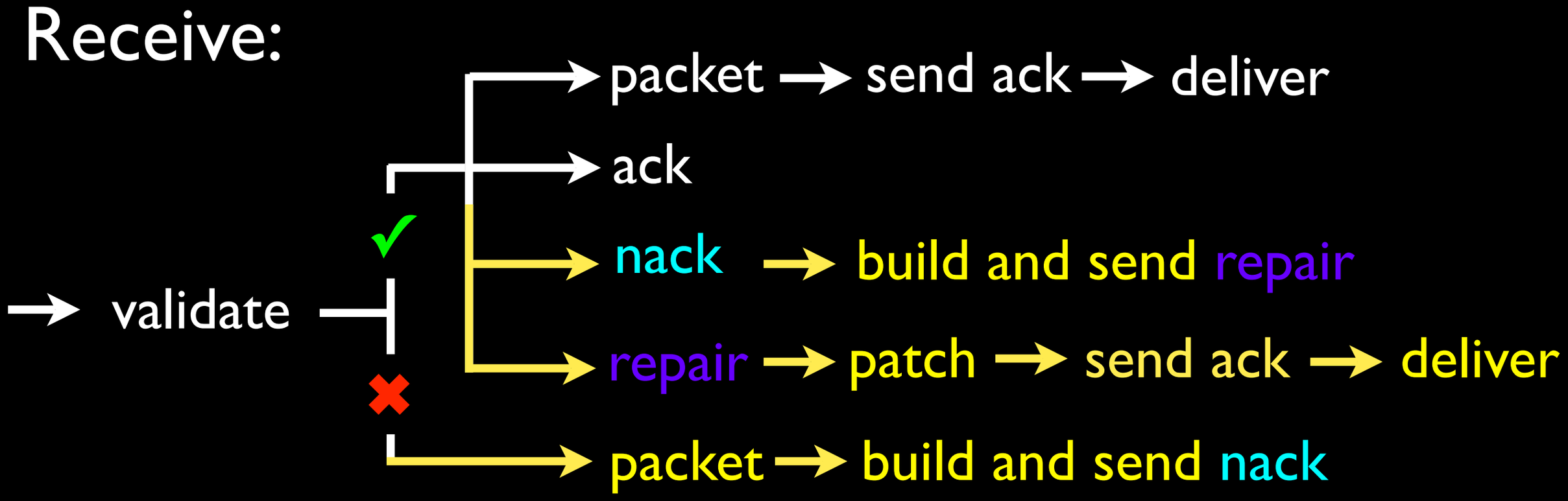
deploy incrementally
not compatible

Broadcom (OpenFWWF) firmware

Transmit:

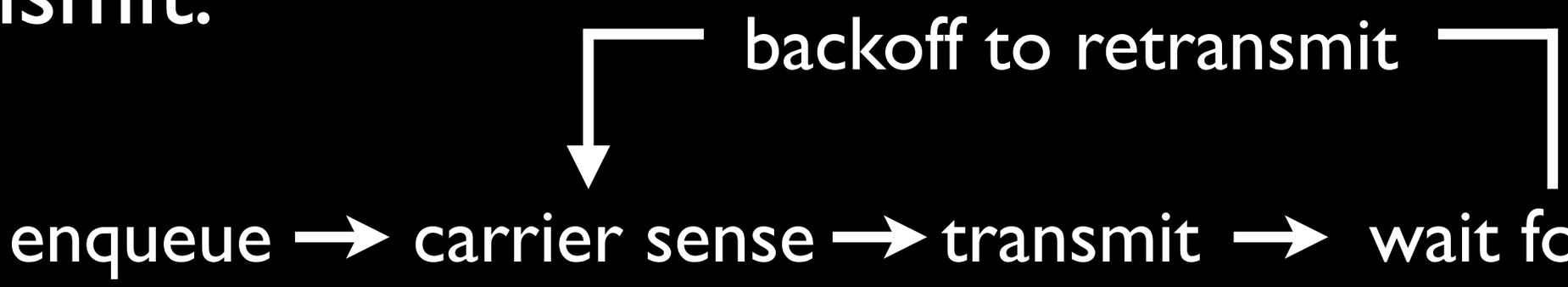


Receive:

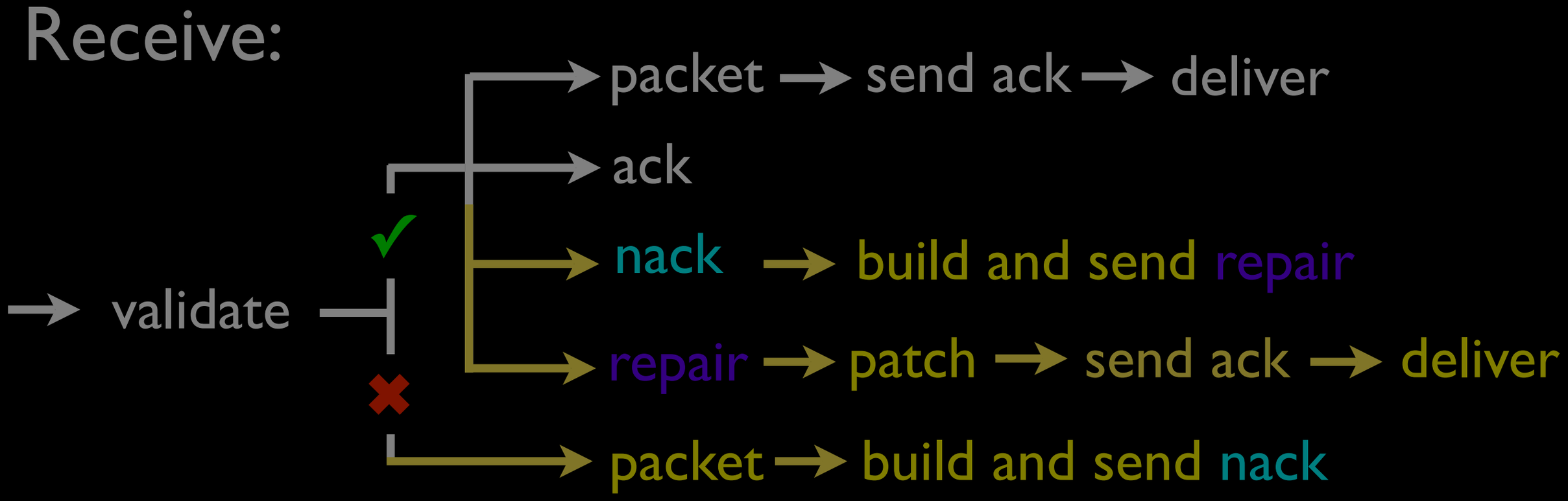


Broadcom (OpenFWWF) firmware

Transmit:

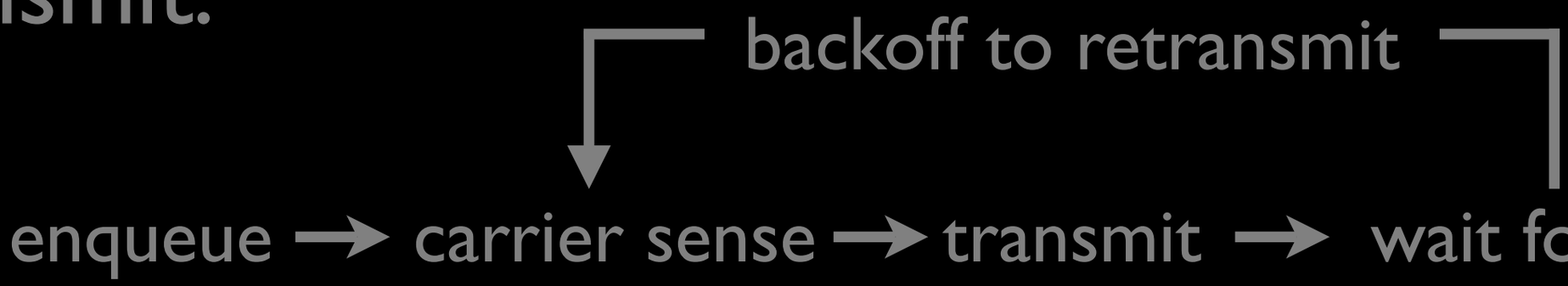


Receive:

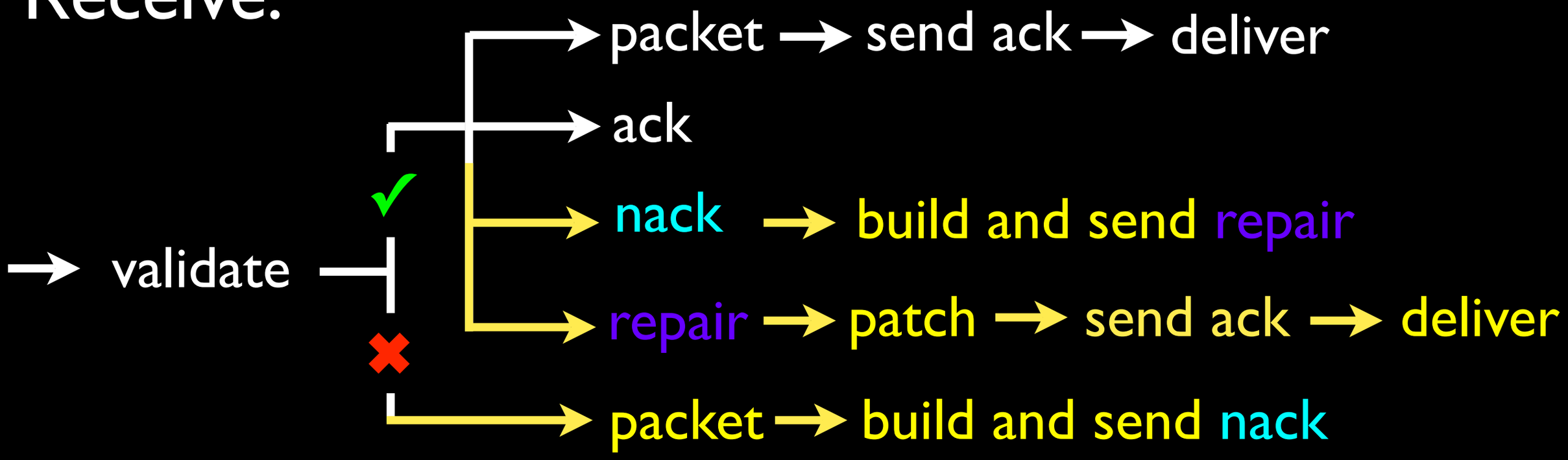


Broadcom (OpenFWWF) firmware

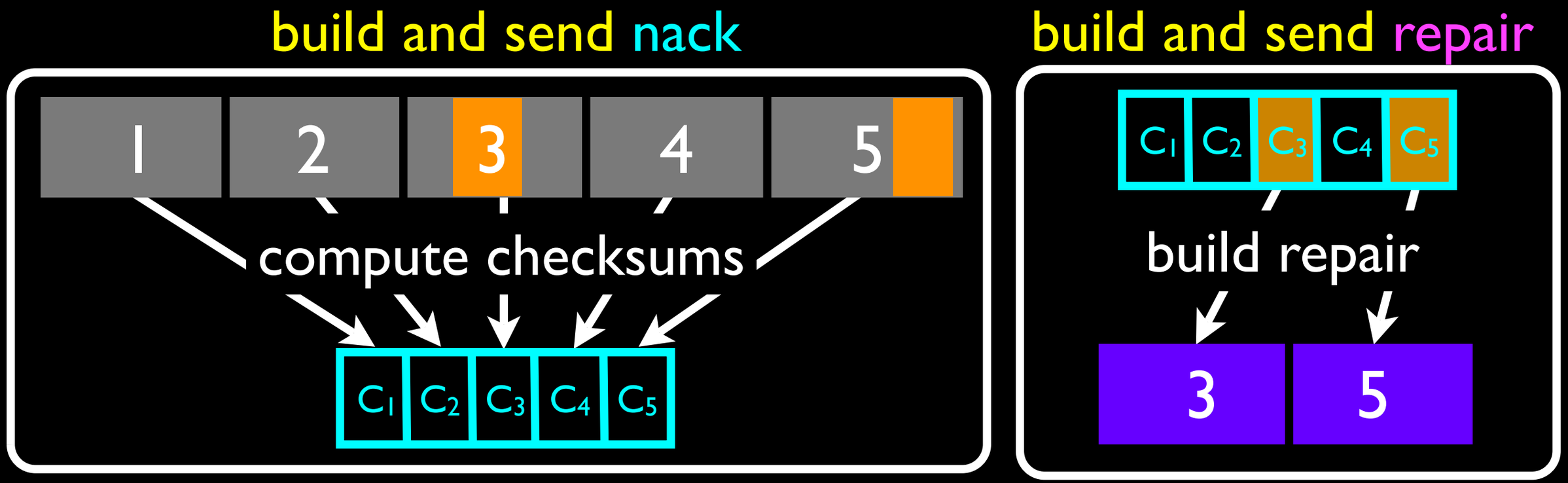
Transmit:

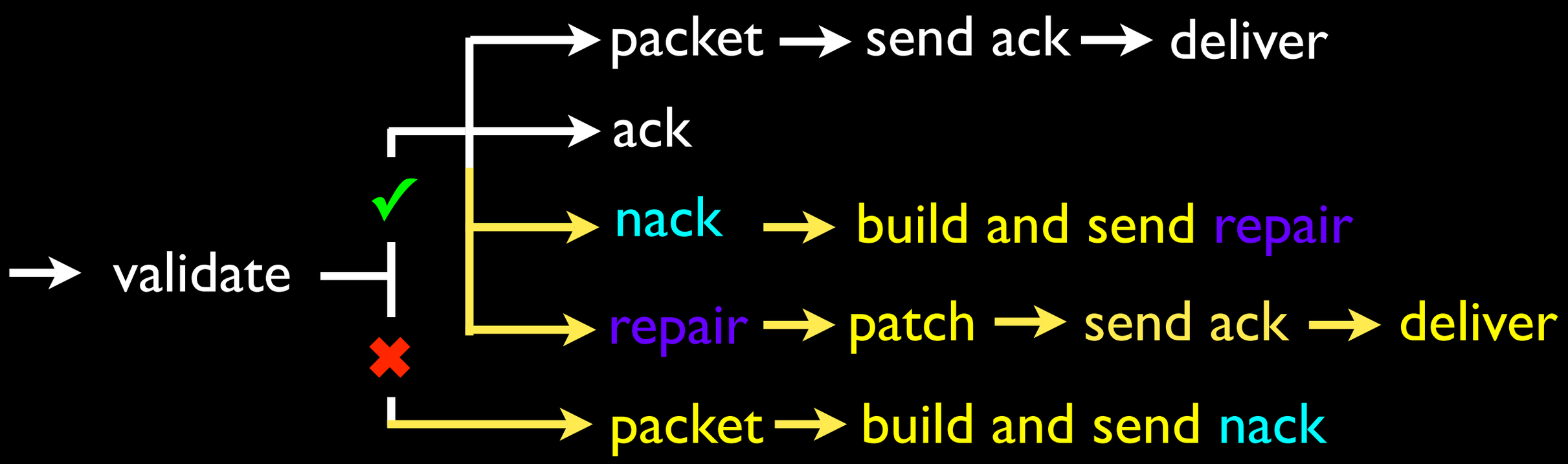


Receive:

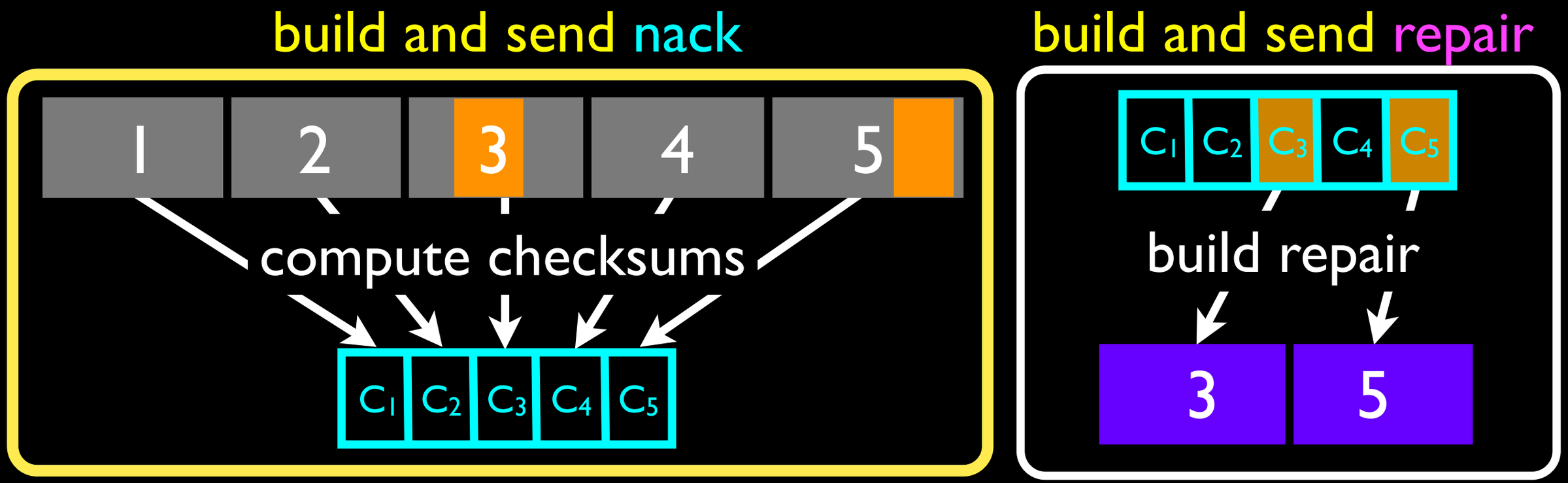


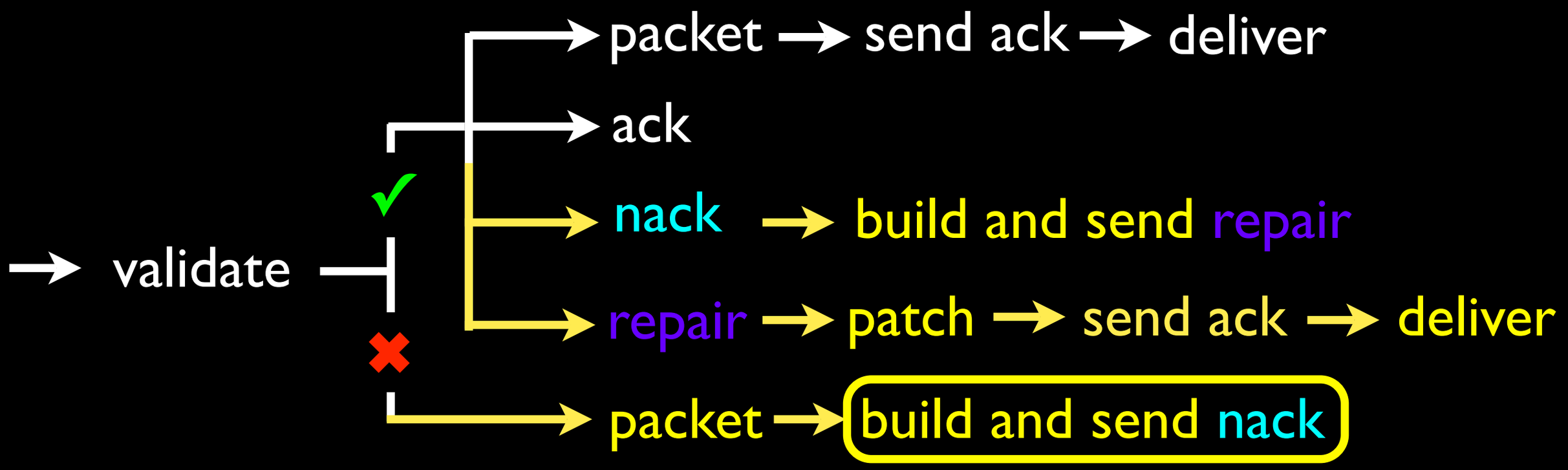
Nacking with checksums





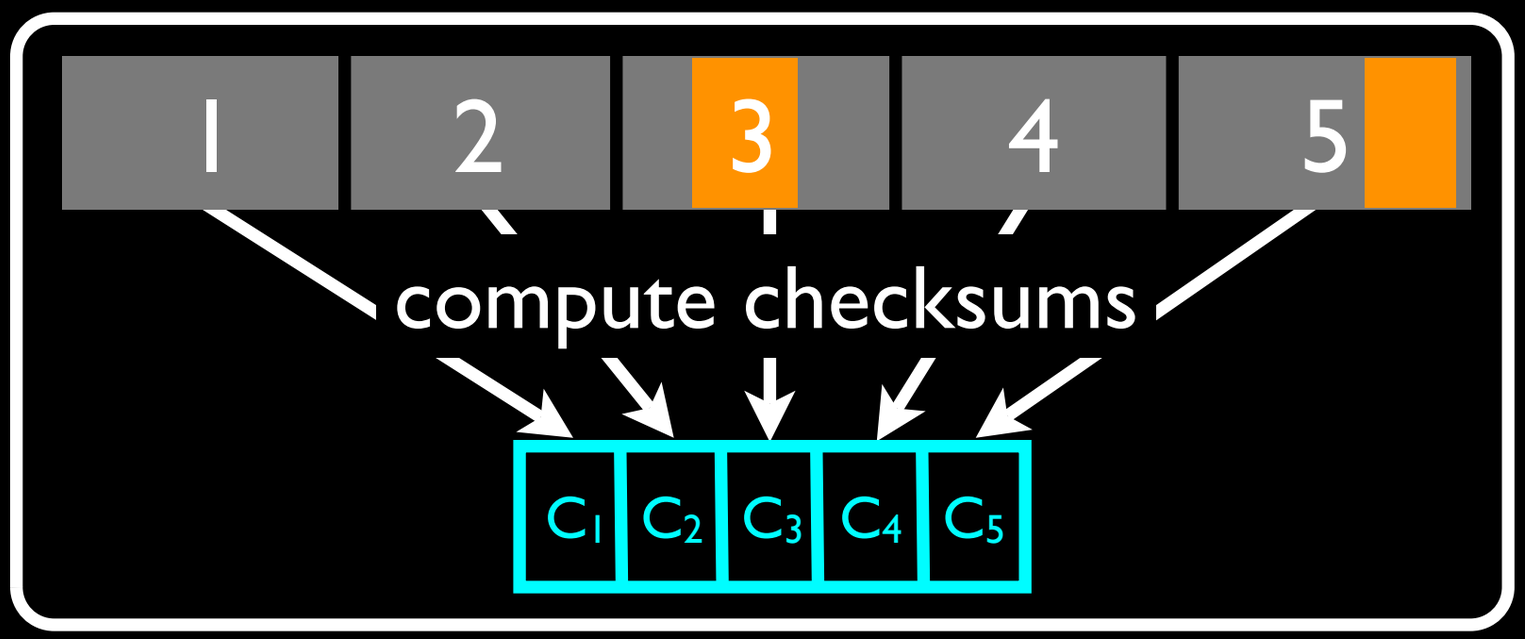
Nacking with checksums



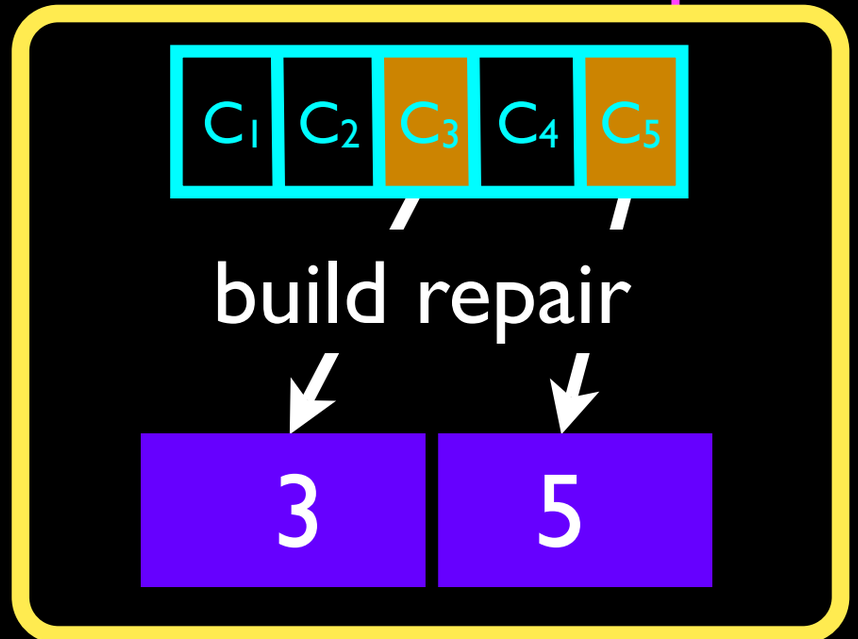


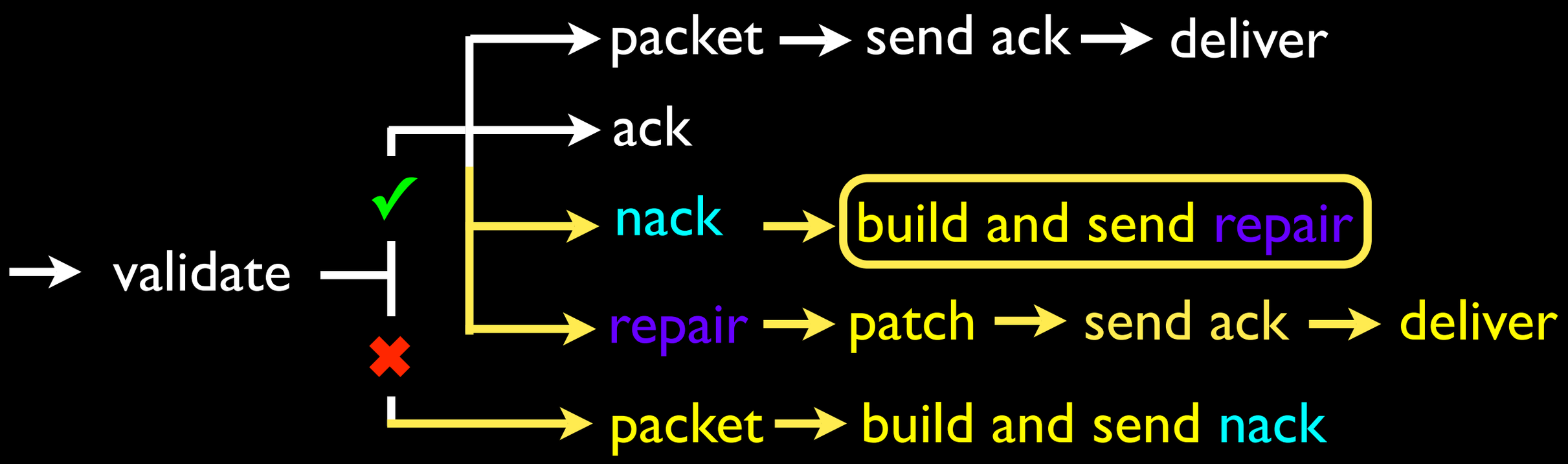
Nacking with checksums

build and send nack



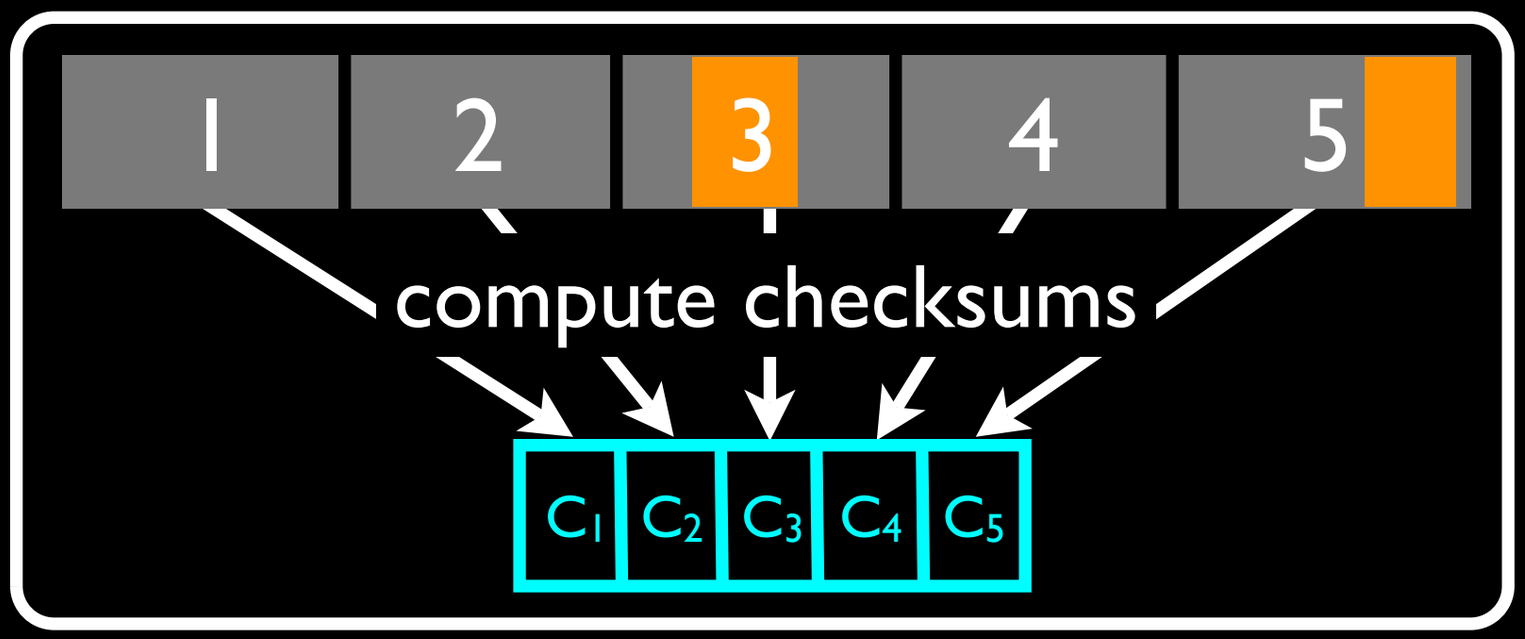
