ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Algorithmen für drahtlose Netzwerke

MACAW

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

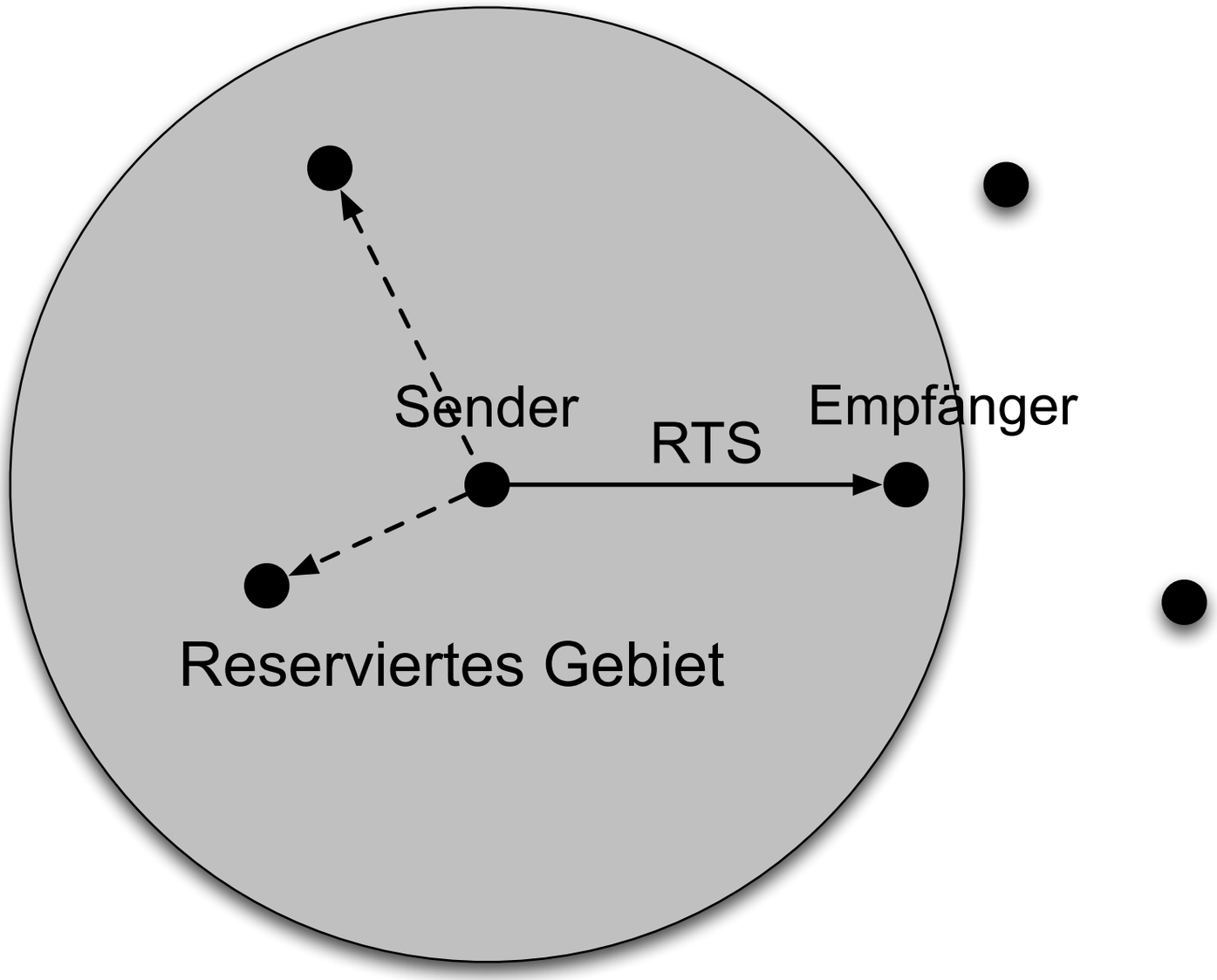
MACAW

- **Bharghavan, Demers, Shenker, Zhang**
  - MACAW: A Media Access Protocol for Wireless LAN‘s, SIGCOMM 1994
  - Palo Alto Research Center, Xerox
- **Ziel**
  - Neuentwurf von MACA
  - Besserer Backoff Mechanismus
  - Gerechtere Bandweiten Aufteilung durch *Streams*
  - Höhere Effizienz
    - durch 4er und 5er-Handshake

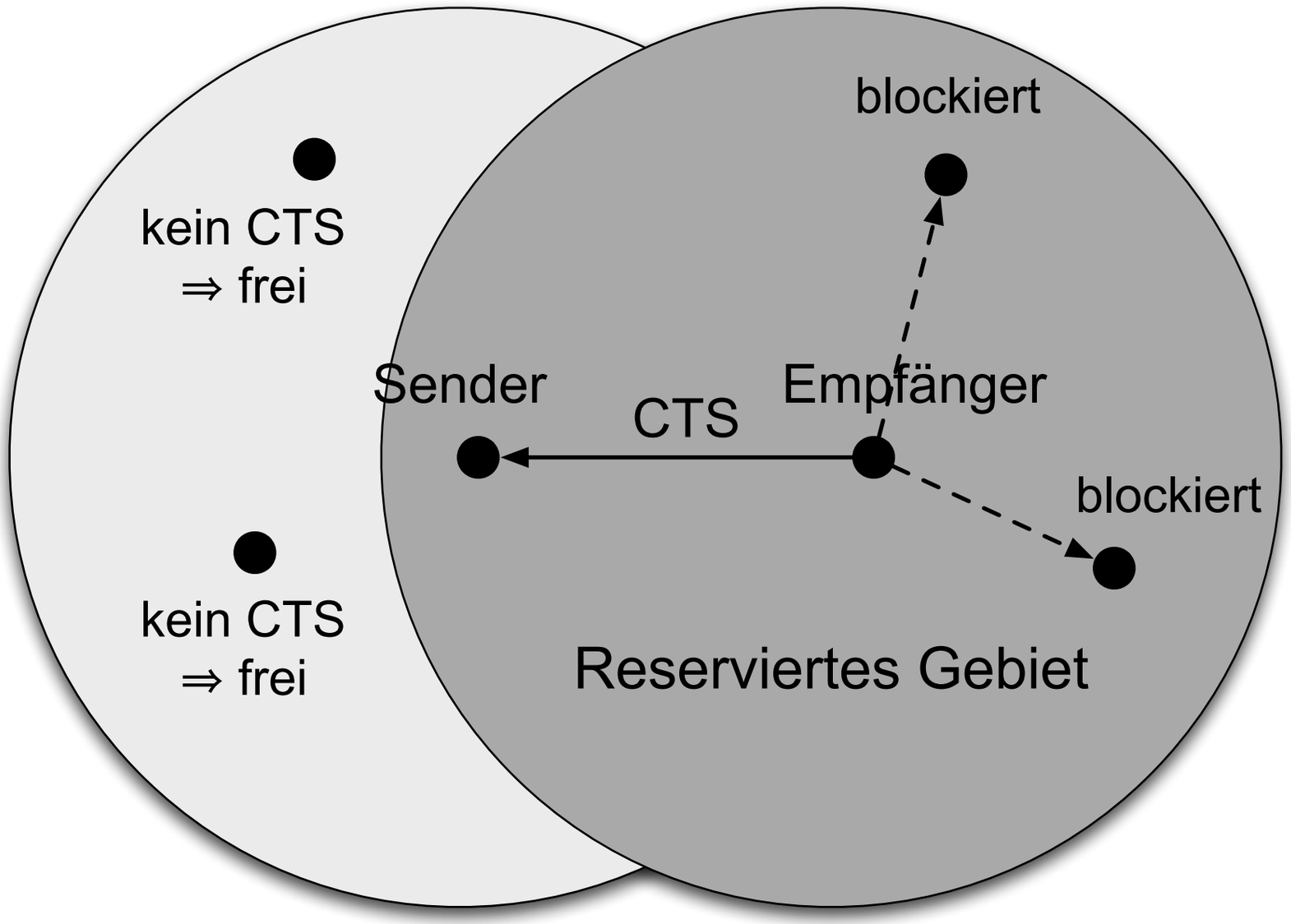
Bestätigungen in der Sicherungsschicht

- **MACA**
  - kennt keine Bestätigungen
  - werden durch Transportschicht initiiert
  - sehr ineffizient
- **Wie sieht MACA mit Bestätigungen (ACK) aus?**

