

BLUETOOTH – SCATTERNETZE

**VIROR-Teleseminar WS
2001/2002**

Ubiquitous Computing

**Christian Lochert
Universität Mannheim**

14.01.2002

Übersicht

1. Technische Grundlagen
2. Bildung von Scatternetzen
3. Routing in Scatternetzen
4. Kritische Würdigung

1. Technische Grundlagen

- ◆ Weltweit freies und unlizenziertes ISM-Band von 2,4 bis 2,5 GHz
- ◆ Frequency Hopping Spread Spectrum
- ◆ Time Division Duplex

Technische Grundlagen

- ◆ Aufteilung des Bandes in 79 Kanäle mit jeweils 1 MHz Größe
- ◆ 1600 hops pro Sekunde
→ Zeitschlitzgröße von $625 \mu\text{s}$

Ergänzung durch:

- ◆ ARQ (Automatic Repeat Request)
- ◆ CRC (Cyclic Redundancy Check)
- ◆ FEC (Forward Error Correction)

Technische Grundlagen

- ◆ Aufteilung der Knoten in Master/Slave
- ◆ Kommunikation **nur** zwischen Master-Slave
- ◆ Statische Aufteilung der Zeitschlitz zum Versenden von Paketen
 - Master in geraden Zeitschlitz
 - Slave in ungeraden Zeitschlitz

Technische Grundlagen

- ◆ Master bestimmt Sprungsequenz der Slaves
- ◆ Sendeleistung: 1 – 100 mW
- ◆ Modulationsverfahren: GFSK
- ◆ Reichweite: 10 m

Kommunikation zwischen Endgeräten

- ◆ Verschiedene Zustände
 - Standby
 - Verbindungsaufbauzustand
 - INQUIRY/ INQUIRY-SCAN
 - PAGE/ PAGE-SCAN
 - Aktiv
 - Stromsparmodus
 - SNIFF
 - HOLD
 - PARK

Bildung von Teilnetzen – Beschränkungen

Es gelten folgende Bedingungen:

- ◆ **Knotentypbeschränkung**

Jeder Knoten ist entweder Master oder Slave

- ◆ **Gradbeschränkung**

Ein Master kann maximal sieben Slaves versorgen

- ◆ **Verbindungsbeschränkung**

Zwei Slaves können nicht *direkt* miteinander kommunizieren

Bildung von Teilnetzen – INQUIRY

- ◆ Knoten A versendet mittels Broadcast INQUIRY-Paket, welches Frage nach gewünschtem Dienst enthält
- ◆ Knoten B in INQUIRY-SCAN Zustand empfängt INQUIRY Nachricht und stellt gewünschten Dienst bereit
- ◆ Knoten B sendet an Knoten A nach zufälliger Backoffzeit seine weltweit eindeutige BD_ADDR (48 Bit) zurück

Bildung von Teilnetzen – PAGE

- ◆ Knoten A sendet PAGE Nachricht an Knoten B, da BD_ADDR bekannt
- ◆ Knoten B in PAGE-SCAN Zustand empfängt PAGE Nachricht von Knoten A
- ◆ Knoten B sendet Bestätigung an Knoten A

Bildung von Teilnetzen

- ◆ Knoten A wird Master
- ◆ Knoten B wird Slave und erhält von Knoten A 3-bit lange Mitgliedsadresse AM_ADDR
- ◆ Übertragung der Pakete
 - SCO – synchron und verbindungsorientiert
 - ACL – asynchron und verbindungslos

SCO

- ◆ Punkt zu Punkt Verbindung
- ◆ Keine Bestätigungen
- ◆ Feste Bandbreite
- ◆ Symmetrische Datenrate von 64 kbit/s
- ◆ Keine Prüfsummen
- ◆ FEC möglich

ACL

- ◆ Teilen sich restliche Bandbreite auf
- ◆ Bestätigungspakete
- ◆ Symmetrisch mit max. 433,9 kbit/s
- ◆ Asymmetrisch mit max. 723,2 bzw. 57,6 kbit/s
- ◆ CRC und FEC möglich

Weitere Pakettypen

- ◆ NULL

Bestätigung und Flusskontrolle 126 bit lang

- ◆ POLL

Aufforderung eines Slaves zu antworten

POLL-Pakete werden bestätigt

Stromsparmodus

- ◆ SNIFF

Der Slave wacht zu Beginn jeder vereinbarten Periode auf, um angesprochen werden zu können

- ◆ HOLD

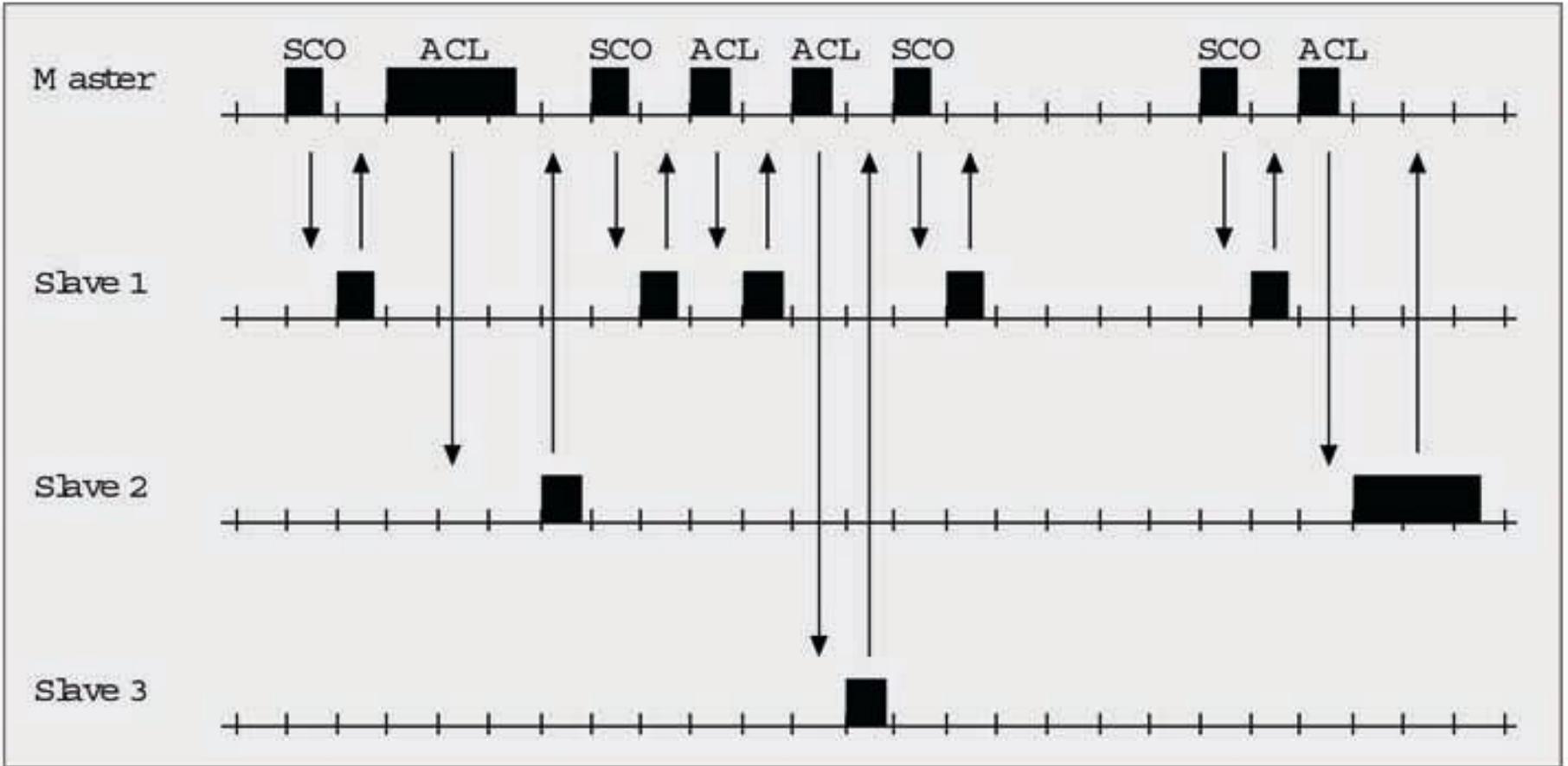
Der Slave trennt für eine vereinbarte Zeitspanne seinen ACL Link

