Digitale Kommunikation und Internetdienste 1

Wintersemester 2004/2005 – Teil 5

Belegnummer Vorlesung: 39 30 02

Übungen: 39 30 05

Jan E. Hennig

AG Rechnernetze und Verteilte Systeme (RVS)
Technische Fakultät
Universität Bielefeld

`jhennig@rvs.uni-bielefeld.de`

basierend auf den Arbeiten von Michael Blume, Heiko Holtkamp, Marcel Holtmann und I Made Wiryana

Übersicht

- ARP-Protokoll
- Token Ring (IEEE 802.5)
- Wireless (IEEE 802.11)

Erinnerung

- Ethernet: CSMA/CD, bin. exp. Backoff-Algorithmus
- 48-bit MAC-Adressen sind phys. Anschluß zugeordnet
- OSI-Schicht 1: Repeater, Hub
- OSI-Schicht 2: Bridge, Switch
- OSI-Schicht 3: Router
- OSI-Schicht 7: Gateway

Adressen

- unterschiedliche Adressen auf verschiedenen Ebenen
- z.B. Ethernet-MAC-Adressen als „physical address”
  Notation: 00:A0:D2:14:E7:B2
- z.B. IP-Adressen als „logical address”
  Notation: 129.70.123.245 (IPv4)
- Problem: Zuordnung oder Übersetzung

Adressintegration

- Physische Adresse in logische Adresse integrieren
- z.B. physische Adresse: 00010001 00101001 (binär)
- entspricht dezimal 33 und 81
- Integration: 129.70.33.81
- funktioniert nicht bei zu langen physischen Adressen
- kann z.B. bei IPv6 zur Autokonfiguration verwendet werden

Zentrale Übersetzungstabelle

- Zuordnungen als Tabelle
- Zuweisung logische → physische Adresse
- z.B. durch Systemadministrator erstellt
- Kopien auf jedem Host abgelegt
- Nachschlagen sehr einfach
- manuelles Anlegen unpraktisch

ARP (1)

- Address Resolution Protocol (ARP) – RFC 826
- Zuordnungstabelle logische → physische Adresse
- wird *ARP cache* oder *ARP table* genannt
- Tabellen werden dynamisch aufgebaut
- Host lernt neue Zuordnungen bei Bedarf
- Einträge altern und laufen nach bestimmter Zeit aus (z.B. 15min)

ARP (2)

- ARP nutzt die Möglichkeit eines Broadcasts aus
- Host A möchte ein IP-Datengramm an Host B senden
- aus Struktur der IP-Adresse ist erkennbar, daß Host B im gleichen Netzwerk ist
- prüfen, ob bereits Eintrag in ARP table vorhanden
- falls nicht: Aussenden einer *ARP-Anfrage (ARP query)*

ARP (3)

- ARP-Anfrage enthält Ziel-IP-Adresse
- jeder Host im Netzwerk erhält die Anfrage
- und prüft, ob es sich um seine IP-Adresse handelt
- falls ja, sendet er eine ARP-Antwort
- diese enthält die physische Adresse von Host B
- Host A fügt damit einen Eintrag seiner ARP-Tabelle hinzu

ARP (4)

- ARP-Anfrage enthält auch physische und logische Adressen des Senders
- damit kann beim Broadcast jeder weitere Host seine Zuordnungen anpassen
- z.B. den Verfall-Zähler zurücksetzen
- Ziel-Rechner nimmt Zuordnung neu auf
- denn hohe Wahrscheinlichkeit, daß Sender bald Kommunikationspartner

ARP-Paketformat

- *Hardware Type*: Art des physikalischen Netzwerks, z.B. Ethernet
- *Protocol Type*: Protokollnummer des höheren Protokolls, z.B. IP
- *HLen, PLen*: Länge von Hardware und Prokokolladressen
- *Operation*: Anfrage oder Antwort
- *source* and *target hardware* (Ethernet) and *protocol addresses* (IP)

Abseits vom Ethernet

- Ist Ethernet die einzige Technologie auf dem Network Layer?
- Gibt es andere LAN-Technologien?
- Kann man gar ohne Kollisionen auskommen?