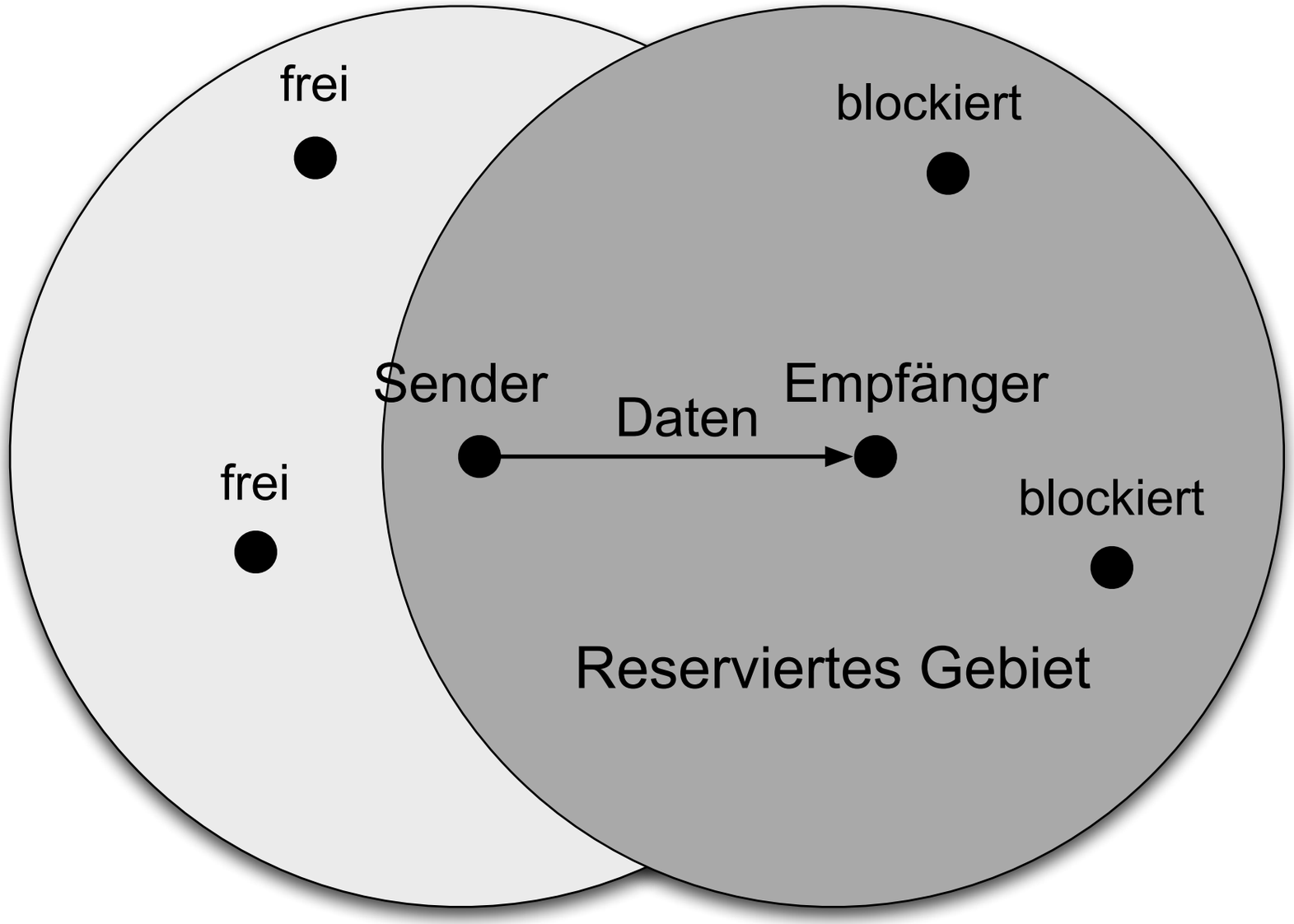
MACA mit ACK
RTS



MACA mit ACK
CTS

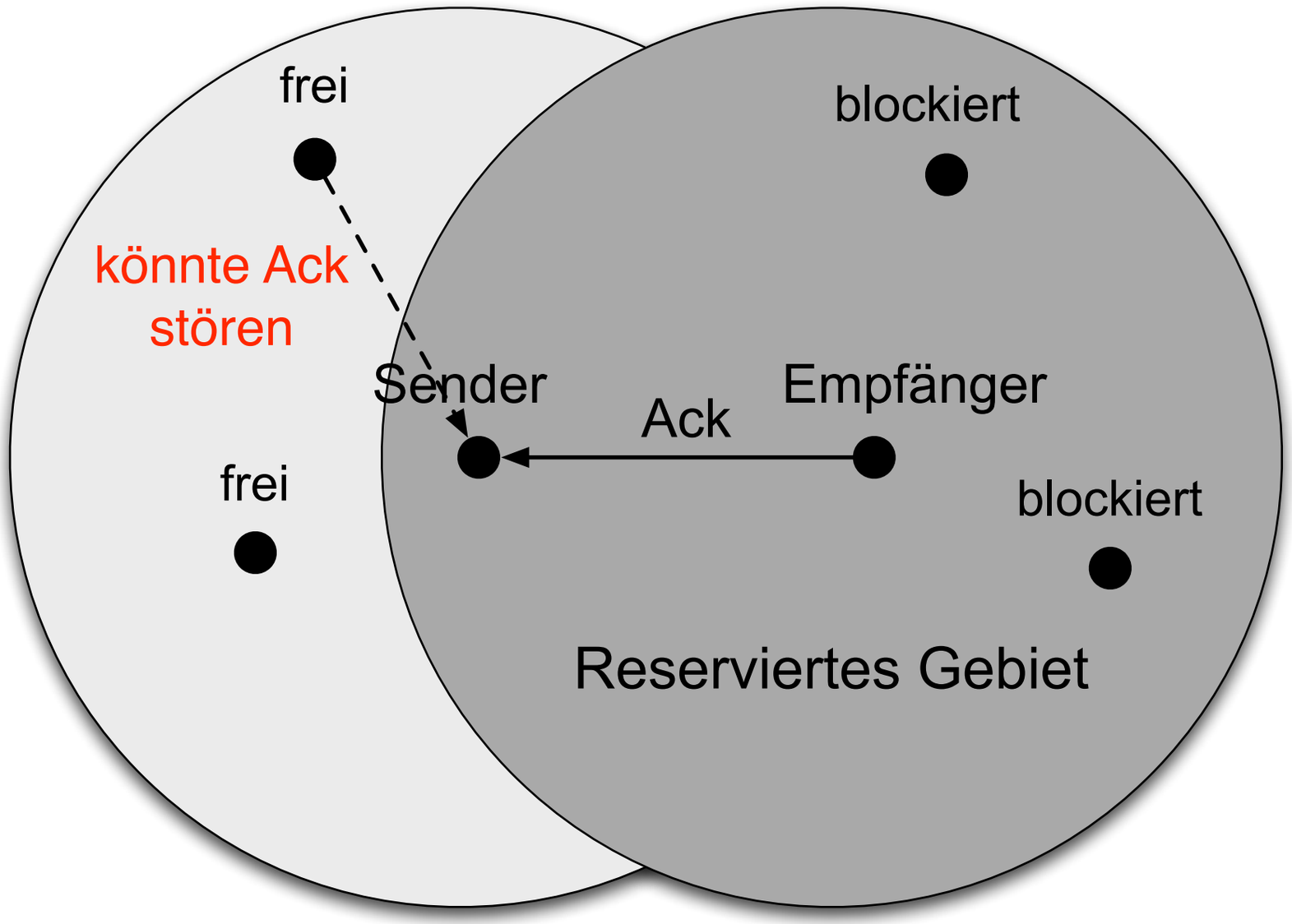


MACA mit ACK
DATA



MACA mit ACK
ACK

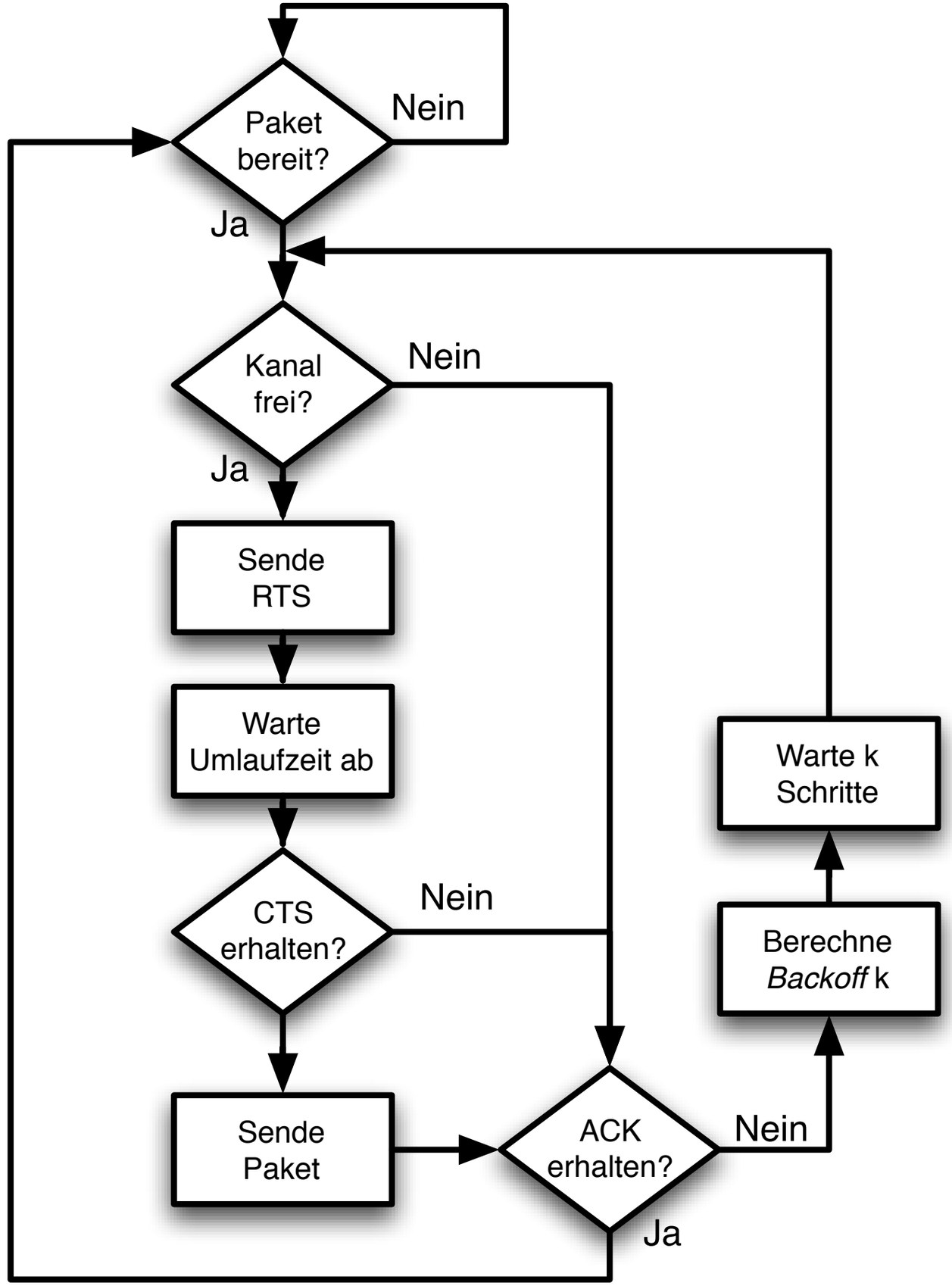
Hidden Terminal



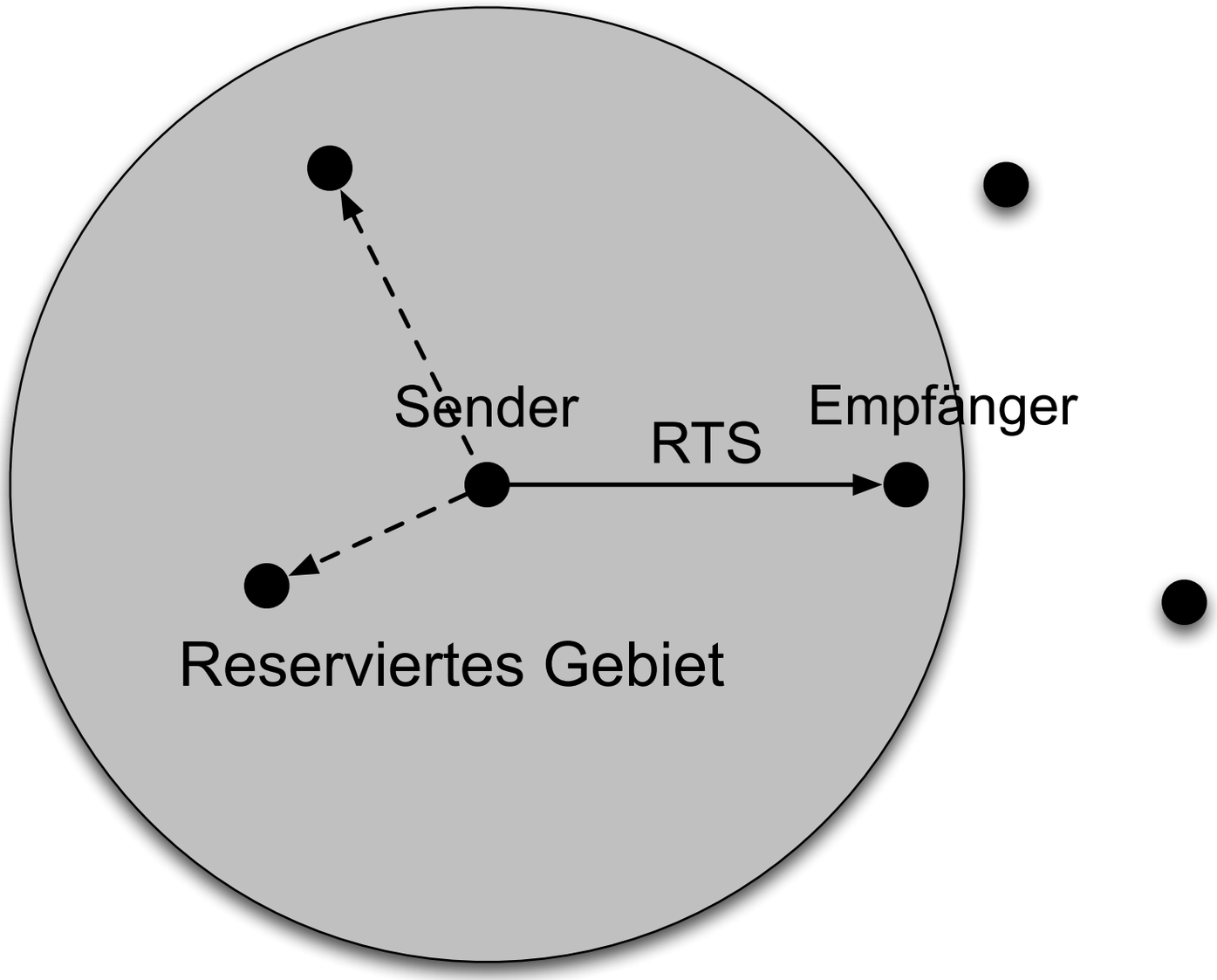
MACAW
4er Handshake

- **Hauptbeteiligte**
  - Sender sendet RTS
  - Empfänger antwortet mit CTS
  - Sender sendet Datenpaket
  - Empfänger bestätigt mit ACK
- **Unbeteiligte**
  - Knoten, die RTS oder CTS hören, sind für diese Zeit blockiert
  - RTS und CTS beschreiben die Dauer der Übertragung
- **Sender wiederholt RTS, falls kein ACK empfangen wurde**
  - Falls Empfänger ACK doch geschickt hat
  - dann schickt der Empfänger statt CTS ein ACK und der

