

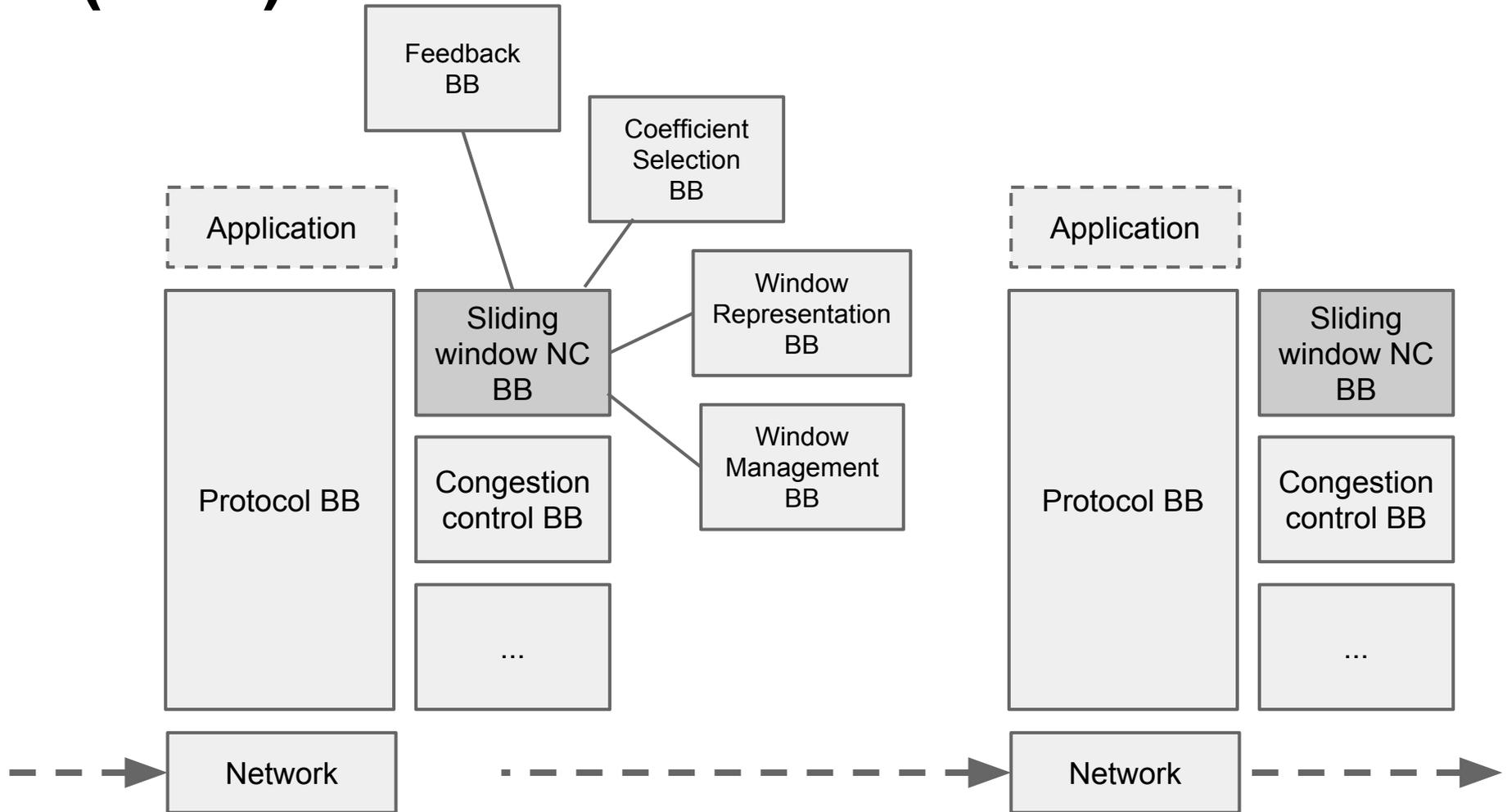
Considerations for an sliding window NC codec

Morten V. Pedersen

mvp@es.aau.dk

Aalborg University / Steinwurf ApS

Identifying the NC Building Blocks (BBs)



Can we define protocol BB <-> sliding window BB interface?

Drafts have been submitted to the nwcrg, but each define their own interface towards the protocol.

Question:

- Can we narrow down and specify these interfaces?
- What is the best way to specify such interfaces?

Example API:

```
uint32_t fec_encoding_id();
uint32_t fec_instance_id();

handle_t open_window(uint32_t max_size);
void add_symbol(handle_t window, uint32_t index,
               const uint8_t* data, uint32_t size);

// Read/write symbols
uint32_t payload_size();
uint32_t write_payload(handle_t window, uint8_t*)
uint32_t read_payload(handle_t window, uint8_t*)
bool is_uncoded(handle_t window, uint32_t index)

// Feedback packets
bool has_feedback();
uint32_t feedback_size();
uint32_t write_feedback(uint8_t*)
uint32_t read_feedback(uint8_t*)
....
```

Benefit of defining API

- Split what is protocol specific and what is FEC block specific
- Ease protocol integration
- Ease performance evaluation
 - Run protocol independent performance evaluation
- RG action: Define a “minimal” API for how the protocol can interact with the sliding windows BB?

The End

Evaluation Metrics

How to evaluate/compare the different sliding window schemes?

- **Decoding probability**
 - Tricky(er) for the non PRNG based schemes.
 - Perhaps based on arrival and erasure traces.
- **Decoding delay/latency**
 - Put an uncoded packet in - when does it come out at the sources.

Evaluation Metrics

- Overhead
 - Signaling (feedback and protocol overhead)
- Complexity
 - #Finite field calculations
 - Memory consumption
 - Computation cycles
- Etc.
- RG action: Create document on evaluation metrics.