- ◆ PARK

Im Park-Modus gibt der Slave seine AM_ADDR ab und erhält eine eindeutige PM_ADDR und eine AR_ADDR.

Mit der PM_ADDR kann der Master den Slave wieder aufwecken

Mit der AR_ADDR kann der Slave versuchen, sich wieder mit dem Master zu verbinden



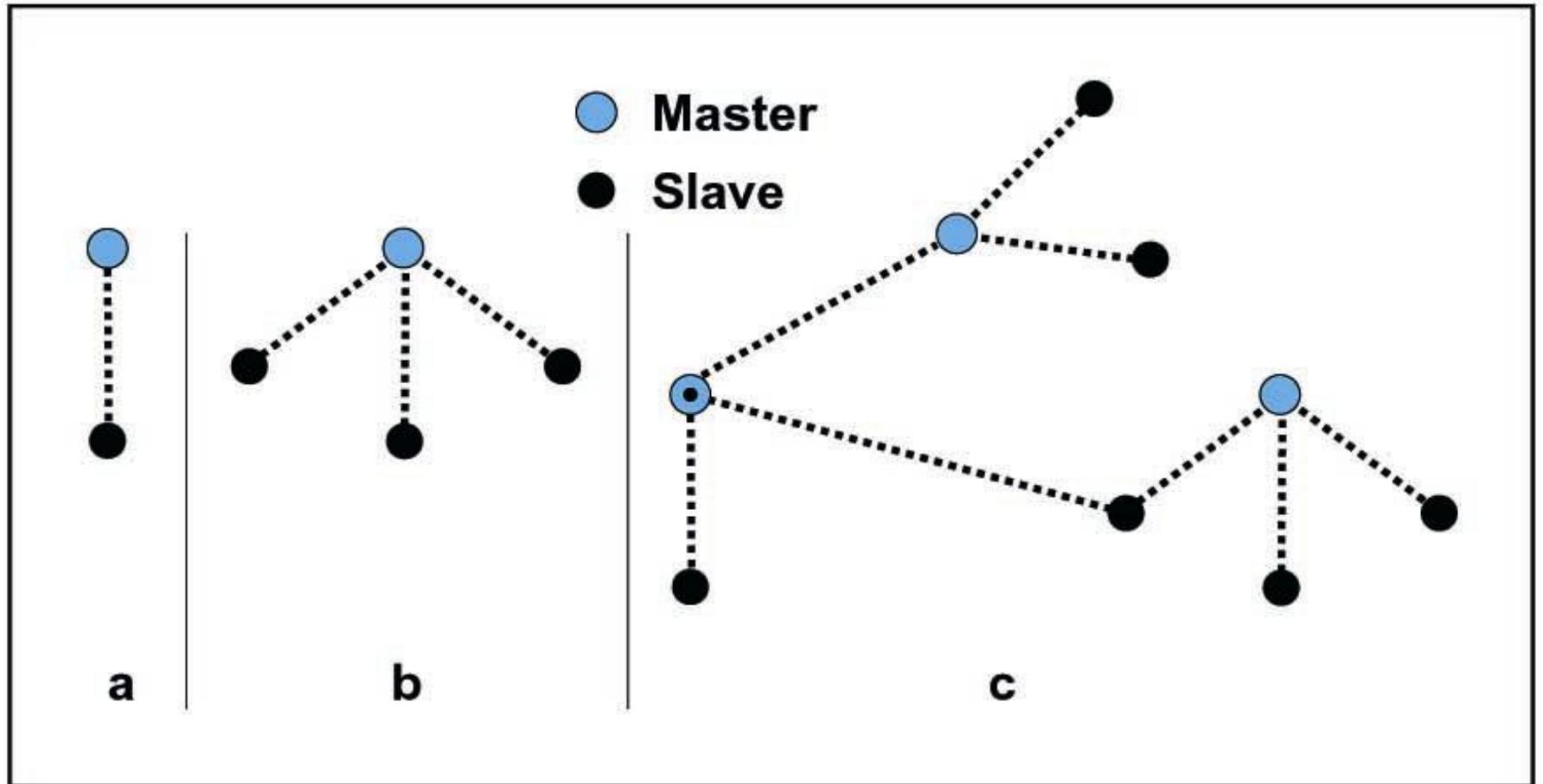


Figure 1.2: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c).