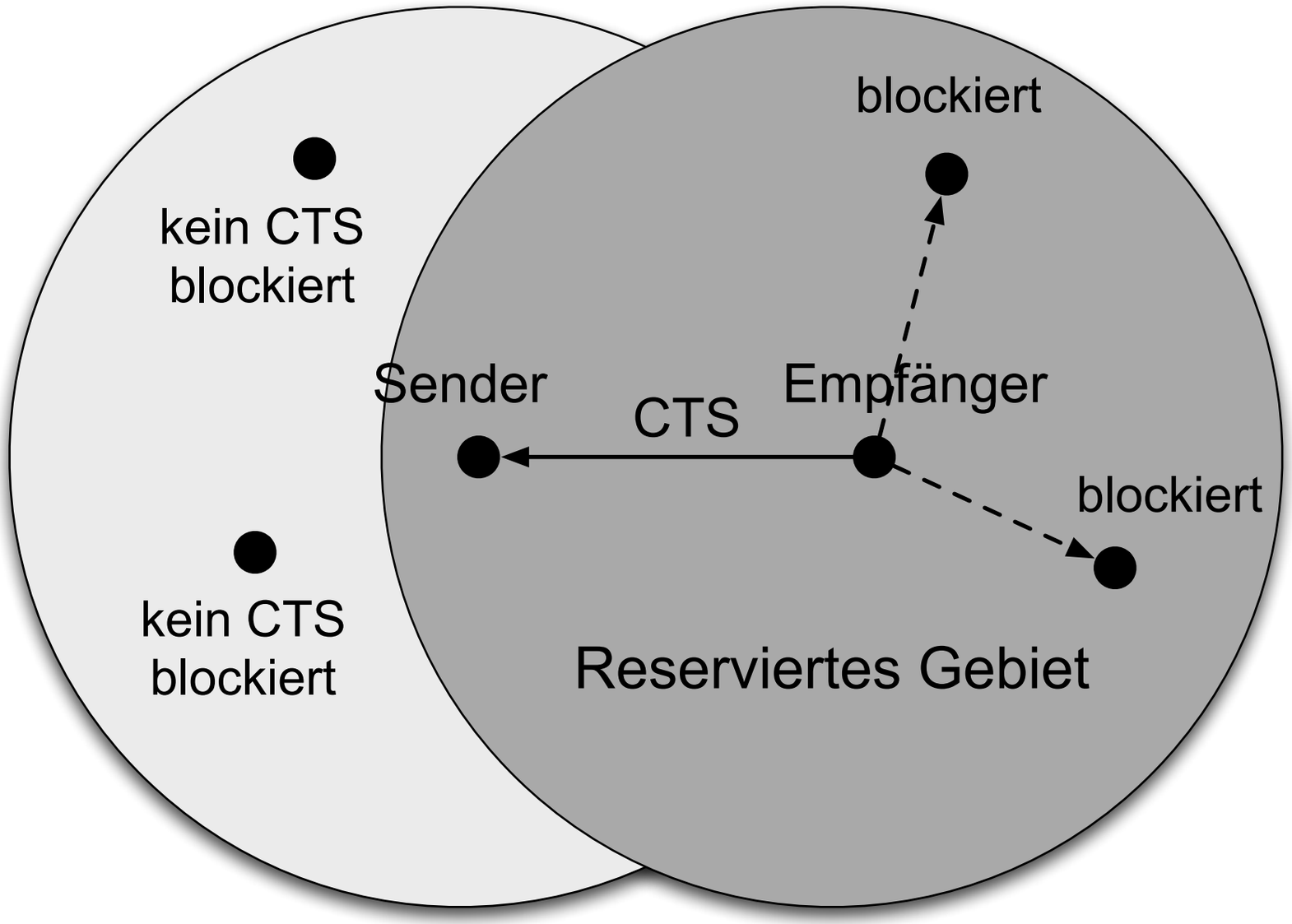
MACAW


Paket bereit?
Nein
Ja
Kanal frei?
Nein
Ja
Sende RTS
Warte Umlaufzeit ab
CTS erhalten?
Nein
Sende Paket
ACK erhalten?
Nein
Ja
Warte k Schritte
Berechne *Backoff* k