Token Ring (1)

- Wie bei Ethernet gibt es viele Token-Ring-„Standards”
- wichtig, da verbreitet: IBM Token Ring, IEEE 802.5
- Grundprinzip: Knoten zu einem Ring zusammengeschlossen
- Daten bewegen sich in eine Richtung den Ring entlang
- jeder Knoten nimmt Daten vom Vorgänger (upstream neighbour) an
- und sendet sie an seinen Nachfolger (downstream neighbour) weiter

Token Ring (2)

- Medium wird als gemeinsam genutztes Medium verstanden
- feste Regeln, wann ein Knoten senden darf bzw. muß
- jeder Knoten bekommt jedes Paket zu sehen
- adressierter Zielknoten speichert das Datenpaket und verarbeitet es weiter
- „Token”: Spielstein oder -marke
- tatsächlich: eine bestimmte Sequenz an Bits

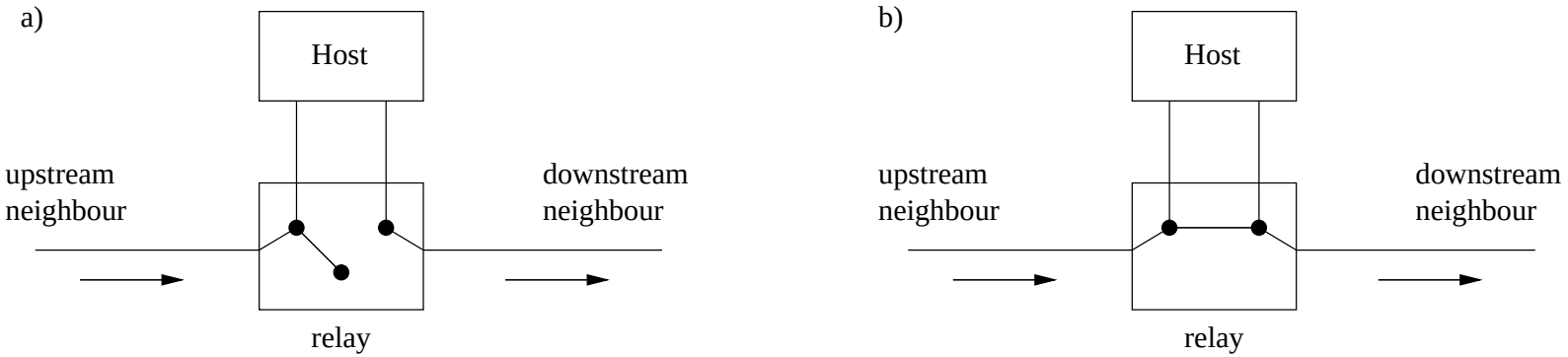
Token Ring (3)

- Token wird nach und nach herumgereicht
- Knoten, der das Token besitzt darf seine Daten senden
- er nimmt das Token für eine gewisse Zeit aus dem Netz und speichert es zwischen
- fügt seine Daten-Frames ein
- anschließend sendet er das Token weiter
- andere Knoten leiten Daten weiter, auch das Token, wenn sie nichts senden möchten

Token Ring (4)

- Daten wandern so den Ring entlang
- ein sendender Knoten empfängt sein Paket wieder
- weiß dann, daß die Daten von jedem Knoten gesehen wurden
- zwar keine Kollisionen
- aber Problem: hoher Verwaltungsaufwand