2. Bildung von Scatternetzen

- 2.1 Bluetooth Topology Construction Protocol (BTCP)
- 2.2 Bildung von Scatternetzen mittels Baumstruktur (TSF)
- 2.3 Bildung von Scatternetzen durch Benutzung des SNIFF-Modes

2.1 Bluetooth Topology Construction Protocol (BTCP)

Vorbedingungen:

- ◆ Ein Brückenknoten kann mit maximal zwei Piconetzen verbunden sein
- ◆ Ein Scatternetz sollte aus der minimalen Anzahl an Piconetzen bestehen
- ◆ Das resultierende Scatternetz sollte vollständig verbunden sein
- ◆ Zwei Piconetze teilen sich nur eine Brücke
- ◆ Alle Knoten befinden sich in erreichbarer Umgebung

BTCP

- ◆ Paper von Salonidis, Bhagwat, Tassiulas; Infocom 2001
- ◆ 3 Phasen-Konzept
- ◆ Zentraler Knoten übernimmt gesamten Verbindungsaufbau

BTCP – Phase 1

Bestimmung eines zentralen Knotens durch

- ◆ Initialisierung $VOTES := 1$ in allen Knoten
- ◆ Initialisierung $ALT_TIMEOUT := 2527,223$ ms
- ◆ Alternierung aller Knoten $INQUIRY/ INQUIRY-SCAN$
- ◆ Bei Verbindung Knoten x Knoten y:
 - $VOTES(x) > VOTES(y) \Rightarrow x$ Gewinner
 - $VOTES(x) = VOTES(y) \Rightarrow$ Knoten mit höherer BD_ADDR gewinnt

BTCP – Phase 1

Sei oBdA Knoten x Gewinner:

- ◆ $VOTES(x) += VOTES(y)$
- ◆ Rücksetzen von `ALT_TIMEOUT`
- ◆ Verlierer übergibt alle bisher erhaltenen Adressen und geht in `PAGE-SCAN` Zustand

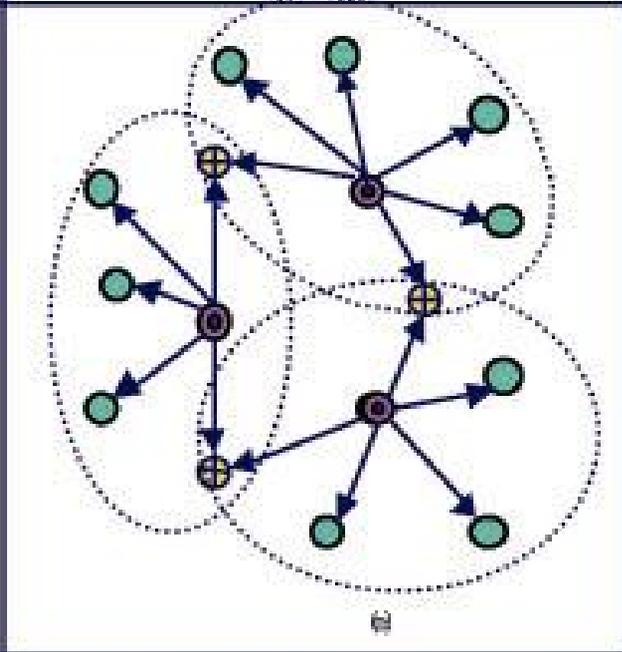
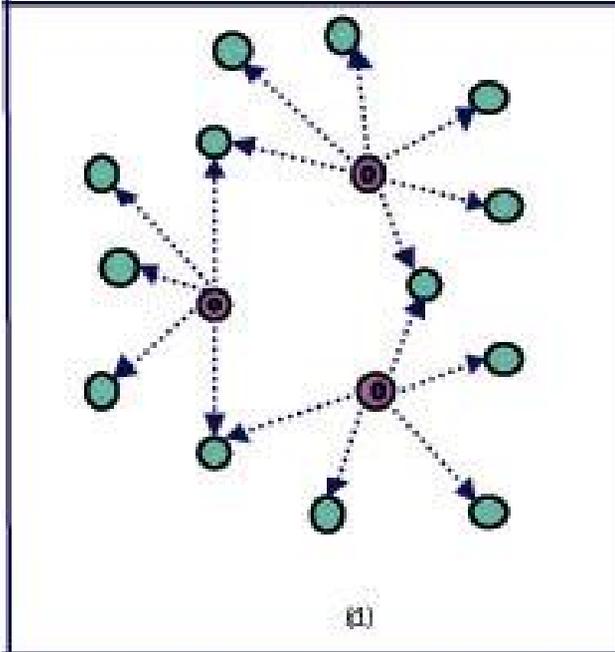
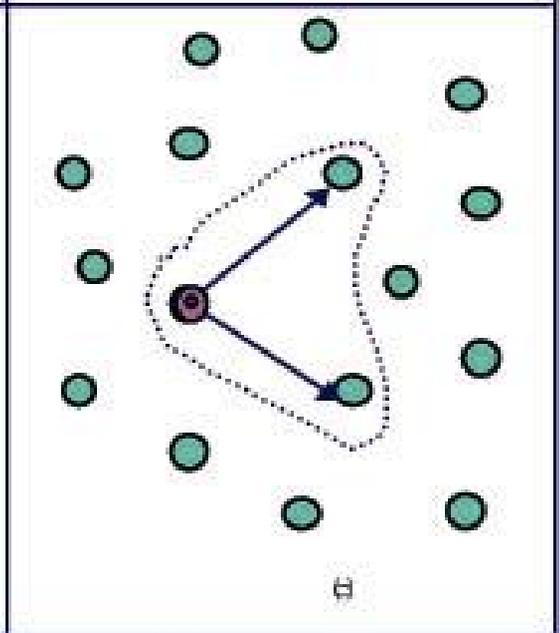
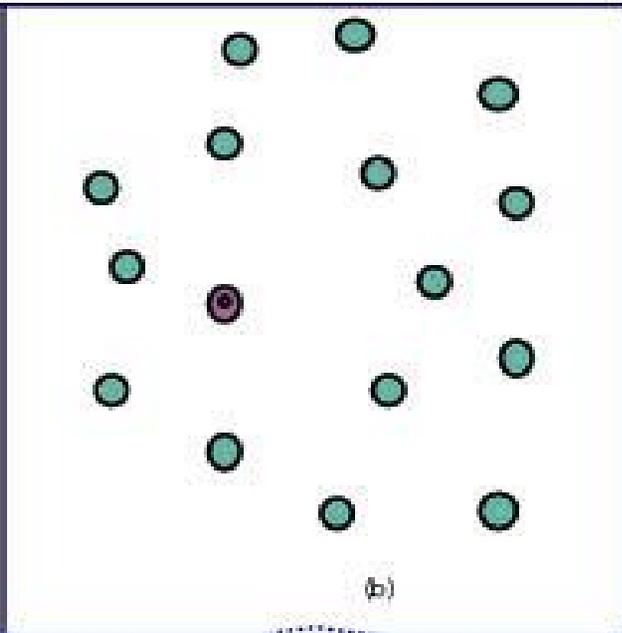
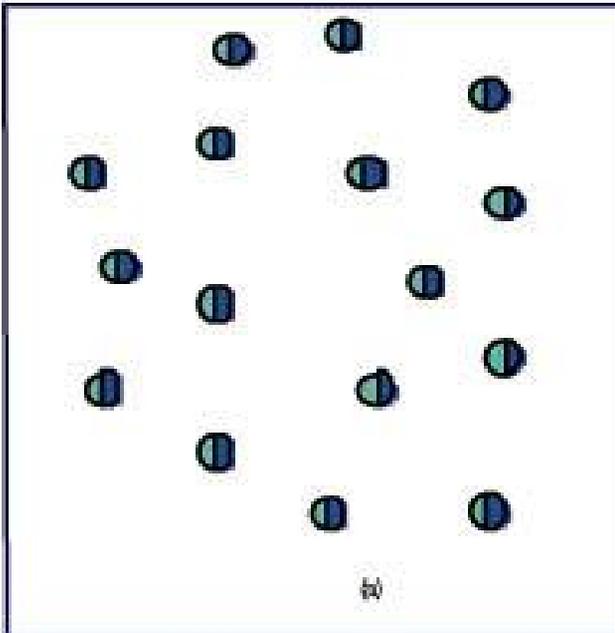
- ◆ $ALT_TIMEOUT(x) = 0 \Rightarrow$ Knoten x ist Zentraler Knoten

