

Adaptive Boundary Protocol

(Will be filled in later)

February 15, 2026

1 Hybrid Directional with Adaptive Boundaries

Combine directional assignment with dynamic boundary adjustment based on usage patterns.

1.1 Protocol Description

Initial Setup:

- Alice: pads $[1, \lfloor n/3 \rfloor]$
- Bob: pads $[\lfloor n/3 \rfloor + d + 1, \lfloor 2n/3 \rfloor]$
- Charlie: pads $[\lfloor 2n/3 \rfloor + d + 1, n]$
- Each party uses pads sequentially within their region

Adaptive Borrowing: When party P exhausts their region:

- P can "borrow" from an adjacent idle region if the gap constraint is satisfied
- Borrowing extends P 's region by k pads (e.g., $k = d$)
- Borrowed regions are marked to prevent conflicts

1.2 Analysis

Advantages:

- Adapts to actual usage patterns
- Can perform better than $\frac{2n}{3}$ waste when parties have unequal sending rates
- Runtime: $O(1)$ per message in typical case

Waste Analysis:

- Best case (equal sending): waste $\approx 2d$ pads (for gaps)
- Worst case (only one sender): waste $< \frac{2n}{3}$ due to borrowing
- Expected waste depends on borrowing efficiency

2 Algorithm

Algorithm 1 Adaptive Boundary Protocol

```
1: State:  $region\_start[P]$ ,  $region\_end[P]$ ,  $next\_pad[P]$ 
2:
3: procedure SENDMESSAGE( $P$ ,  $message$ )
4:   if  $next\_pad[P] \leq region\_end[P]$  then
5:     if GapConstraintSatisfied( $P$ ,  $next\_pad[P]$ ) then
6:       UsePad( $P$ ,  $next\_pad[P]$ ,  $message$ )
7:        $next\_pad[P] \leftarrow next\_pad[P] + 1$ 
8:       return SUCCESS
9:     end if
10:   else
11:     if CanBorrow( $P$ ) then                                 $\triangleright$  Try to borrow from adjacent region
12:       ExtendRegion( $P$ ,  $k$ )
13:       Retry SendMessage( $P$ ,  $message$ )
14:     else
15:       return FAIL
16:     end if
17:   end if
18: end procedure
19:
20:
21: procedure CANBORROW( $P$ )
22:   Find adjacent party  $Q$  with unused pads
23:   return ( $region\_end[P] + d + k < next\_pad[Q]$ )
24: end procedure
25:
26: procedure EXTENDREGION( $P$ ,  $k$ )
27:    $region\_end[P] \leftarrow region\_end[P] + k$ 
28: end procedure
```