build and send repair



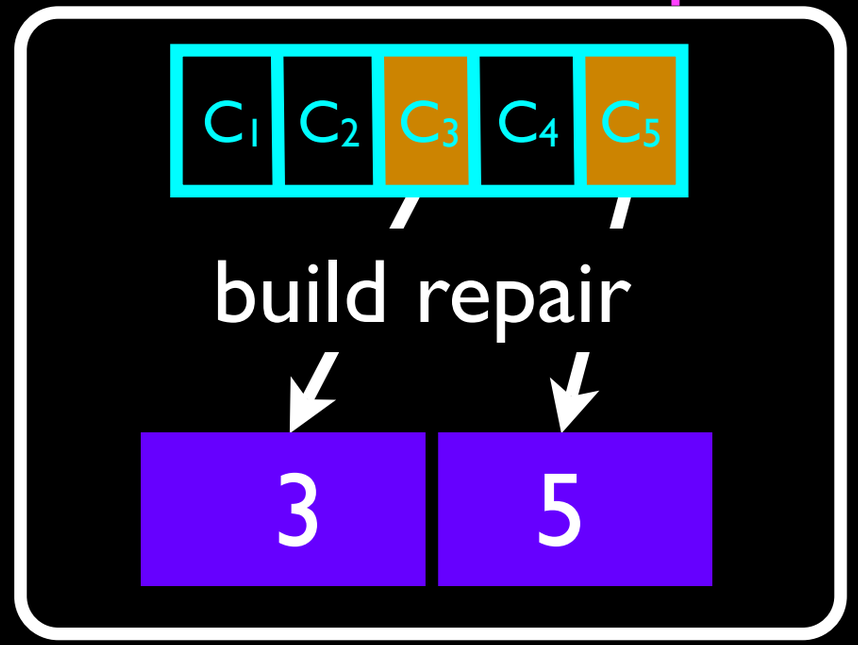


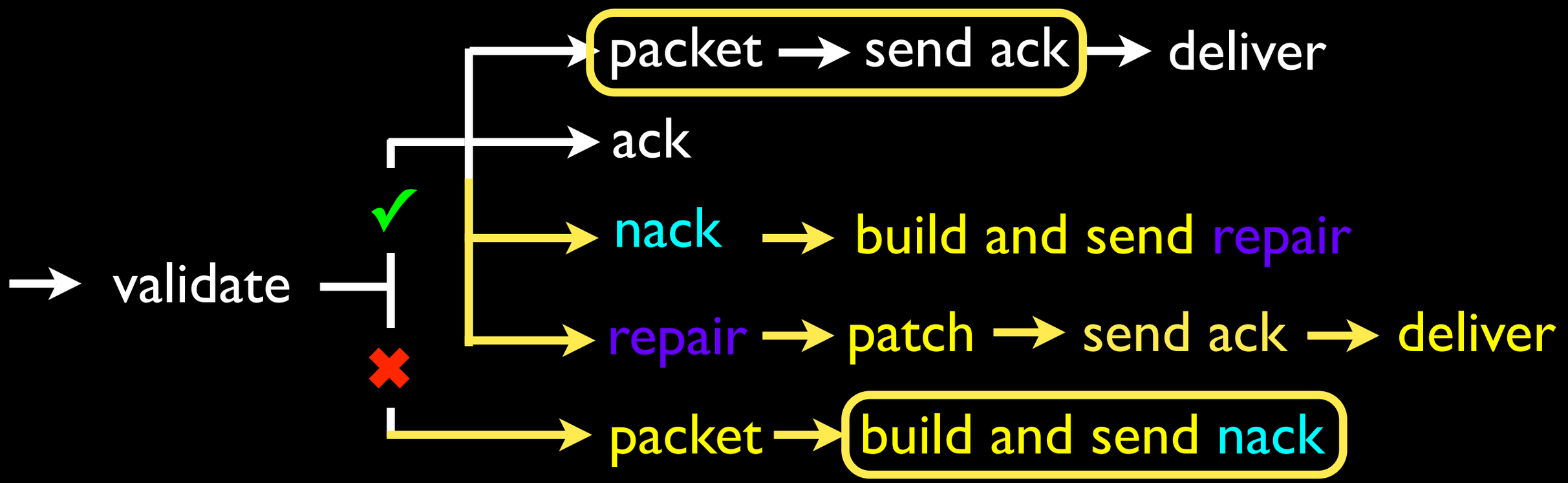
Nacking with checksums

build and send nack



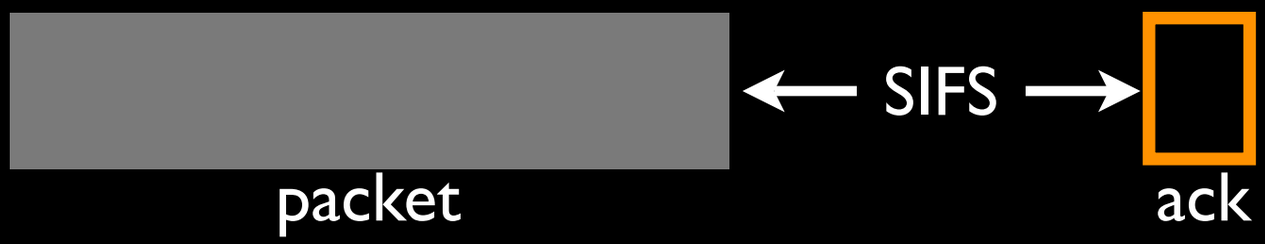
build and send repair



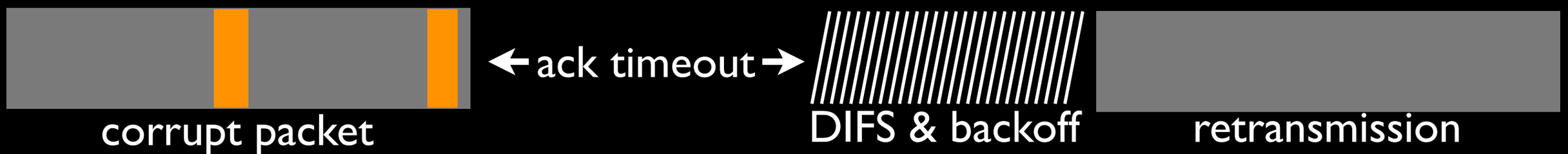


802.11's timing

Correct:

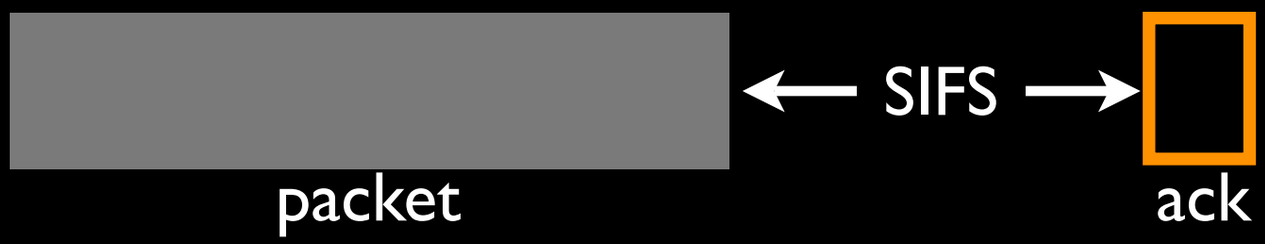


Retransmission:

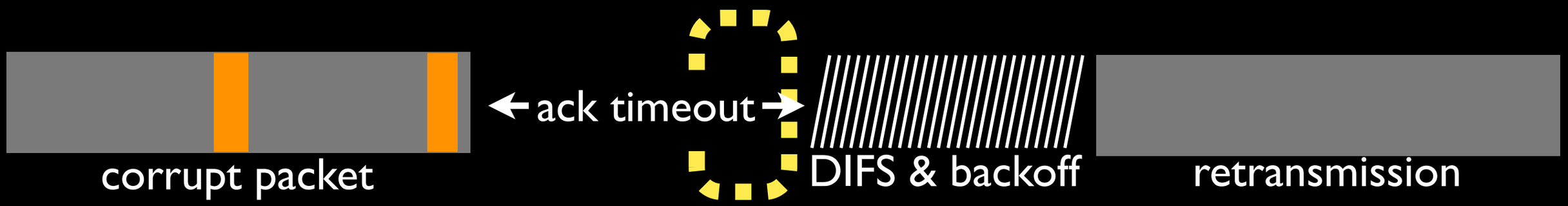


802.11's timing

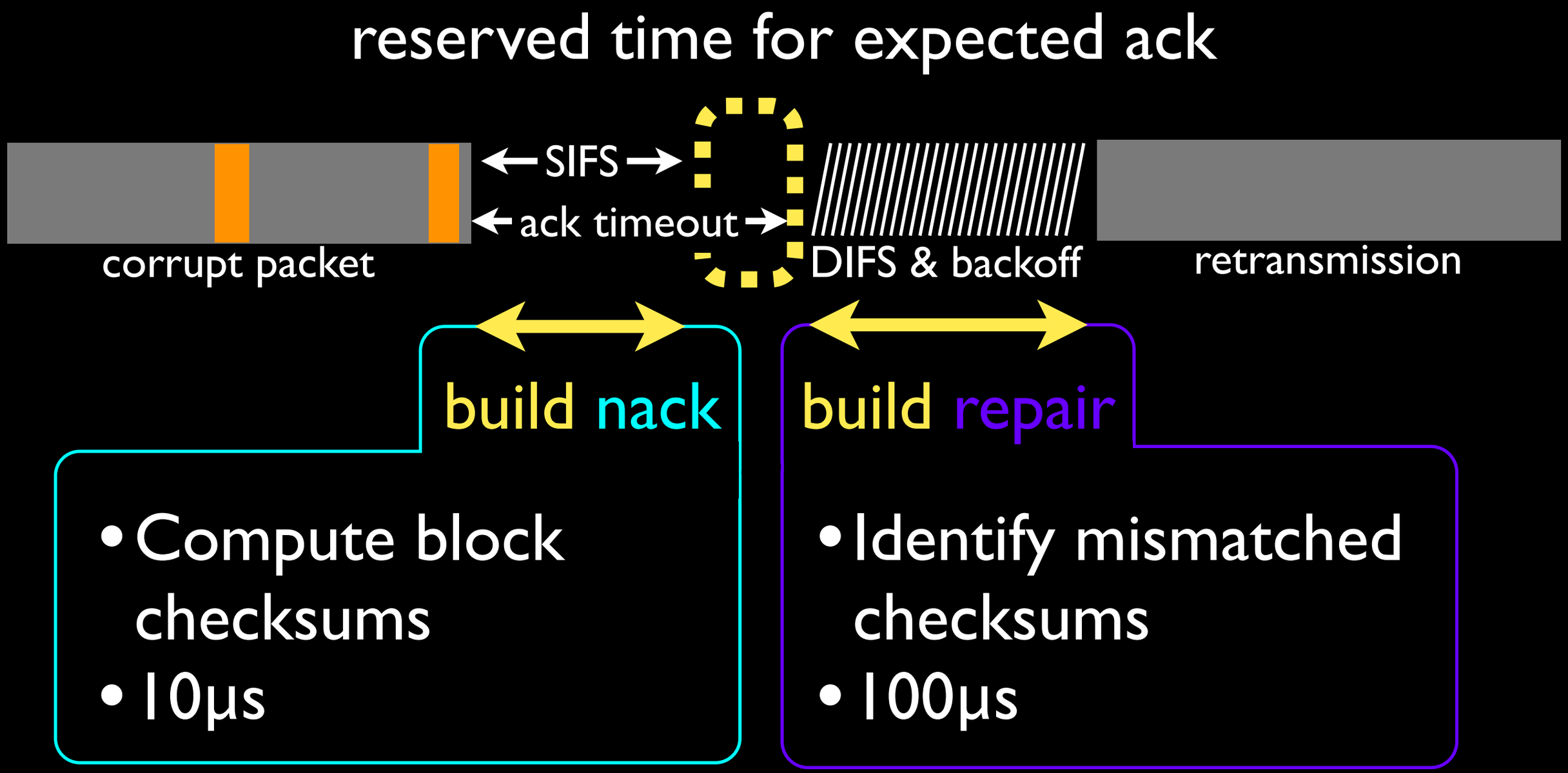
Correct:



Retransmission:



Exploiting 802.11's timing



Fletcher-32

- Operations per 16 bits: two adds, two decrements
- All single bit errors, burst errors in a single 16-bit block, and two-bit errors separated by at most 16 bits
- CRC-32 & Fletcher-32 find all block errors in 9,911,800 802.11 error packets
- **Maranello** relies on whole packet CRC-32

Card's CPU is not fast enough

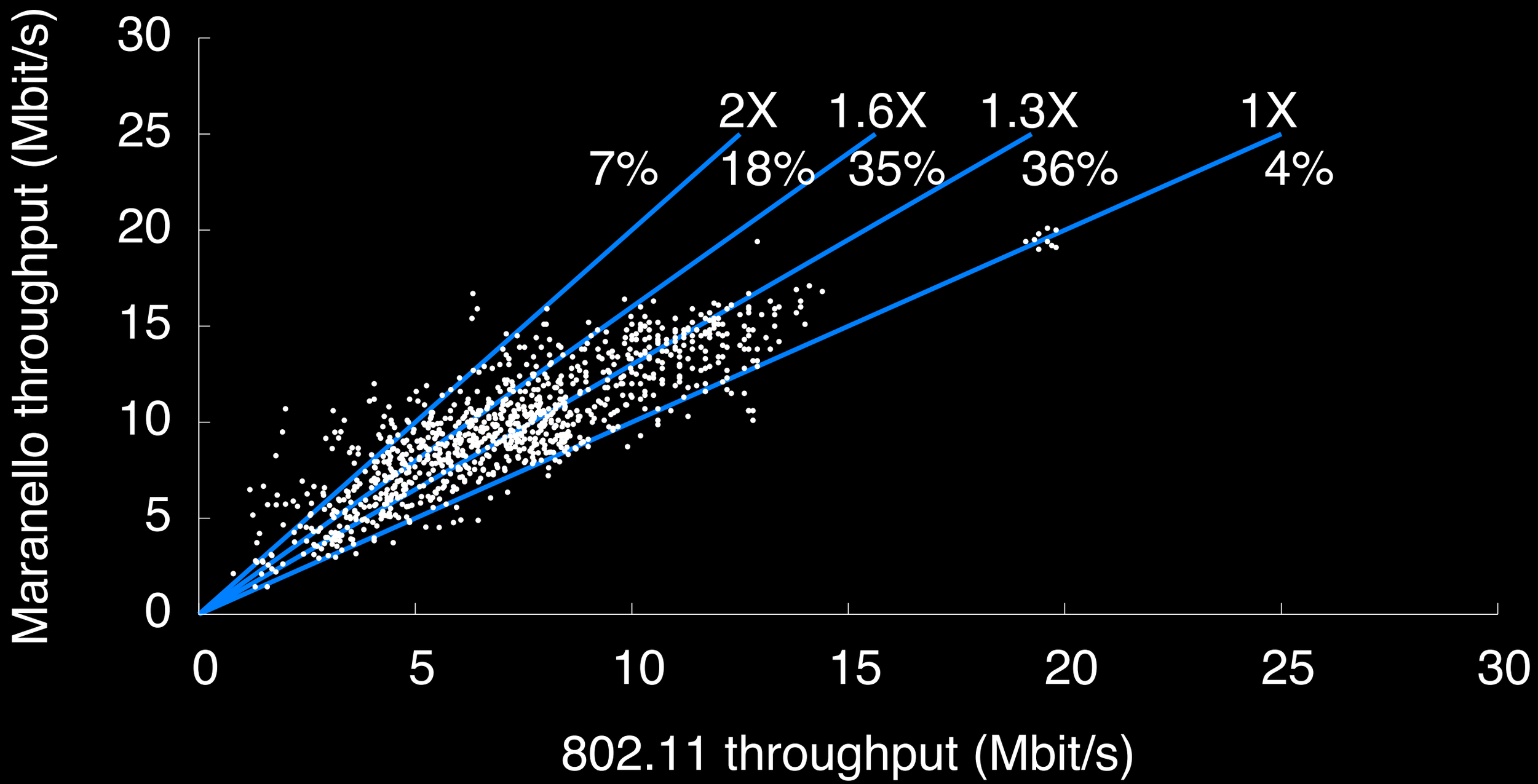
- Problem: 4µs per 64 byte block (SIFS = 10µs)