BTCP – Phase 2

- ◆ Zentraler Knoten weiß, wie viele Knoten Scatternetz formen wollen ($|\text{Knoten}| > 7$)
- ◆ Anzahl aller beteiligten Knoten aufgrund Berechnungsmethode der lokalen Master maximal 36
- ◆ Brückenknoten und lokale Master werden von zentralem Knoten ausgewählt und benachrichtigt
- ◆ Übermittlung von Verbindungslisten an lokale Master der Piconetze

BTCP – Phase 3

- ◆ Lokale Master verbinden sich aufgrund der erhaltenen Verbindungslisten mit anderen Knoten
 - ◆ Brückenknoten warten auf Verbindung mit beiden Master und versenden **CONNECTED** an beide Master
 - ◆ Wenn alle Master alle **CONNECTED**-Nachrichten erhalten haben, terminiert Algorithmus
- => Scatternetz hergestellt



	A Remating node
	Coordinator/Master
	Node in PAGE SCAN /Slave
	Bridge Node
	B is a slave of A

2.2 Bildung von Scatternetzen mittels Baumstruktur (TSF)

Gewünschte Leistungsmerkmale:

- ◆ Vermeidung von Datenoverhead
- ◆ Vereinfachung der internen Kommunikation
- ◆ Verbindungswunsch aller Knoten in naher Zeitdistanz gewährleistet
- ◆ Einzelne Knoten können in bestehendem Netz einfach integriert werden
- ◆ Erhaltung der Baumstruktur bei allen Vorgängen

Algorithmus

Vorbedingungen:

- ◆ Knoten entweder Teil eines vorhandenen Baums oder
- ◆ Knoten frei
- ◆ 2 Zustände:
 - FORM – zum Aufbau eines Bluetooth-Link
 - FORM:INQUIRY
 - FORM:INQUIRY-SCAN
 - COMM – zur Kommunikation innerhalb eines Link

Algorithmus

- ◆ Die Verweildauer in einem Zustand wird durch eine Zufallsfunktion bestimmt

Diese ist abhängig:

- ◆ Von der Initialisierungszeit $T_{\text{inq}} = 2T_{\text{sync}} + T_{\text{b0}}$
- ◆ Einem festen Parameter D
- ◆ Innerhalb des COMM-Zustandes von der „Beschäftigkeit“ f_{comm} eines Knoten

Zufallsfunktion f_{comm}

f_{comm} ist Funktion über das Alter eines Knoten seit Eintritt in das Scatternetz und über den Anteil an Kindern in dem Baum

$$f_{\text{comm}} = \begin{cases} 0 & : \text{freier Knoten} \\ d & : \text{nicht freier Knoten \& Alter} < \varepsilon \\ Ad & : \text{nicht freier Knoten \& Alter} > \varepsilon, A > 1 \end{cases}$$

Verhinderung von Schleifen

- ◆ Freie Knoten können nur mit freien Knoten oder Nicht-Wurzelknoten Links herstellen. Dabei werden sie automatisch Slave.
- ◆ Wurzelknoten von Bäumen mit mehr als einem Knoten können nur Verbindungen zu anderen Wurzelknoten herstellen.
- ◆ Nicht-Wurzelknoten versuchen nicht größere Bäume mit nicht freien Knoten zu bilden.

Herstellung einer Verbindung

- ◆ Jeder Knoten wechselt immer zu Beginn seinen Status, bspw. von INQUIRY nach INQUIRY-SCAN
- ◆ Empfängt dieser Knoten eine INQUIRY-Nachricht, wird ein Link hergestellt.
- ◆ Der Master wird Wurzelknoten
- ◆ Der Slave wird Blattknoten bzw. der gesamte Teilbaum wird dem Wurzelknoten unterstellt.
- ◆ Wurzelknoten, die bereits sieben Slaves besitzen, wechseln mit einem Slave die Rolle.

Trennen einer bestehenden Verbindung

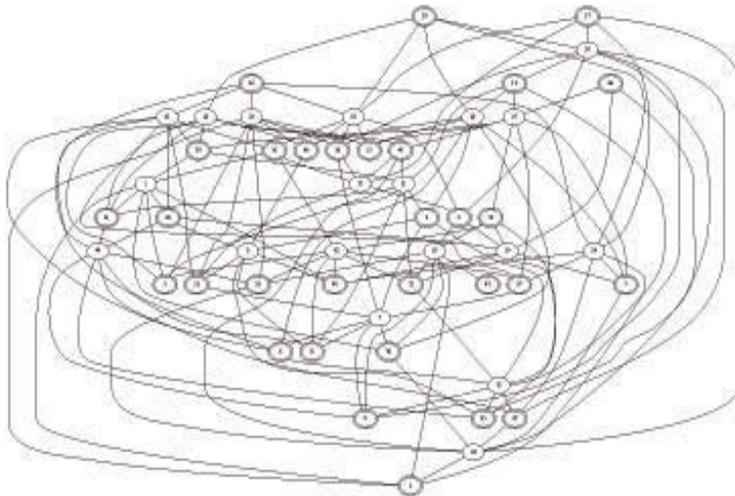
Master verliert Verbindung zu einem Slave:

- ◆ Master überprüft, ob er nun freier Knoten ist

Slave verliert Verbindung zu einem Master:

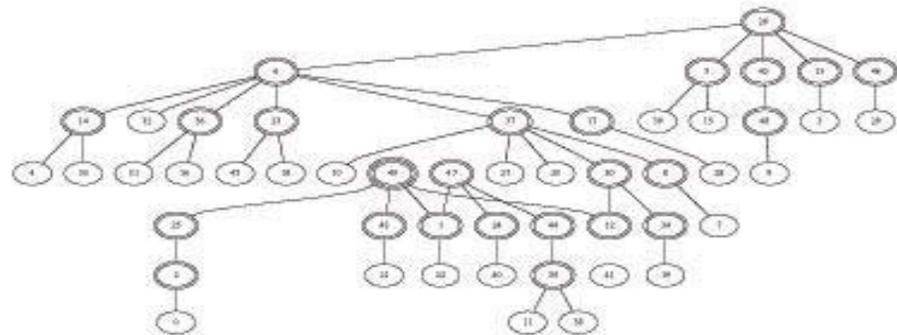
- ◆ Slave setzt $\text{Alter} = 0$
- ◆ Blattknoten werden freie Knoten
- ◆ Innere Knoten werden Wurzelknoten eines Teilbaums

Simulationsergebnisse



Scatternetz PROB

Scatternetz TSF



Simulationsergebnisse

- + Verbindungsaufbau bei TSF (3 s) besser als bei PROB
- Ab 40 Knoten aber benötigt TSF (70 s) doppelte Zeit wie PROB
- Durchschnittliche Paketverzögerung bei TSF in etwa identisch mit PROB oder etwas besser
- Resultate stark abhängig von Routingalgorithmen

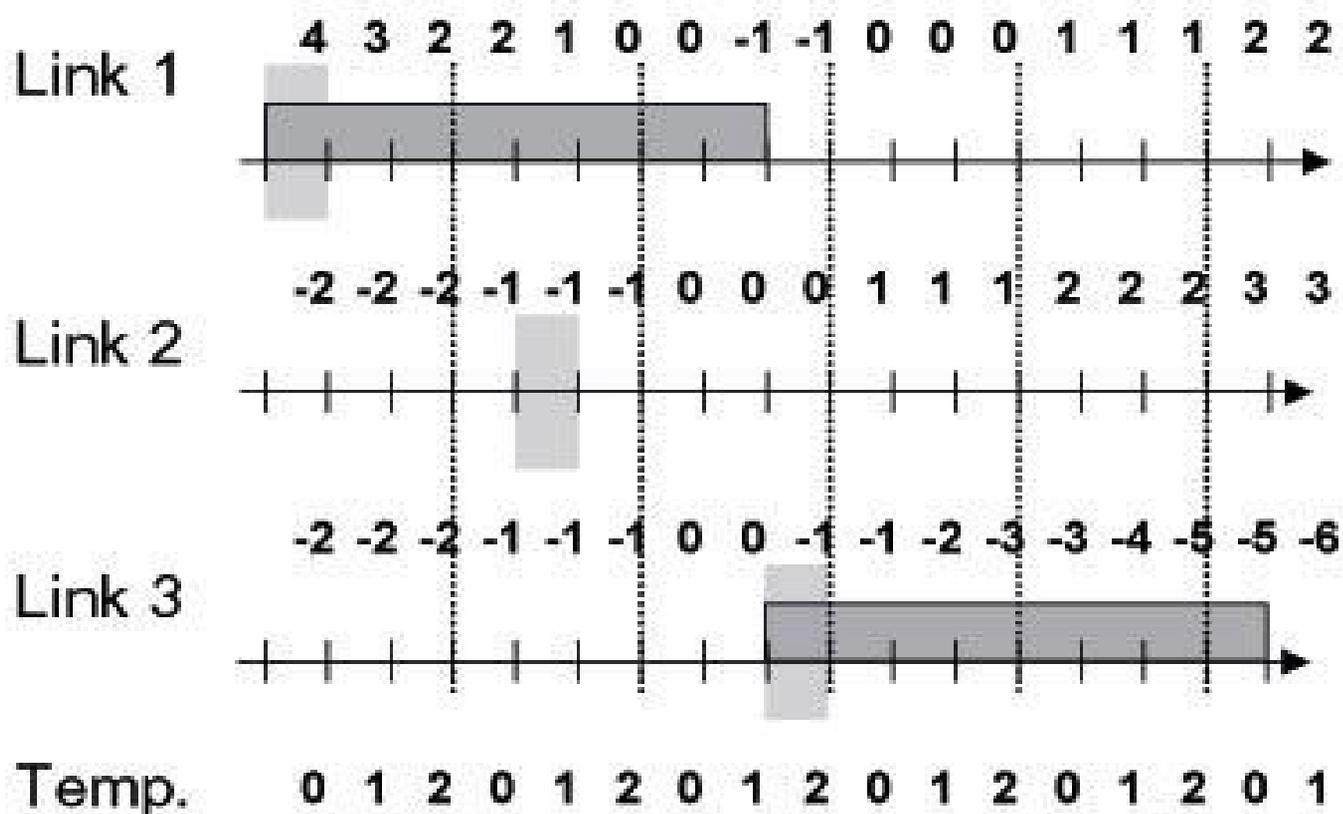
2.3 Bildung von Scatternetzen durch Benutzung des SNIFF-Modus

- ◆ Steigerung der Bandbreite durch gleichzeitige Nutzung des Mediums (möglich durch Frequency Hopping)
- ◆ Routerknoten halten eine Verbindung aktiv während die anderen Verbindungen in den sniff-Zustand wechseln

Kreditbasierte Scatternetz-Planung

- ◆ Presence points bilden den Startpunkt der Inter-Piconetz-Kommunikation
- ◆ Sniff slots repräsentiert durch presence points
- ◆ Knoten müssen zur Kommunikation auf den presence point des eigenen Piconetzes warten

Activity (Transmit or Listen)
 Sniff Slot



Prioritäten generieren Fairness

- ◆ Jeder verbundene Knoten erhält eine Priorität
- ◆ Zusätzliche Priorität einer temporären Liste, um konstante Anzahl an Krediten zu gewährleisten
- ◆ Wenn ein Knoten Pakete sendet oder empfängt wird pro slot ein Kredit von seiner Priorität abgezogen und der temporären Liste hinzuaddiert
- ◆ Sobald ein Knoten eine höhere Anzahl an Krediten hält, wird die Verbindung unterbrochen und dieser Knoten darf senden
- ◆ Wenn die temporäre Liste $n-1$ Kredite enthält, werden diese umverteilt