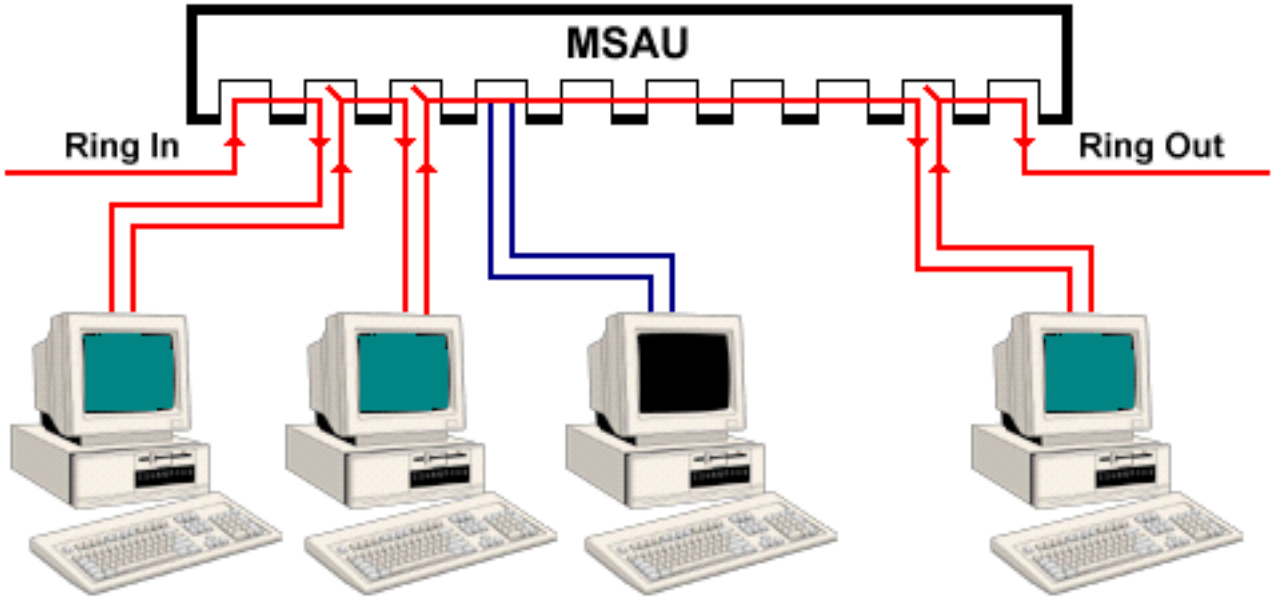
Überbrückung



- Ring muß vollständig bleiben
- daher Relais: a) Host im Betrieb, b) Host außer Betrieb
- ohne Strom schließt Relais automatisch
- Host wird dann umgangen

MSAU

- Erweiterung: *multi station access unit (MSAU)*
- ermöglicht eine Art Sterntopologie



Details

- IBM Token Ring schreibt MSAUs vor, IEEE 802.5 nicht
- Datenraten: 4 oder 16 Mbps
- Manchester-Kodierung
- IBM: 260 Stationen pro Ring
- IEEE 802.5: 250 Stationen pro Ring

Media Access Control

- Knoten besitzen Eingang, Ausgang und dazwischen Speicher
- IEEE 802.5 fordert 1 Bit Zwischenspeicher
- Token: 24 Bits
- somit wären minimal 24 Stationen nötig (Übertragungszeit vernachlässigt)
- Lösung: Monitor-Station, die Verzögerungen einfügt

Token

- *Beschlagnahmen (seizing)* des Tokens erfordert Ändern eines Bits im zweiten Byte
- bereits gesendete Bits bilden Präambel des eingefügten Datenpakets
- Besitzt eine Station das Token, darf sie ein oder mehrere Pakete senden
- wieviele hängt von der Sendedauer ab:
- maximale *token holding time (THT)* darf nicht überschritten werden

Zugriff

- THT ist Abwägung zwischen Netzauslastung und Wartezeit anderer Knoten
- von THT abgeleitet: *token rotation time (TRT)*
- TRT $\leq$ Anzahl aktiver Knoten $\times$ THT + Ring-Latenz
- Ring-Latenz: Zeit, die das Token benötigt für eine Rundreise, wenn niemand Daten sendet

Zuverlässigkeit

- IEEE 802.5 sieht zwei Bits vor, beide anfangs 0
- A-Bit: wird vom Empfänger gesetzt, wenn dieser ein Paket für ihn erkennt
- Sender kann daraus schließen, ob Ziel verfügbar ist
- C-Bit: wird vom Empfänger gesetzt, wenn dieser ein Paket komplett angenommen hat
- falls A aber nicht C gesetzt: Fehler, z.B. zu wenig Bufferspeicher, später nochmal probieren

Prioritäten

- 3-bit Prioritäten-Feld
- Beschlagnahmen des Tokens nur erlaubt, wenn Paketpriorität mindestens so hoch wie Tokenpriorität
- Knoten, der Paket hoher Priorität senden möchte, setzt dazu Reservierungsbits im Frame-Header eines Datenpakets
- (es sei denn dieses hat bereits höhere Priorität)
- zwingt Station mit Token dazu es abzugeben
- Problem: es kann vorkommen, daß Pakete niedriger Priorität bei hoher Zahl von Paketen hoher Priorität „ewig” liegen bleiben