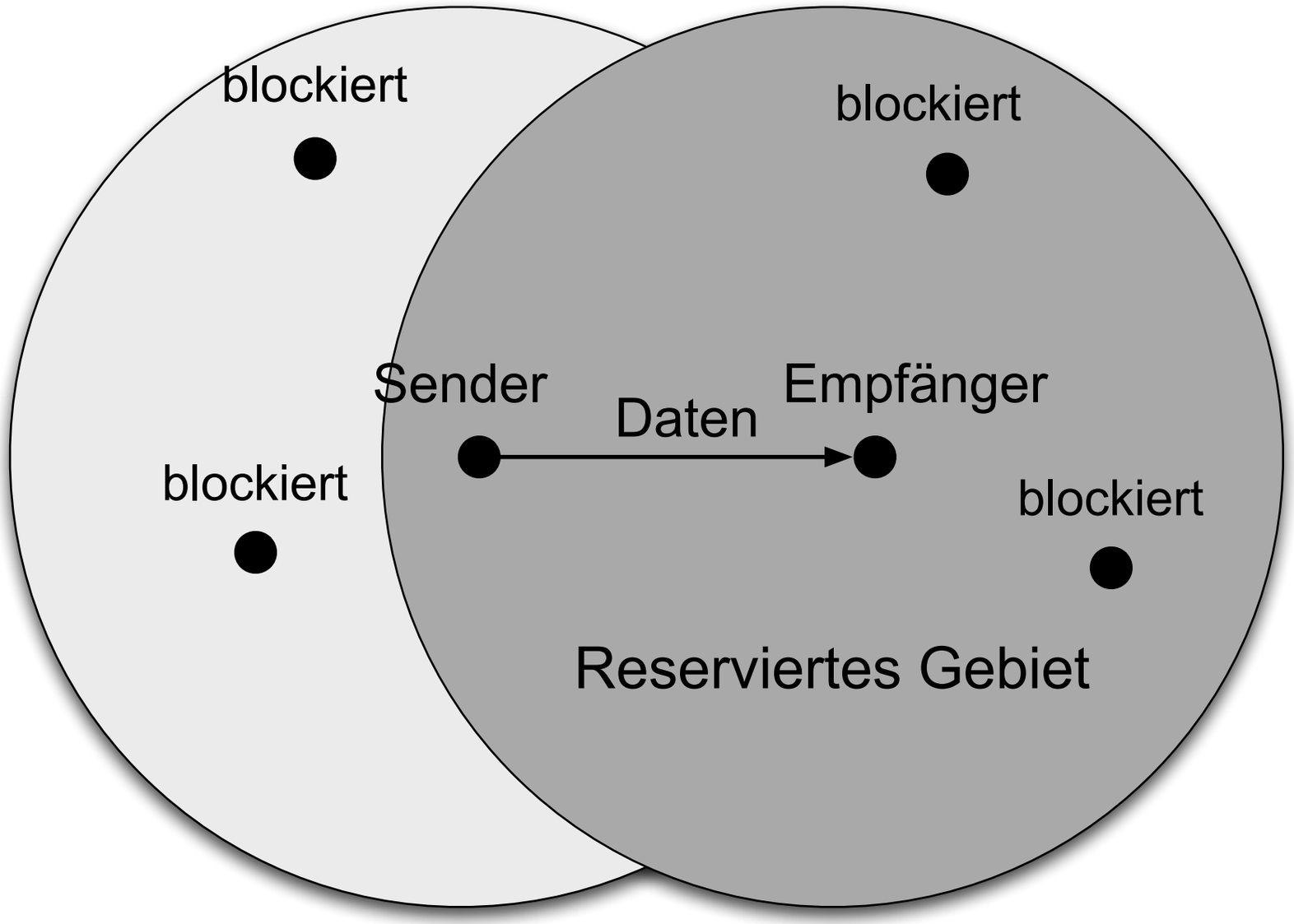
MACA 4er-Handshake
RTS



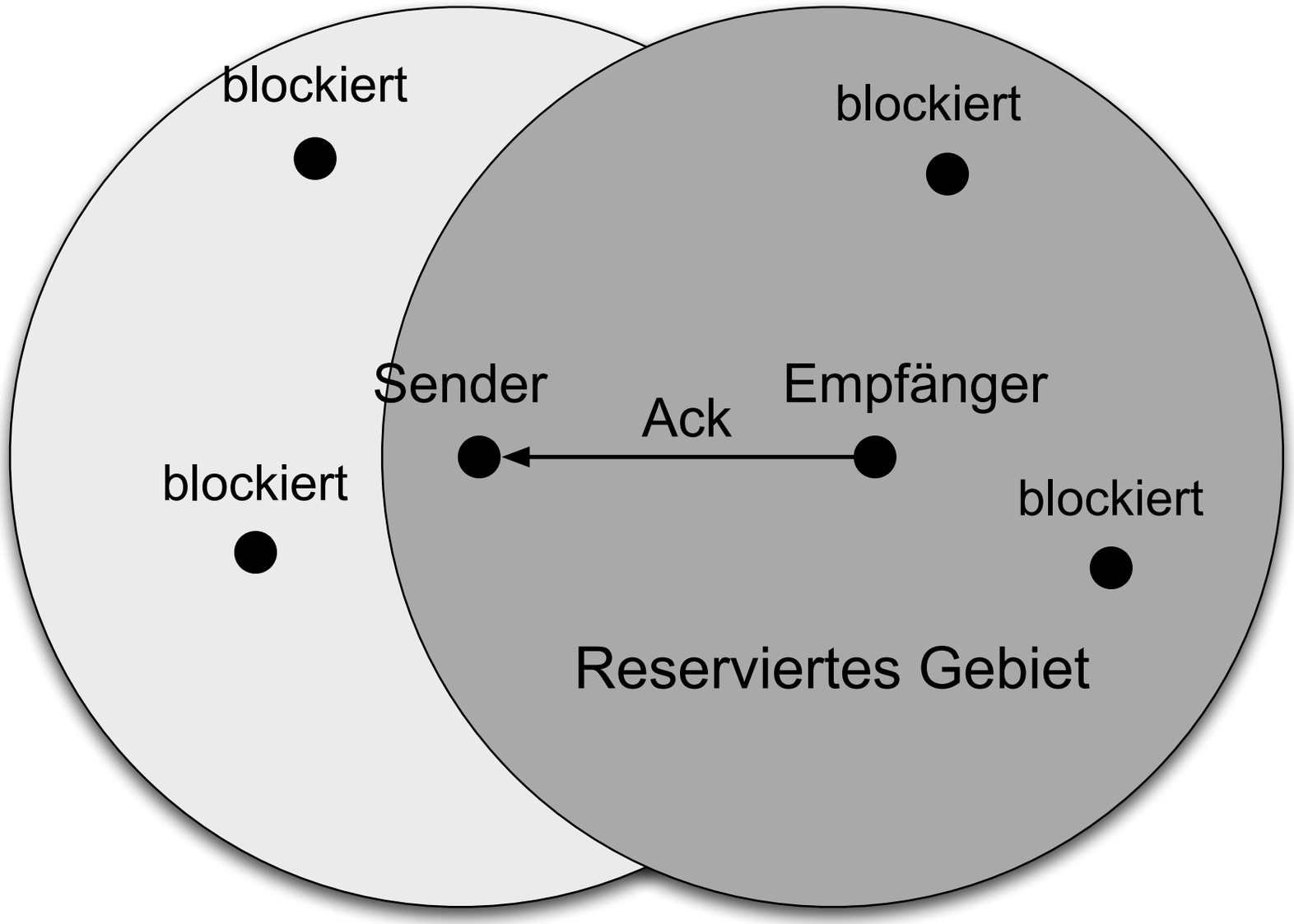
MACAW 4er-Handshake
CTS



MACAW 4er-Handshake Daten



MACAW 4er-Handshake
RTS



Bestätigungen

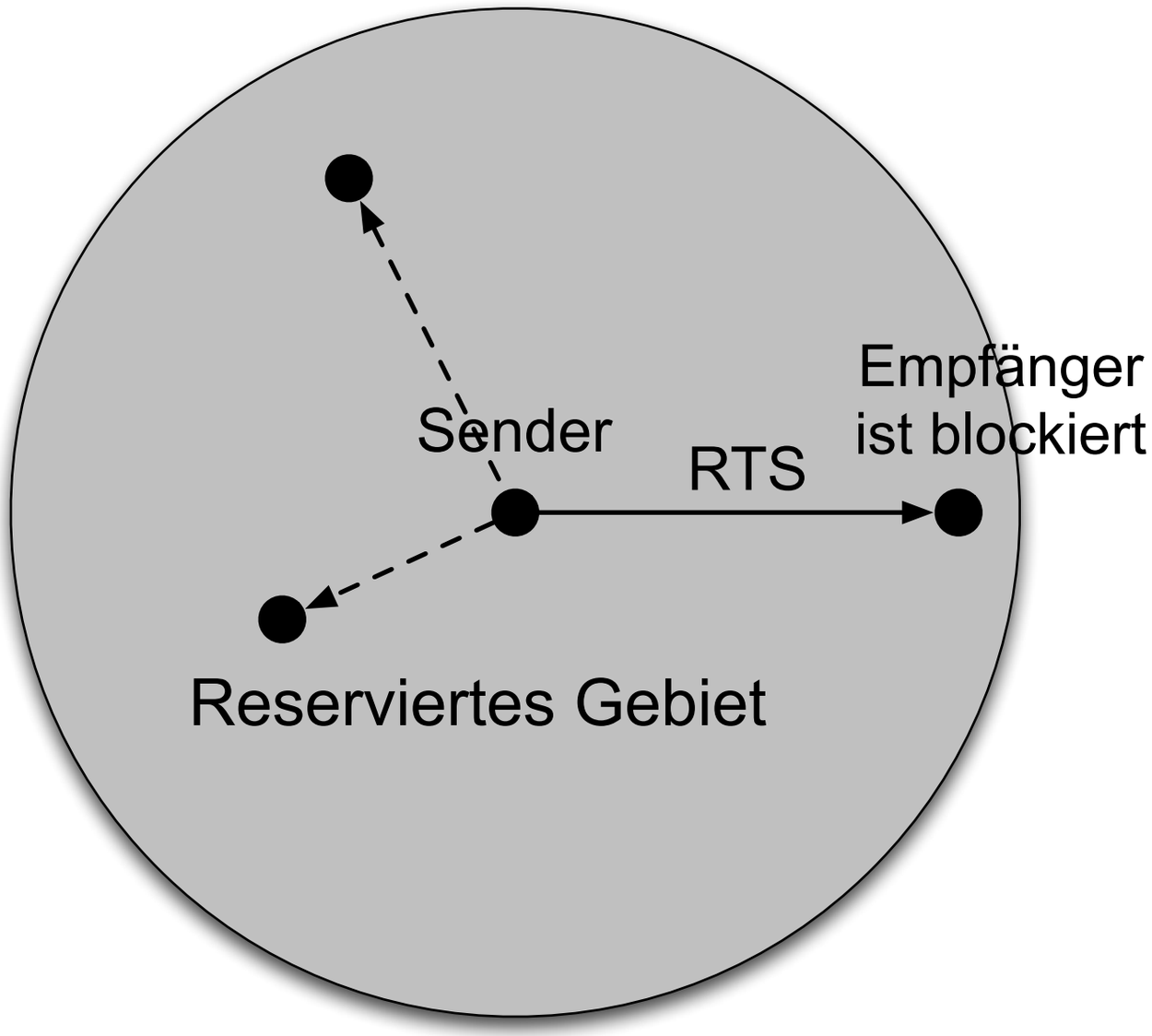
- **Hinzufügen von Bestätigungen (ACK) zu MACA**
  - In MACA Aufgabe der Transport-Schicht
- **Führt zu drastischen Verbesserungen des Durchsatzs selbst bei moderaten Fehlerraten**

Fehlerrate	Durchsatz	
	RTS-CTS-DATA	RTS-CTS-DATA-ACK
0	40	37
0,001	37	37
0,01	17	36
0,1	2	10