Probleme durch Wartezeiten

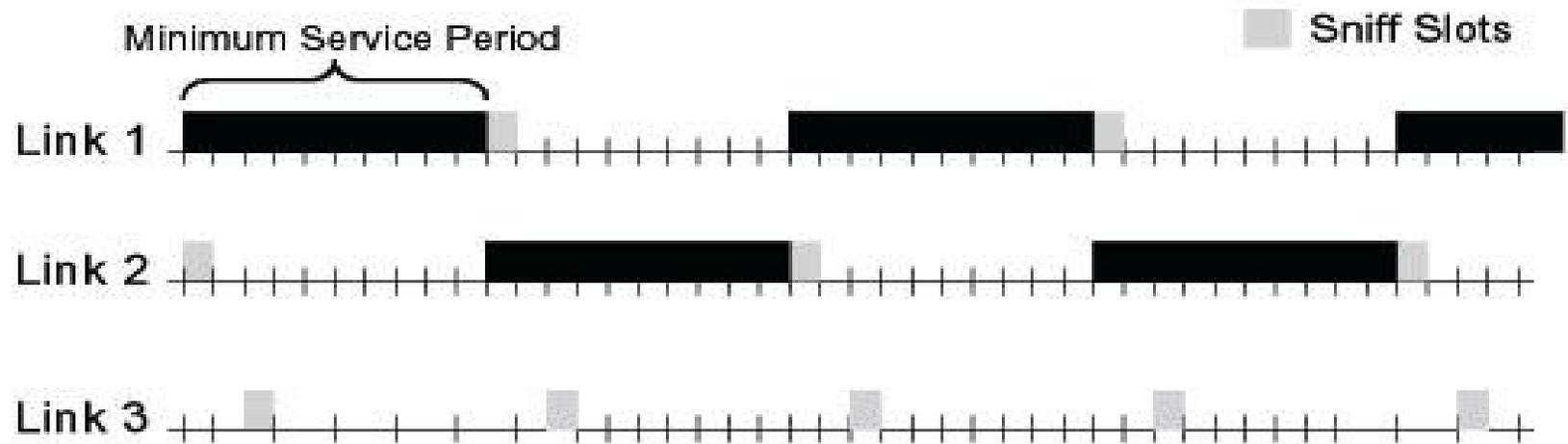
- ◆ Durch das alleinige System der Prioritäten kommt es zu häufigen Verbindungswechseln
- ◆ Abhilfe schafft Mindestverbindungsdauer

⇒ **PROBLEM: starvation**

⇒ Sniff slots können während einer bestehenden Verbindung auftreten

⇒ Einige Knoten dürfen nie senden, obwohl ihre Priorität dies fordern würde

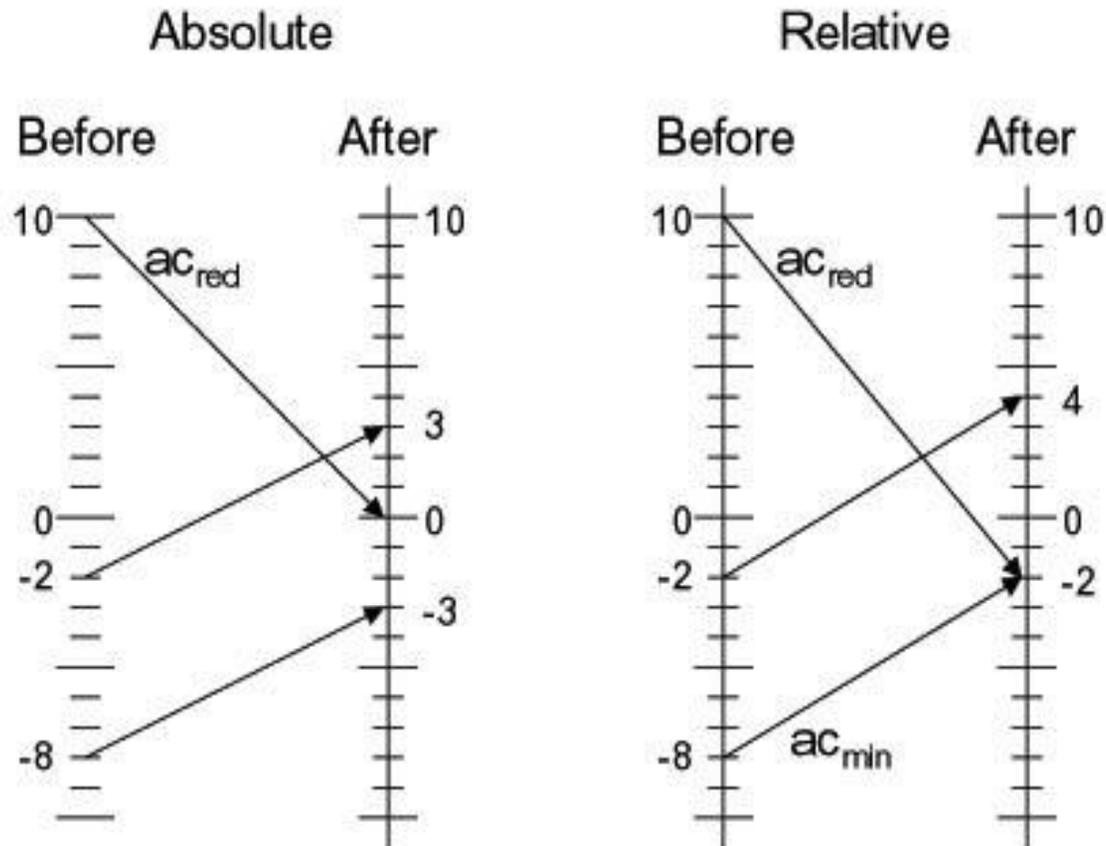
Starvation



Lösung des starvation-Problems

- ◆ Unterschiedliche Zuteilung der Startkredite
- ◆ Festlegung einer Verbindungswechselschwelle $N_{\text{switch_th}}$ als untere Grenze bzw. $2 \cdot N_{\text{switch_th}}$ als obere Grenze
- ◆ Umverteilung der Kredite durch Betrachtung der Differenz zwischen den Knoten