- Insight: Compute block checksums while blocks are received

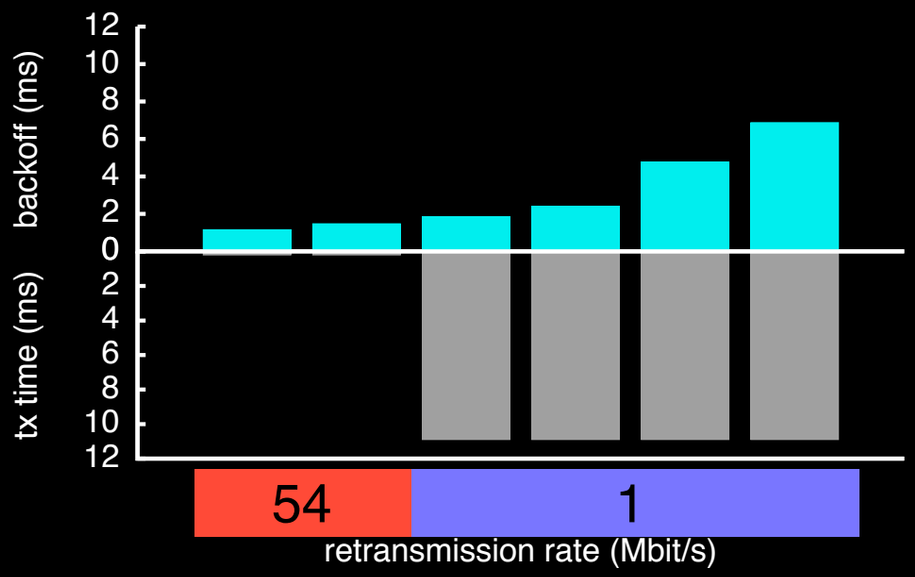
Maranello design review

Goal	Solution
Compatible with 802.11 (Maintain link layer latency)	• Broadcom firmware • Replace ack with nack (Fletcher-32 while blocks are received)
Incremental deployment	Legacy 802.11 ignores nacks
No extra bits for correct packets	Receiver sends checksums, sender finds the errors