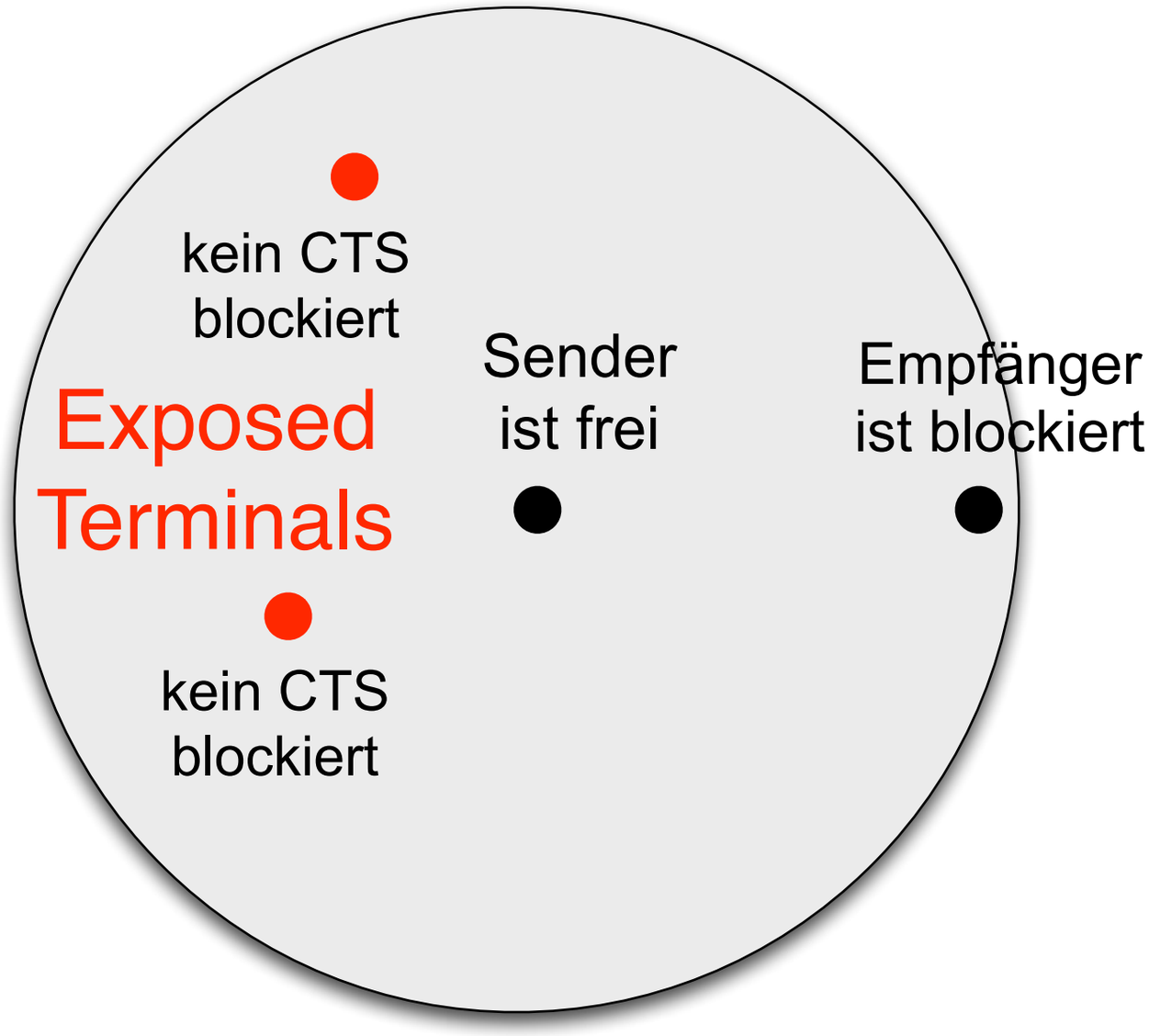
MACAW
4er Handshake

- **Worst-Case Blockade**
  - Sender sendet RTS
  - Empfänger ist blockiert
  - Sender ist frei
  - Aber Umfeld des Senders ist blockiert

MACAW 4er-Handshake
RTS



MACAW 4er-Handshake
CTS steht aus



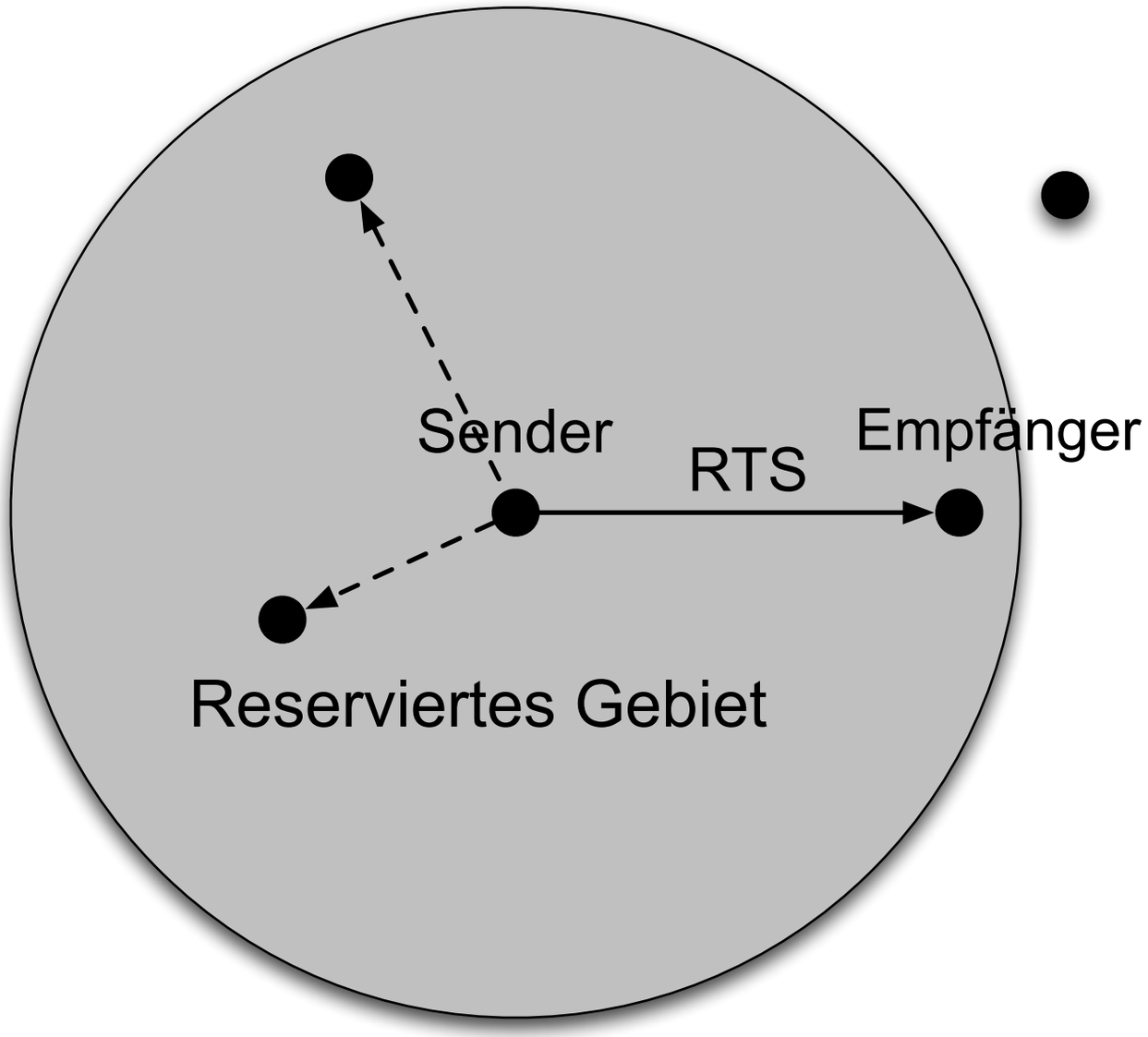
MACAW
5er Handshake

- **Exposed Terminal problem in 4er-Handshake verschlimmert**
  - Abgehörte RTS führen zur Blockade
  - Selbst wenn kein Datentransfer stattfindet
- **Lösung**
  - Exposed Terminals werden informiert, falls Daten transportiert werden
  - Mittels Nachricht DS (data send)
- **5er-Handshake verkürzt Wartezeit für Exposed Terminals**