Redistribution der Kredite



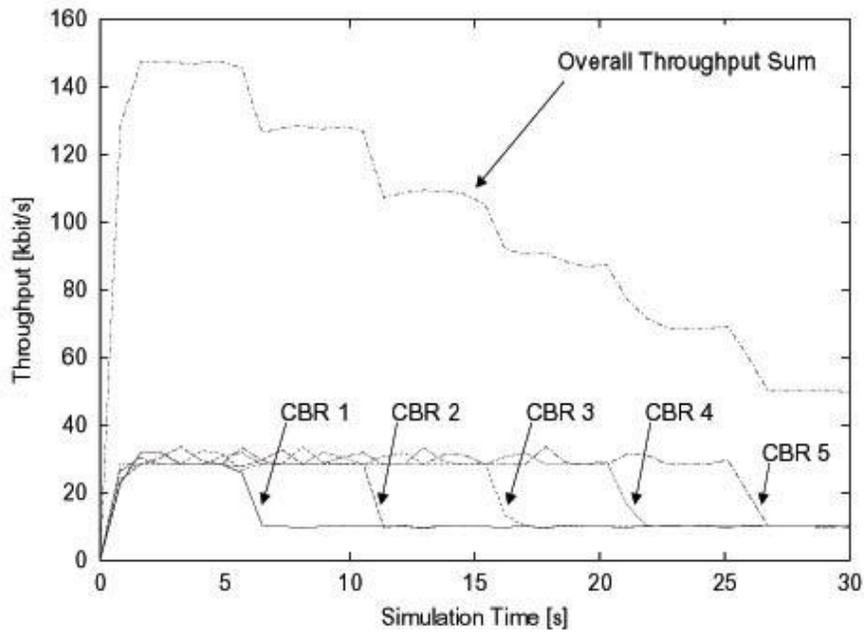
Simulationsergebnisse

- ◆ Simulation eines Scatternetzes mit 6 Knoten
 - ◆ 1 Knoten als Hub
 - ◆ Maximaler Datendurchsatz von $800 \text{ 1/s} * 23 \text{ Byte} * 8 \text{ bit/Byte} = 147,2 \text{ kbit/s}$
- ⇒ Jeder äußere Knoten sendet mit $29,44 \text{ kbit/s}$

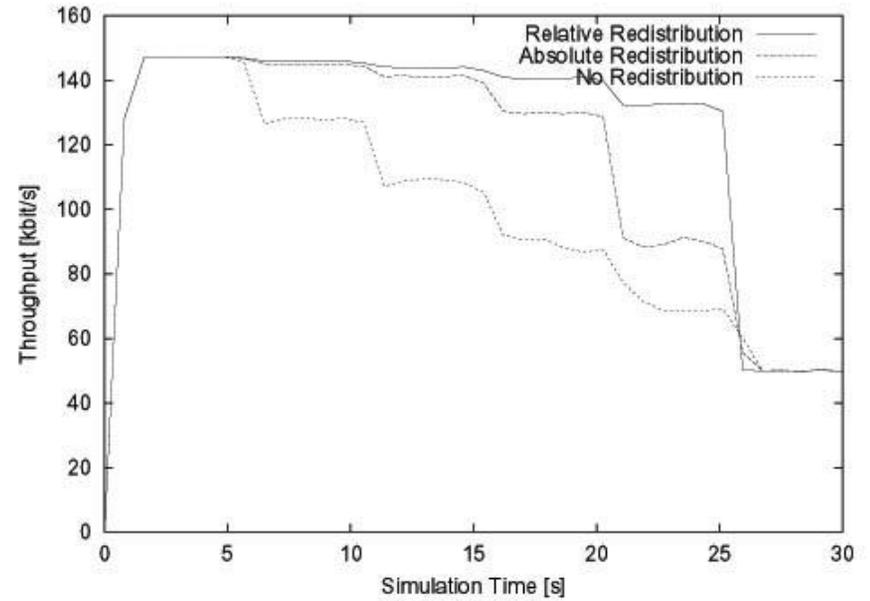
Simulationsergebnisse

Beenden einzelne Knoten ihre Kommunikation

- ◆ Keine Umverteilung der Kredite:
 - Freie Kapazität wird nicht auf die restlichen Knoten verteilt
- ◆ Umverteilung der Kredite:
 - Freie Kapazität wird auf die restlichen Knoten verteilt