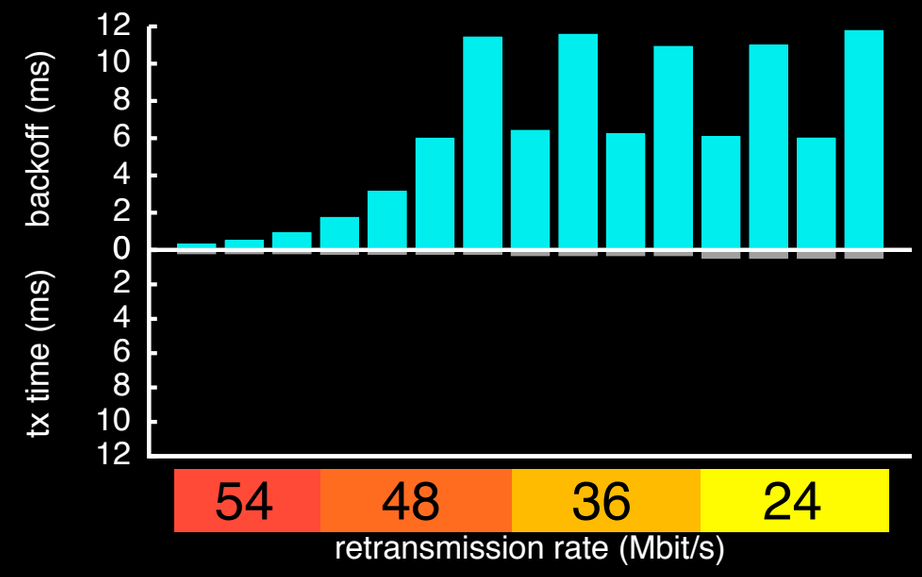
Maranello is faster than 802.11



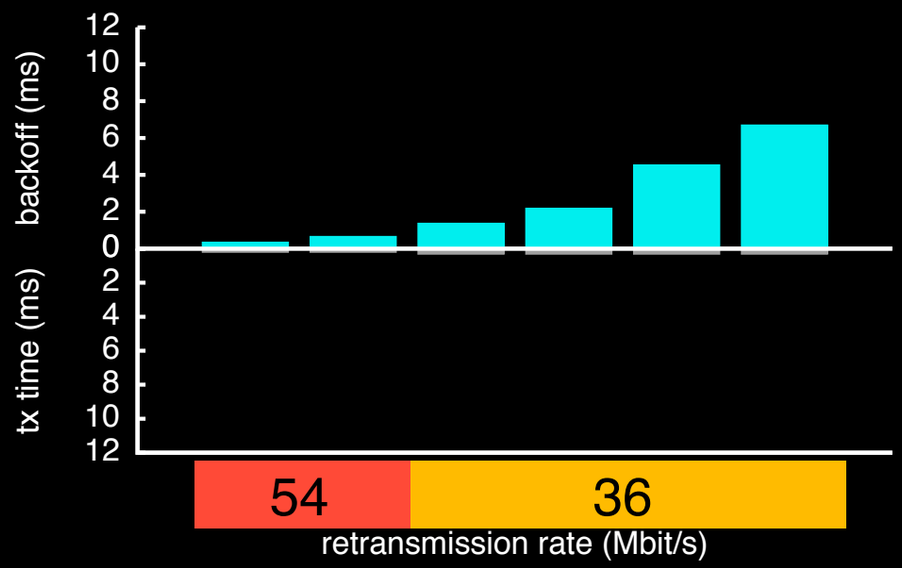
Retransmission behavior varies



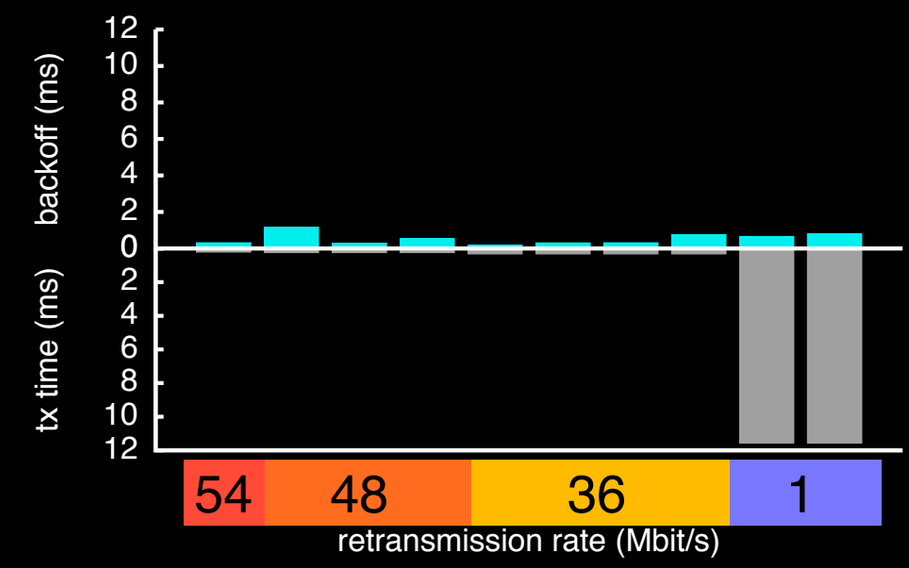
Linux ‘minstrel’ Broadcom



Windows Intel

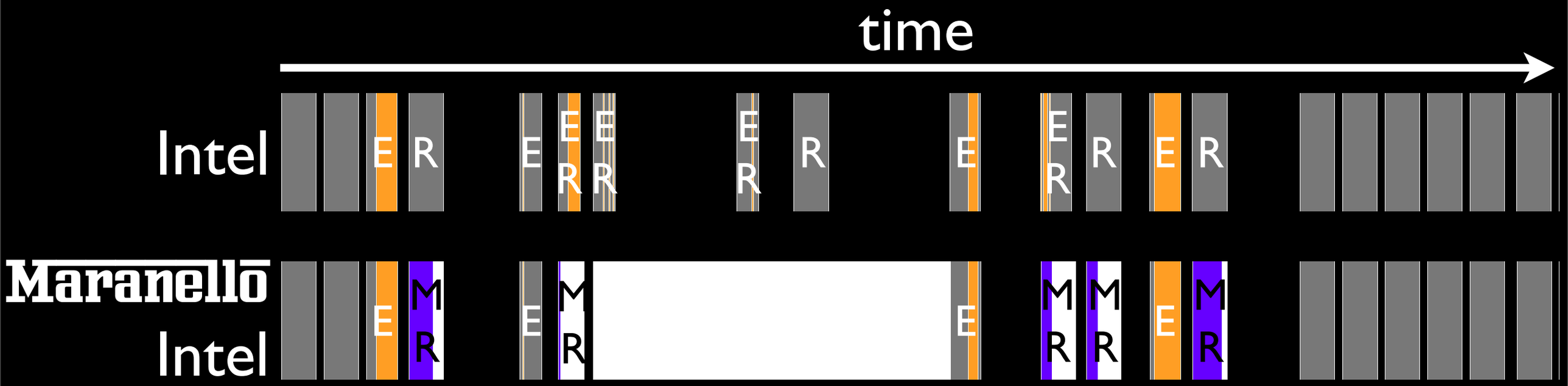


Windows Broadcom



Windows Atheros

Trace-driven Maranello simulator

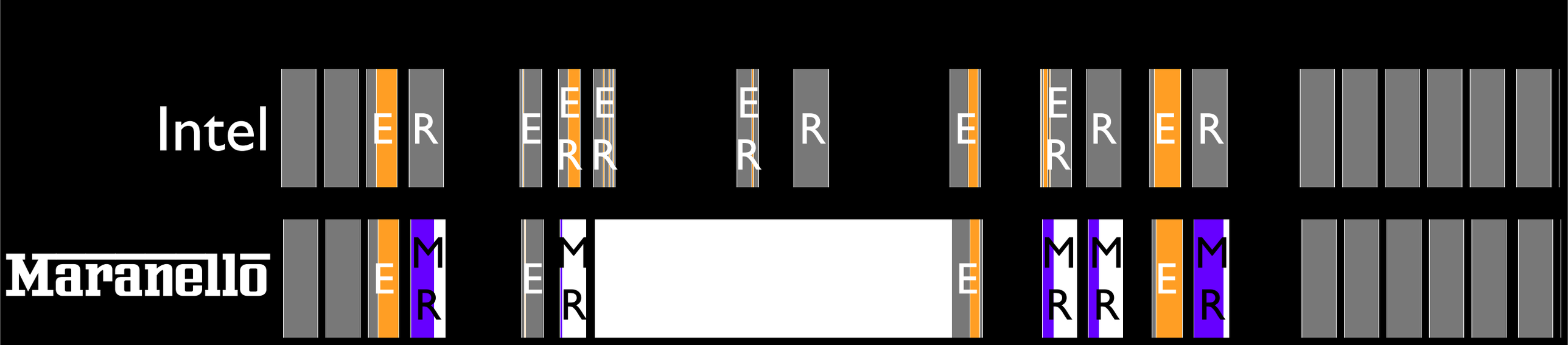


Insight: Maranello packets are subset of 802.11

Assuming Maranello delivers all nacks successfully

Trace-driven Maranellō simulator

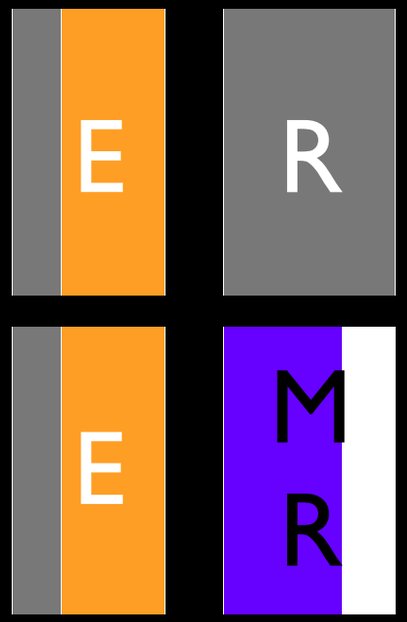
Trace-driven Maranello simulator



Insight: Maranello packets are subset of 802.11

Assuming Maranello delivers all nacks successfully

Trace-driven Maranello simulator



Insight: Maranello packets are subset of 802.11

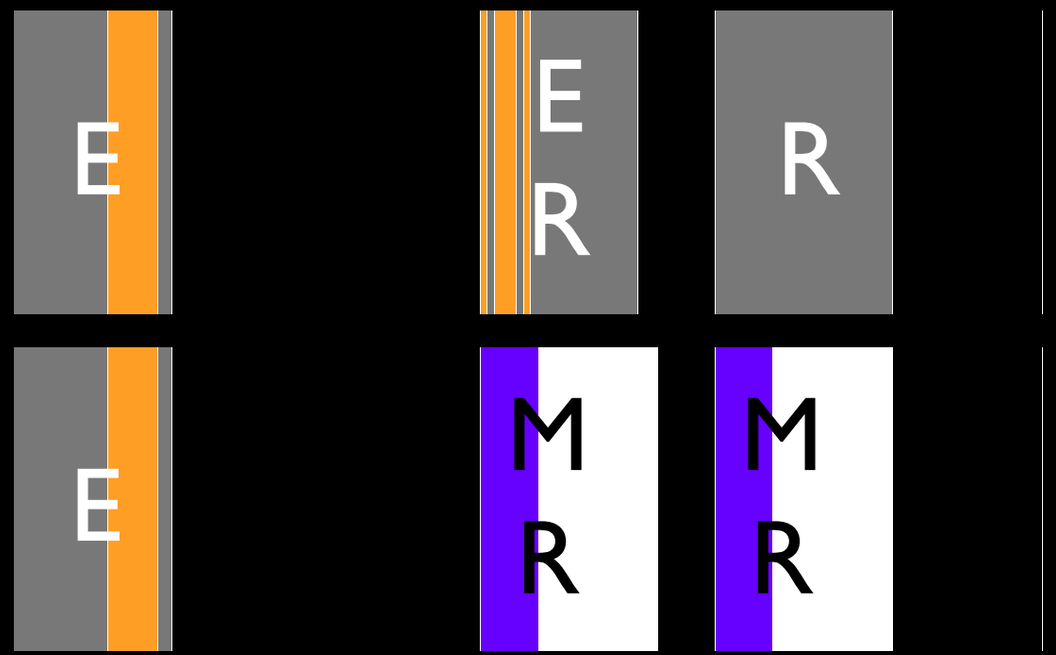
Assuming Maranello delivers all nacks successfully