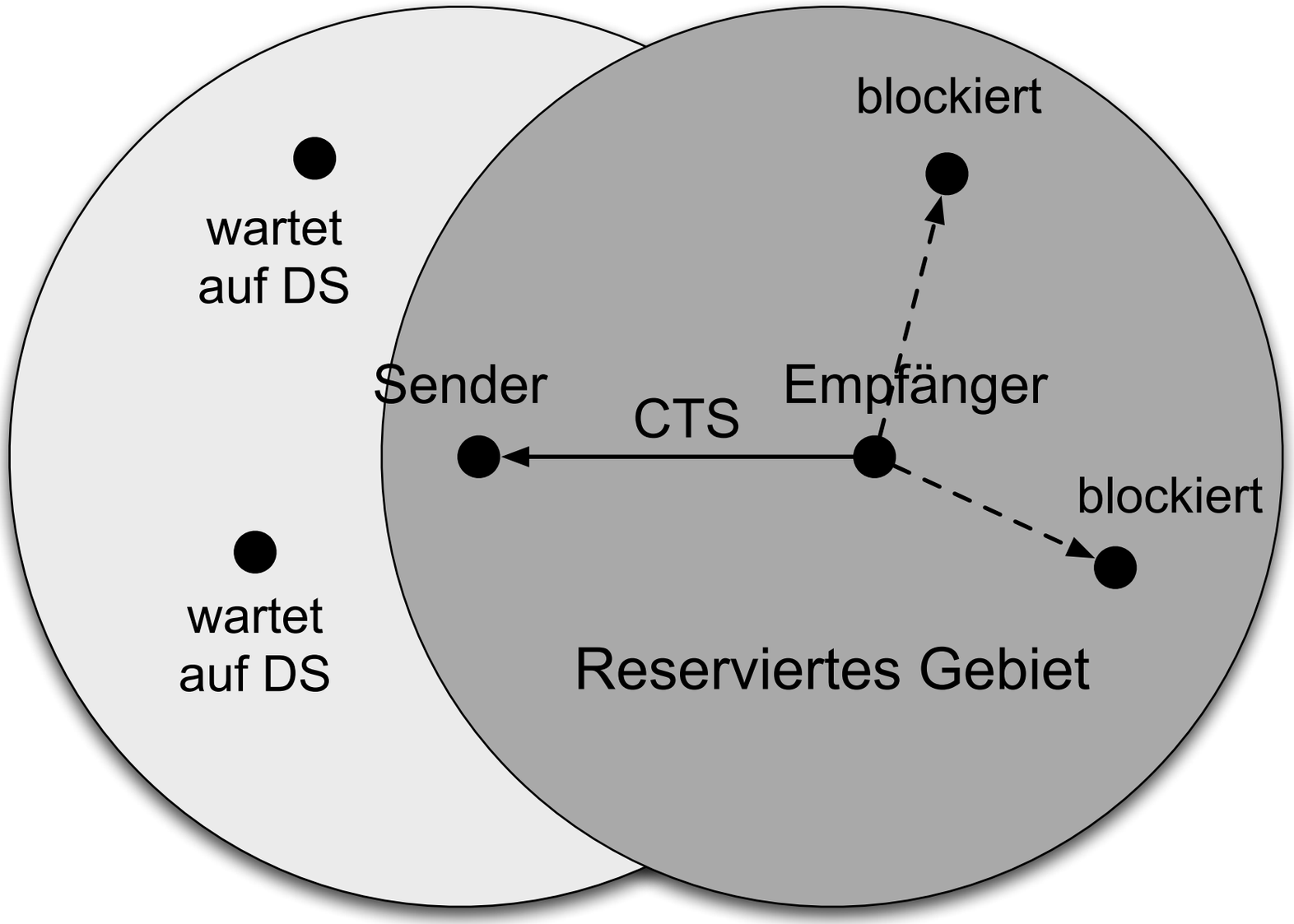
MACAW
5er Handshake

- **Beteiligte**
  - Sender sendet RTS
  - Empfänger antwortet mit CTS
  - Sender sendet DS (Data Send)
  - Sender sendet DATA PACKET
  - Empfänger bestätigen mit ACK
- **RTS und CTS kündigen die Dauer der Übertragung an**
- **Blockierte Knoten**
  - haben RTS und kurz danach DS gehört
  - haben CTS gehört
- **Geringer Aufwand verringert die Anzahl der Exposed Terminals erheblich**