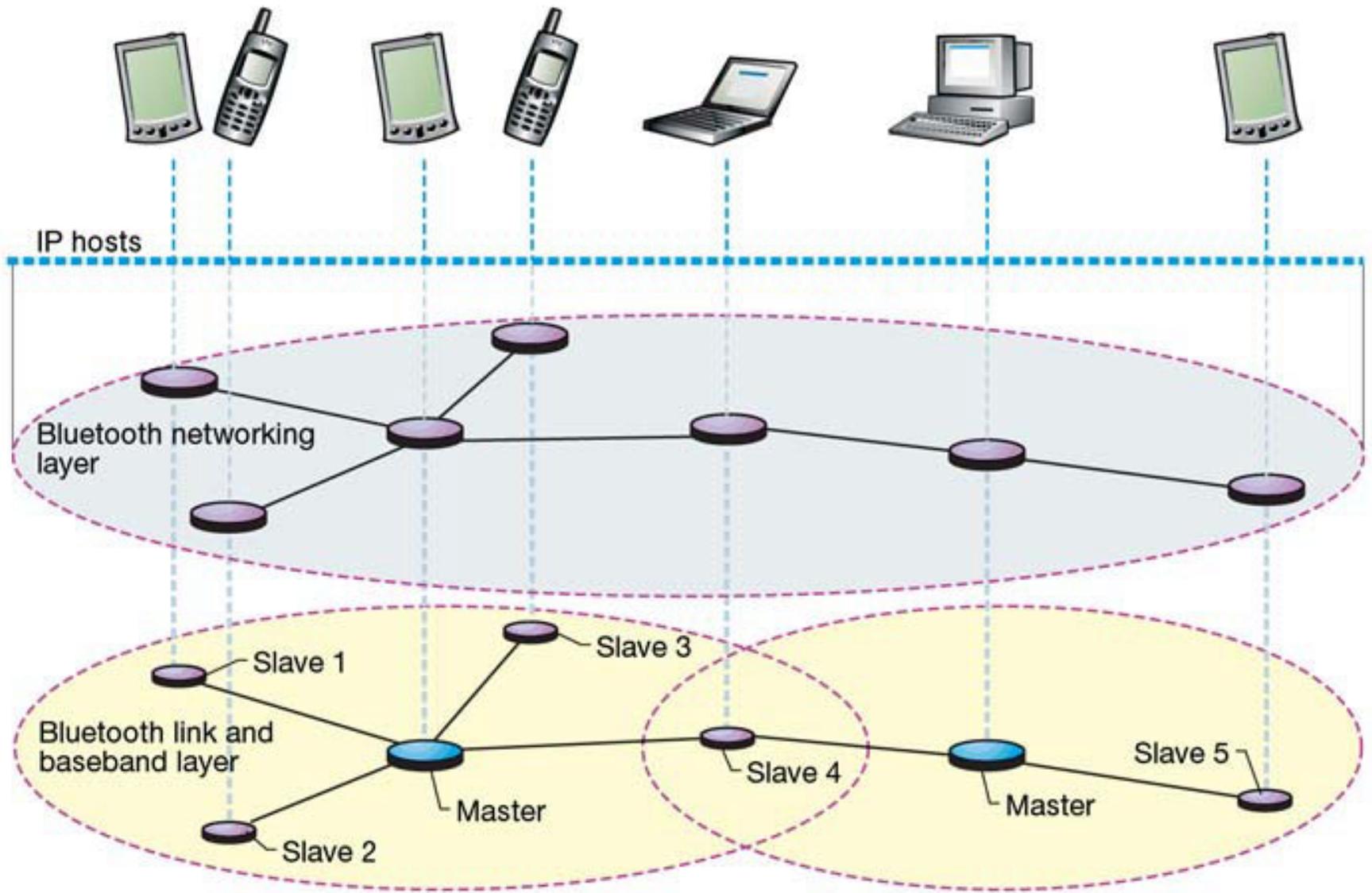
Datendurchsatz ohne Umverteilung



Summe des gesamten Datendurchsatzes

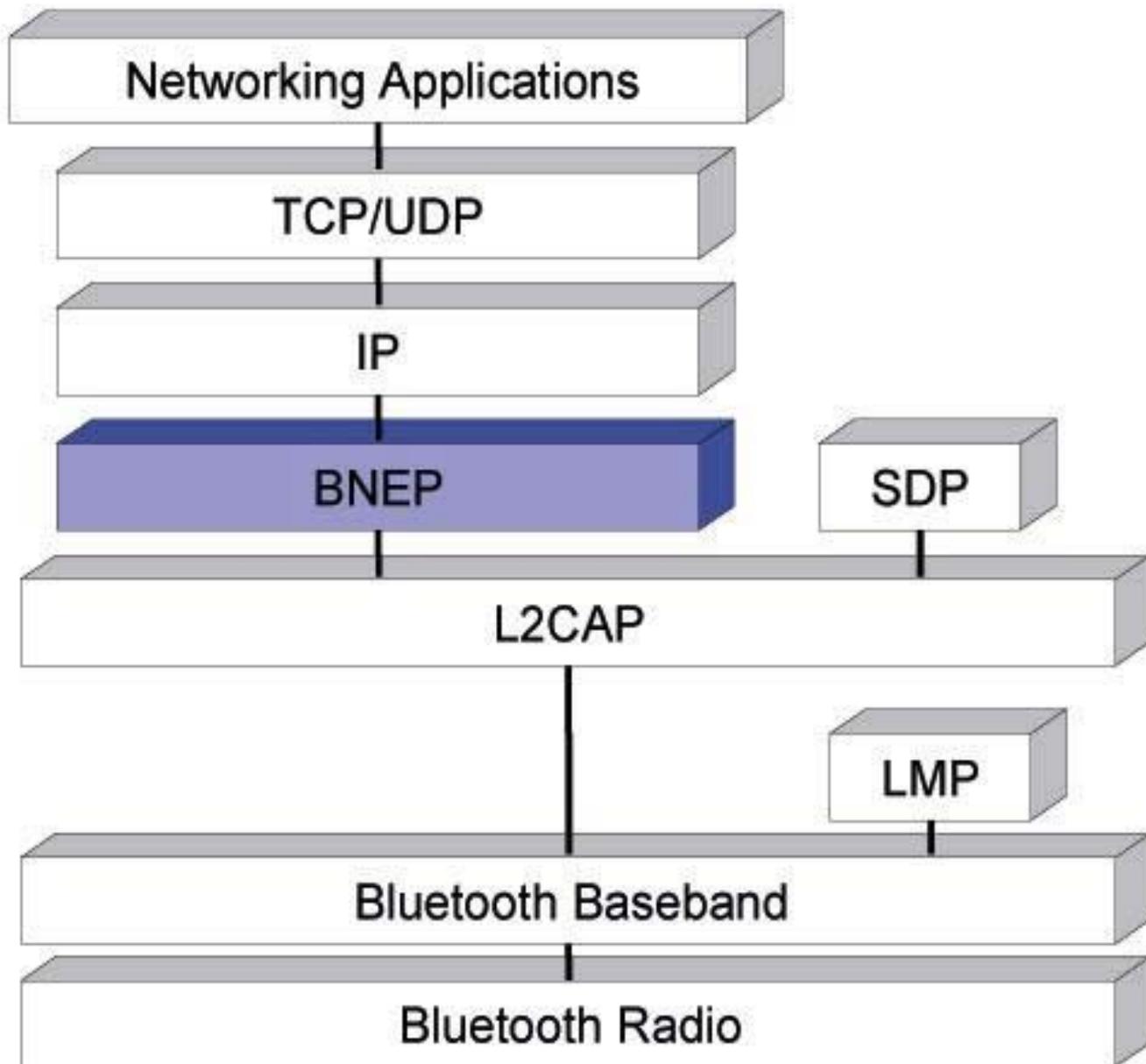
3. Routing in Scatternetzen/ Inter-Piconetzen

- 3.1 Unterschiede zu IP-Routing
- 3.2 Bluetooth Network Encapsulation Protocol (BNEP)
- 3.3 Gesamtheitliche Optimierung des Bluetooth Stack



3.2 Bluetooth Network Encapsulation Protocol (BNEP)

- ◆ BNEP stellt Schnittstelle ähnlich der des Ethernet zu IP dar
 - ◆ BNEP wird auf das Logical Link and Control Adaption Protocol (L2CAP) aufgesetzt
 - ◆ BNEP benutzt BD_ADDR zur Adressierung innerhalb des Scatternetzes
- ⇒ Unabhängiger Dienst zu IP
- ◆ Alle bekannten Routingalgorithmen für mobile ad hoc Netzwerke können verwandt werden



3.3 Gesamtheitliche Optimierung des Bluetooth Stack

- ◆ Verbindungen sollten nur aufrecht erhalten werden, wenn Pakete versandt werden sollen
 - ◆ Untere Schichten müssen nur dann arbeiten, wenn höhere Schichten diese anfordern
 - ◆ Bluetooth-Geräte werden nicht nach ihrer IP-Adresse angesprochen, sondern nach dem Dienst, den sie zur Verfügung stellen
- ⇒ Einführung eines gesamtheitlichen Stacks

Kritik

- ◆ Gesamtlayer verhält sich nachteilig gegenüber Modularisierung
- ◆ Simulationsresultate zeigen jedoch:
 - Ein Stack aus IP und Service Discovery (SD)
 - Ein Stack aus IP, SD und VerbindungsherstellungHaben Vorteile gegenüber Schichtenstack:

Die Anzahl der versandten Nachrichten ist hier mehr als doppelt so hoch.

4. Kritische Würdigung

- ◆ Alle dargestellten Algorithmen nur mittels Simulation getestet
- ◆ Algorithmen schlecht vergleichbar, da
 - Unterschiedliche Voraussetzungen
 - Unterschiedliche Anzahl an Knoten
 - Unterschiedliche Simulationsprogramme
Hauptsächlich `ns-2` und `simjava`

Kritische Würdigung

- + Bedingt geeignet für ubiquitäre Umgebungen
- + Gut geeignet für Konferenzsituationen
- Nur geringe Beachtung der Mobilität
- Keine Lösungen für sichere Kommunikation
- Frage der Energie der Geräte