Trace-driven Maranello simulator

Insight: Maranello packets are subset of 802.11

Assuming Maranello delivers all nacks successfully

Trace-driven Maranello simulator

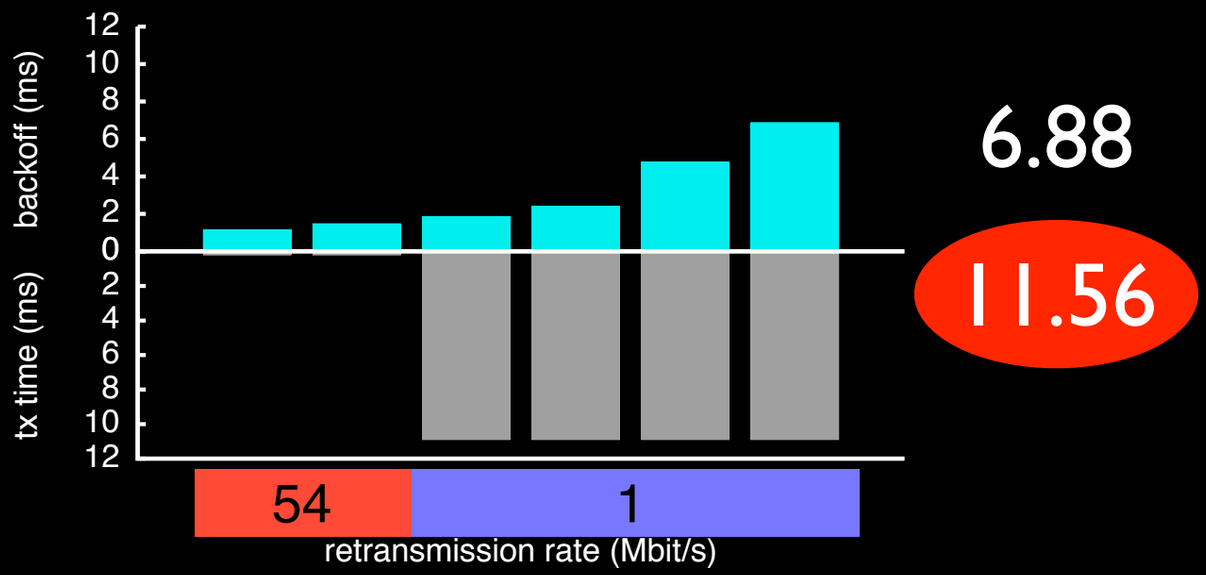


Insight: Maranello packets are subset of 802.11

Assuming Maranello delivers all nacks successfully

Avoiding low rate retransmissions

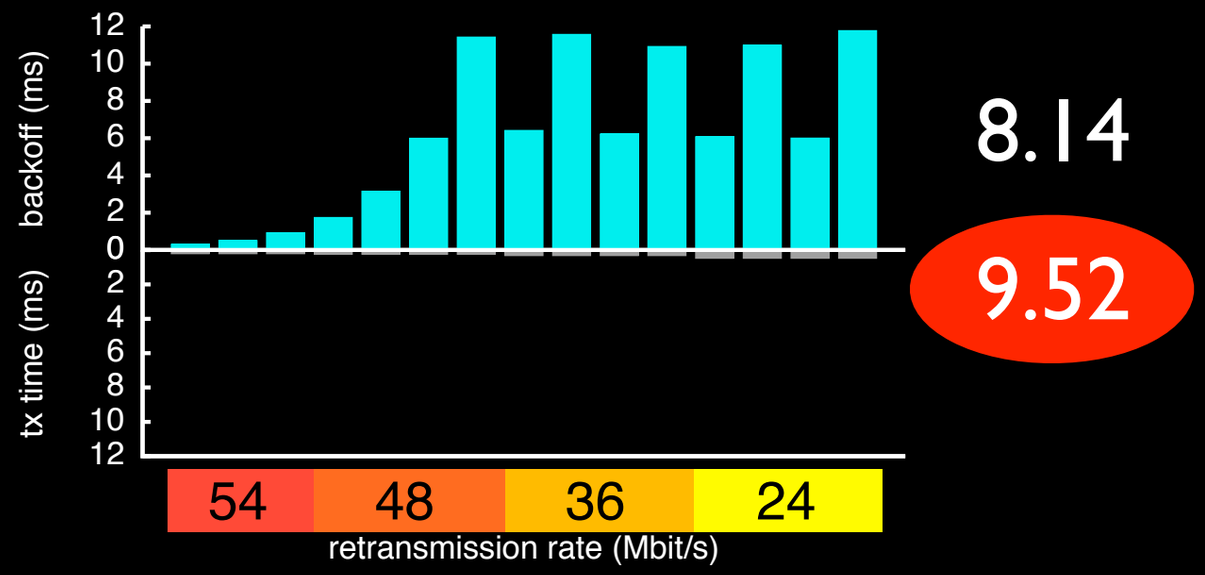
Maranello increases throughput



6.88

11.56

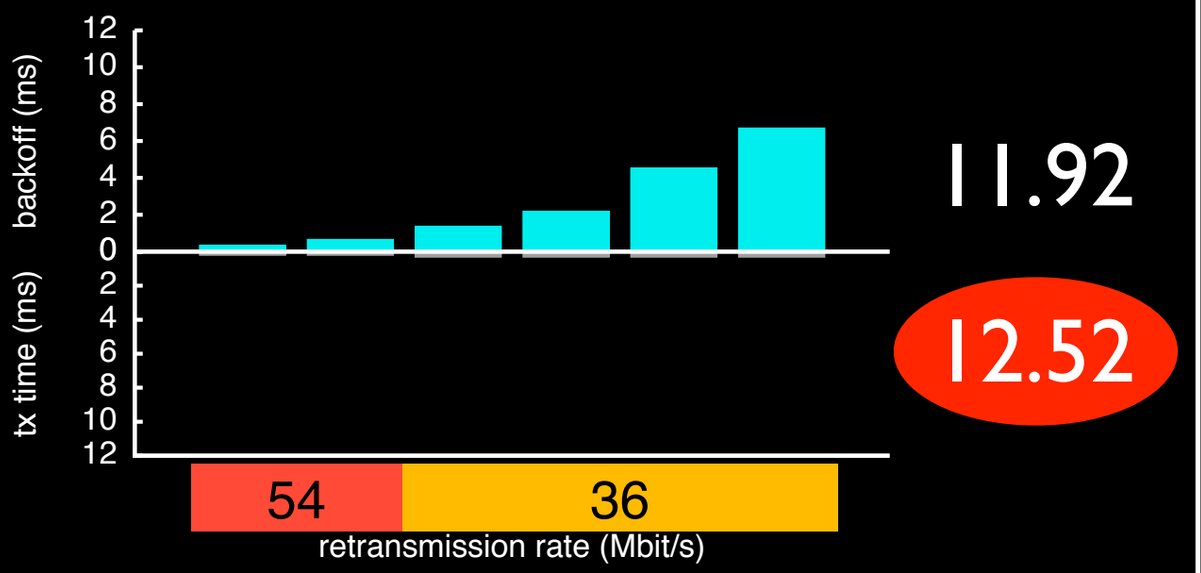
Linux ‘minstrel’ Broadcom



8.14

9.52

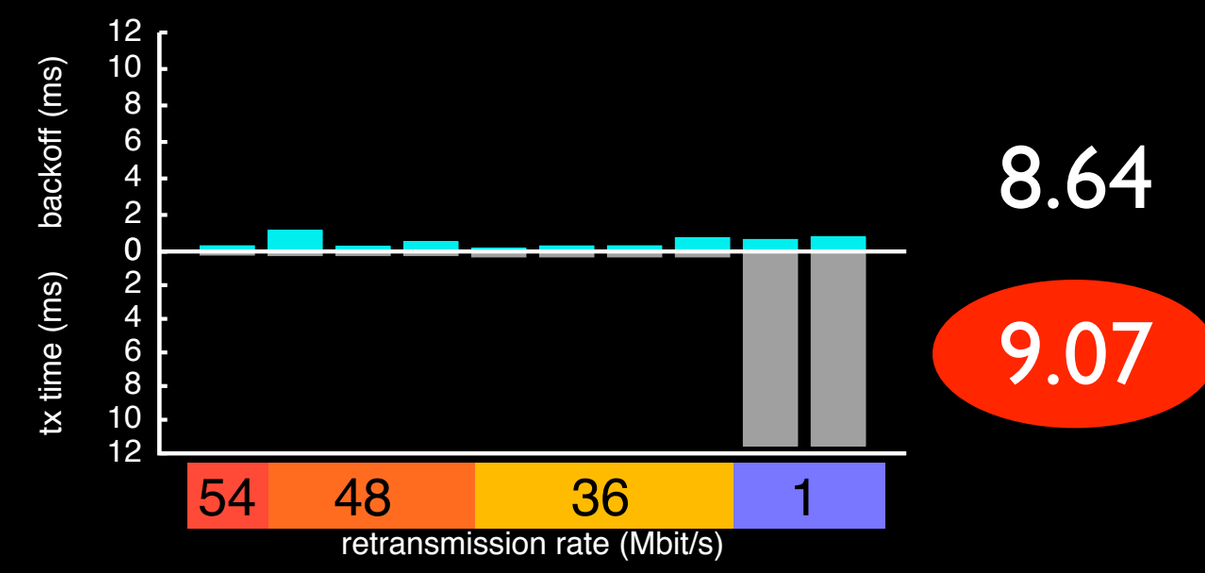
Windows Intel



11.92

12.52

Windows Broadcom

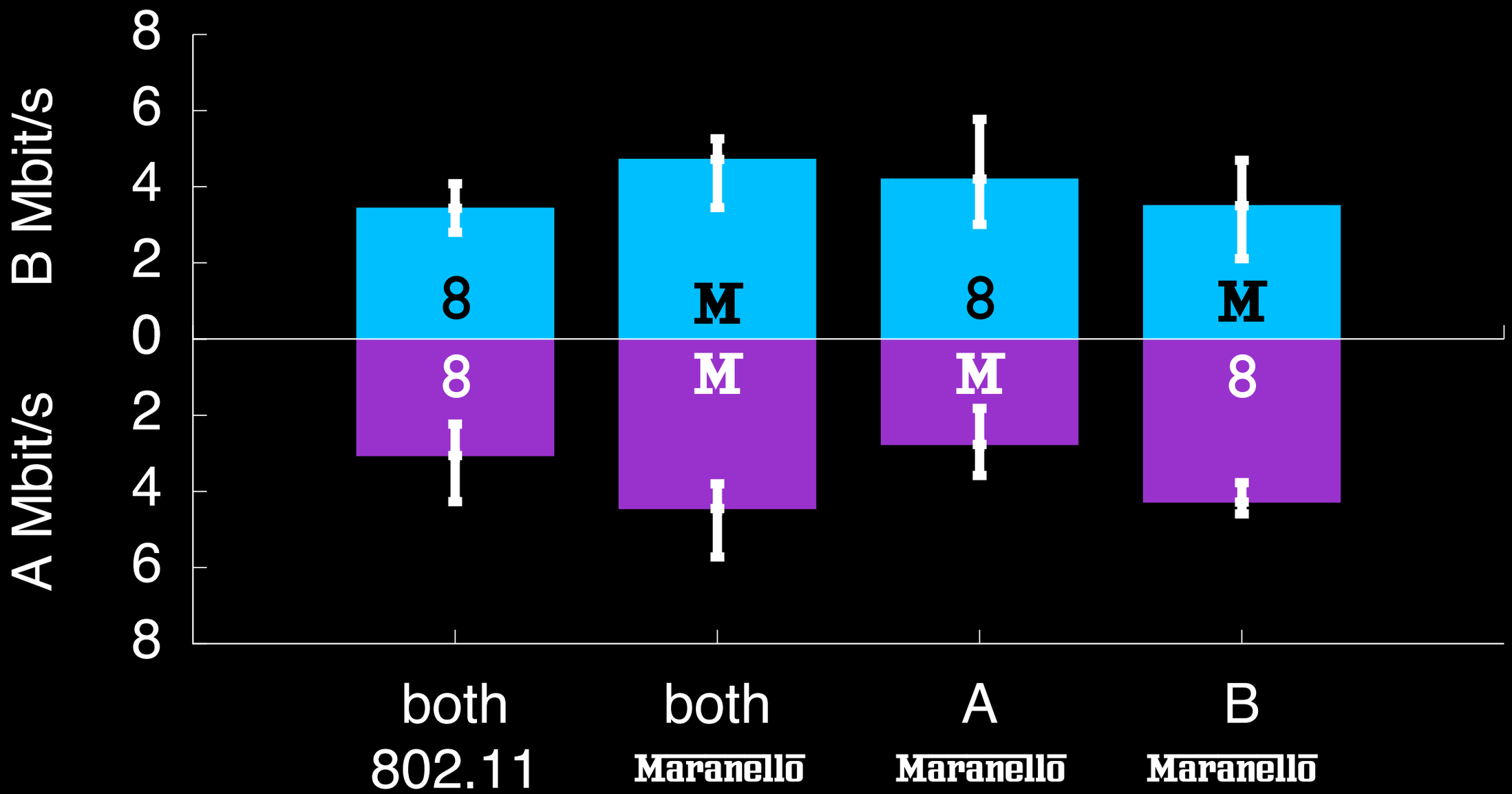


8.64

9.07

Windows Atheros

Compatibility with unmodified 802.11



- **Maranello** is faster than 802.11
- **Maranello** is friendly to 802.11

Conclusions

- Block-based recovery is simple and powerful.
- **Maranello** in firmware is *compatible* with 802.11 and *incrementally deployable* on existing APs and cards.
- **Maranello** increases throughput:
  - In various wireless environments
  - For popular retransmission behaviors
- **Maranello** is fair to unmodified 802.11.

www.cs.umd.edu/projects/maranello