stream, err := client.BidirectionalStream(ctx)

if err != nil {

// Handle error

}

waitc := make(chan struct{})

go func() {

for {

in, err := stream.Recv()

if err == io.EOF {

// read done.

close(waitc)

return

}

if err != nil {

// Handle error

return

}

// Process in

}

}()

for \_, req := range requests {

if err := stream.Send(req); err != nil {

// Handle error

return

}

}

stream.CloseSend()

<-waitc