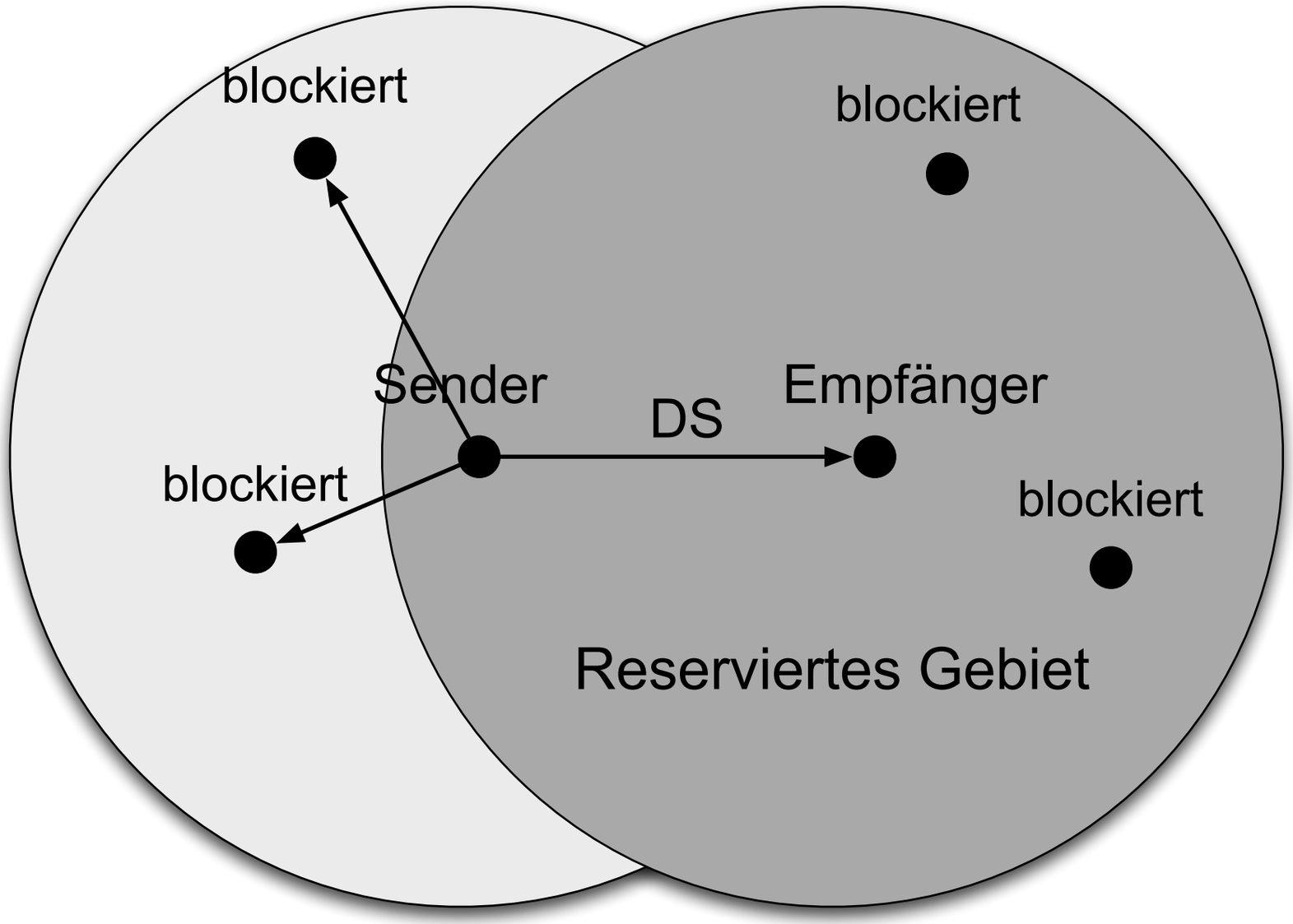
MACAW 5er-Handshake
RTS



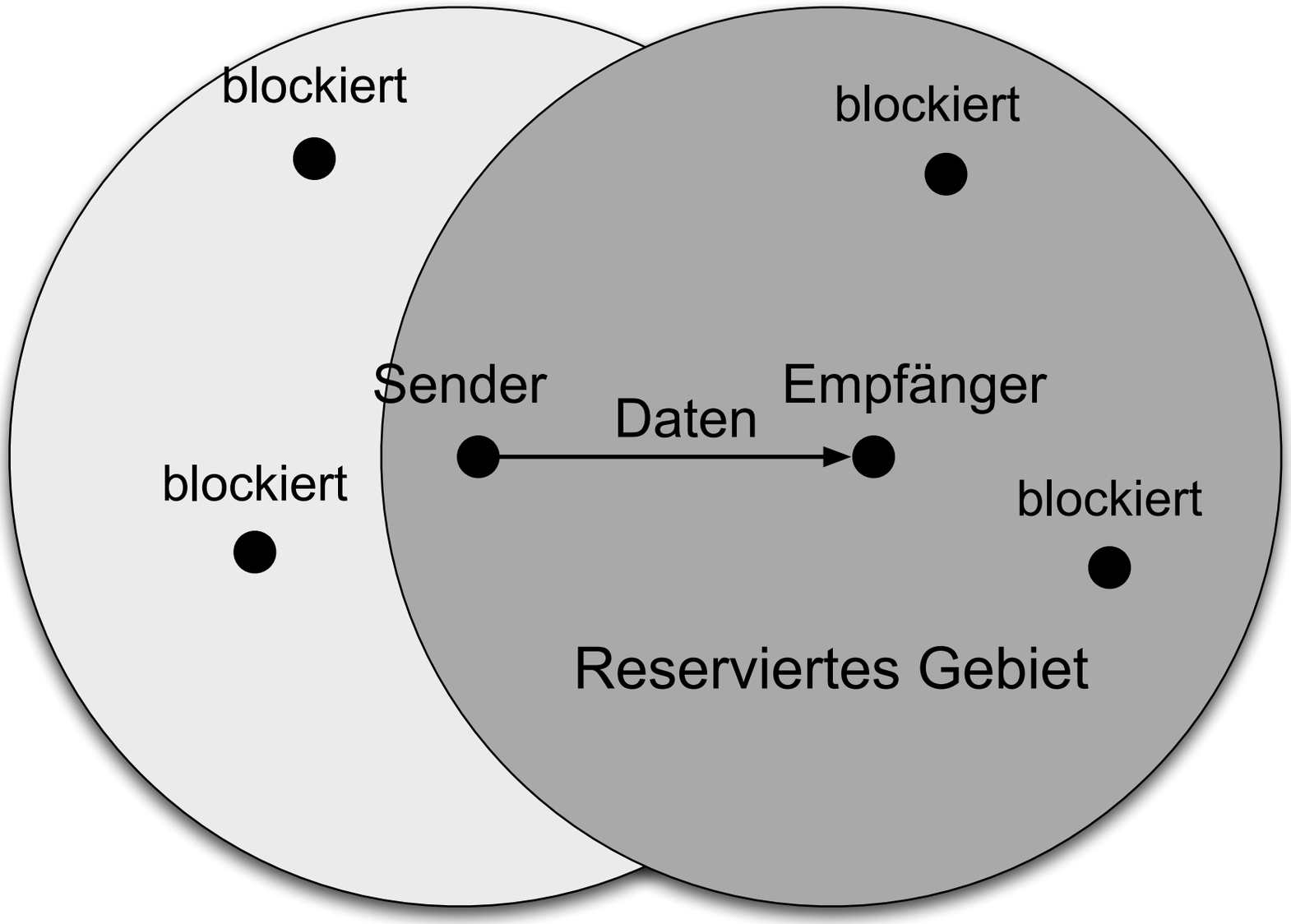
MACAW 5er-Handshake
CTS



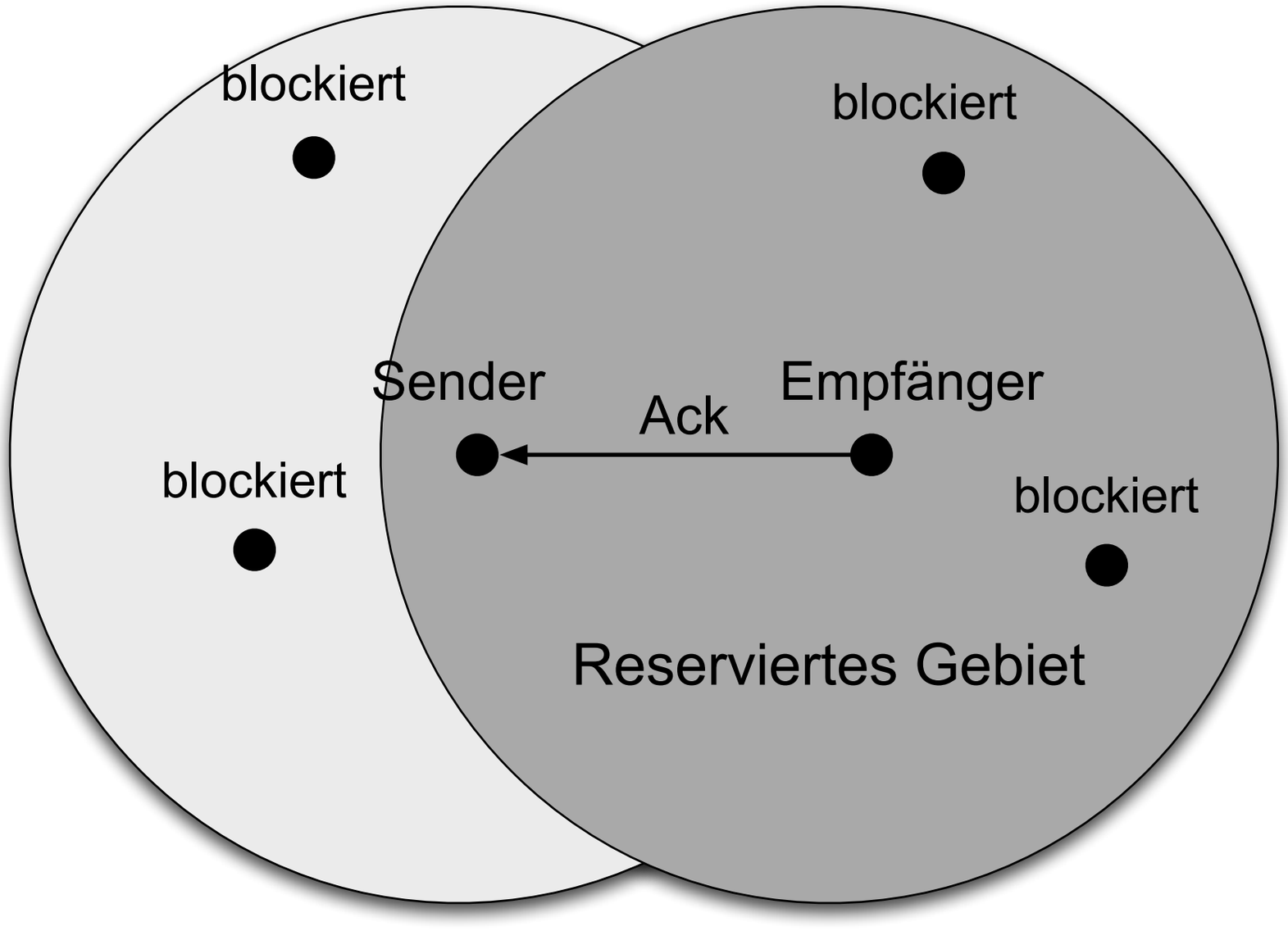
MACAW 5er-Handshake DS



MACAW 5er-Handshake Data



MACAW 5er-Handshake ACK



Unfaire Aufteilung

- **Sowohl 4er als auch 5er-Handshake erzeugen unfaire Aufteilung**
  - A will eine Menge Daten zu B senden
  - D will eine Menge Daten zu C senden
  - C hört B und D, aber nicht A
  - B kann A und C hören, aber nicht D
- **A erhält als erster den Kanal**
- **D sendet RTS und wird blockiert**
  - Backoff für D verdoppelt sich
- **Bei der nächsten Übertragung**
  - A hat kleineren Backoff und
  - größere Chance für Kanalzugriff

RRTS

- **Lösung**
  - C sendet RRTS (Request for Request to Send)
    - wenn ACK empfangen wurde
  - D sendet RTS, etc.
- **Warum RRTS statt CTS?**
  - Wenn Nachbarn CTS hören, sind sie länger blockiert
  - Möglicherweise ist D aber momentan nicht verfügbar

Multiple Stream Model (V-MAC)

- **Üblich: Simple Stream**
  - Alle Pakete werden mit den selben MAC-Parametern übermittelt
- **Multiple Stream**
  - Jeder Datenstrom erhält sein eigenes MAC-Protokoll
  - mit eigenen Backoff-Parametern
- **Damit ergibt eine größere Fairness**
  - Denn Knoten mit mehreren Datenströmen erhalten schnelleren Zugriff auf das Medium

Backoff Algorithms

- **Nach Kollision wird Wartezeit zufällig aus {1,.. Backoff} gewählt**
- **Binärer Exponentielles Backoff (BEB) algorithm**
  - Erhöhung nach Kollision:
    - Backoff = min{2*Backoff, Maximaler-Backoff}
  - Ansonsten:
    - Backoff = Minimaler-Backoff
- **Multiplicative increase linear decrease (MILD)**
  - Erhöhung:
    - Backoff = min{1,5 Backoff, Maximaler-Backoff}
  - Ansonsten:
    - Backoff = max{Backoff - 1, Minimaler-Backoff}

Information-Verteilung für Backoff-Algorithmus

- **Backoff-Parameter werden mitgehört**
  - Teilnehmer passen ihre Parameter den mitgehörten Backoff-Werten an
  - Mit Hilfe von MILD
- **Motivation**
  - Wenn alle Teilnehmer den selben Backoff-Wert haben, dann ist die Fairness erreicht

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Algorithmen für drahtlose Netzwerke

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

CoNe
Freiburg

IIF
INSTITUT FÜR
INFORMATIK
FREIBURG