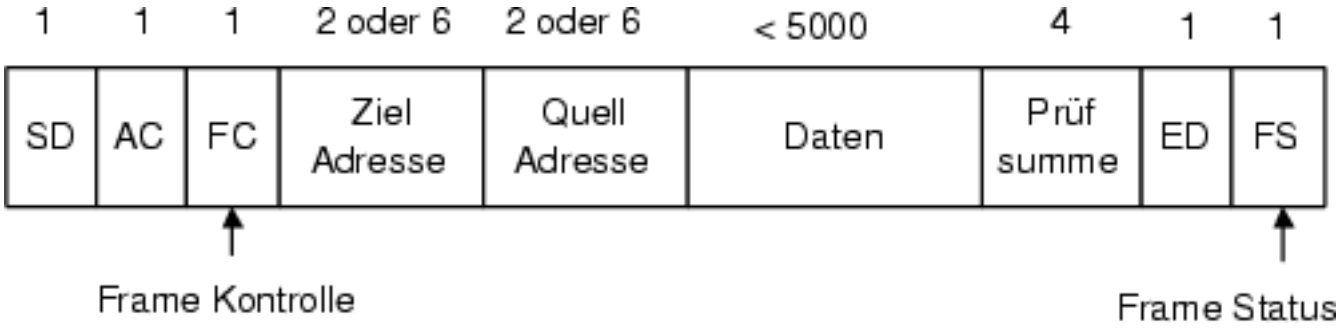
Verwaltung

- Monitor-Knoten mit Verwaltungsaufgaben
- jeder Knoten kann Monitor werden
- bei Ausfall des Monitor-Knotens oder Initialisierung: Neuwahl
- spezielles „claim token”-Paket
- falls mehr als ein Anwärter: höchste Adresse gewinnt

Verwaltungsaufgaben

- Verzögern, falls nur wenige Stationen
- Ankündigen, daß Monitorstation aktiv
- fügt neues Token ein, wenn altes verloren (z.B. Ausfall des Knotens mit dem Token oder Bitfehler)
- dazu wird Zeit zwischen Tokens gestoppt
- wenn mehr als maximale TRT, dann neues Token
- sorgt auch für das Entfernen defekter Pakete (Monitor-Bit)

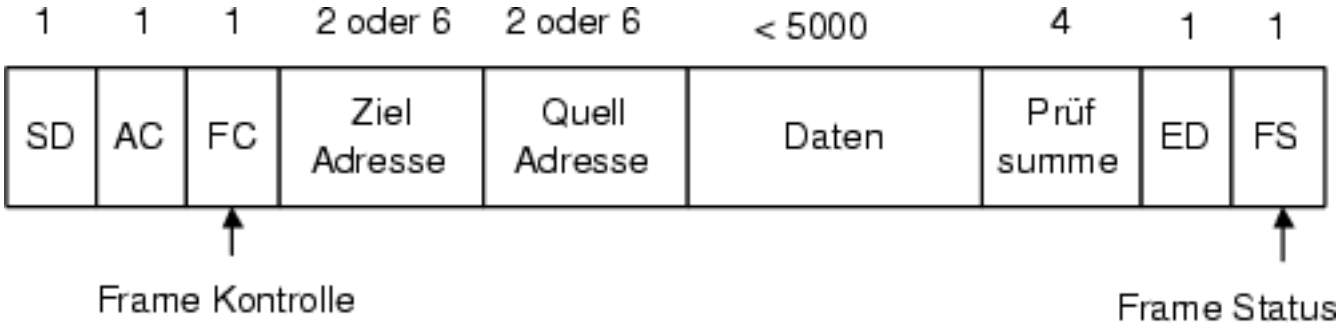
Token Ring Frame (1)



- Start- und Endmarkierungen durch „illegale” Manchester-Codes
- *access control*: Prioritäts- und Reservierungsbits
- *frame control*: bezeichnet übergeordnetes Protokoll
- *destination/source address*: Ziel- und Quelladresse, 48 Bit[1], kompatibel zu Ethernet-MAC-Adressen

[1] 16 Bit definiert, aber nicht mehr gebräuchlich

Token Ring Frame (2)



- *data, checksum*: Datenbereich, CRC-32
- *frame status*: enthält u.a. A- und C-Bits
- 4 Mbps-Ring: max. Framelänge 4096 Byte; nur 1 MAC-Frame auf Ring
- 16 Mbps-Ring: max. Framelänge 17800 Byte; mehrere MAC-Frames, getrennt von IDLE-Zeiten

FDDI

- *Fiber Distributed Data Interface (FDDI)*
- Glasfaser als Grundlage, benutzt 4B/5B-Kodierung
- Doppelring: Umschalten an den Enden bei Unterbrechung
- Durch „Concentrator” auch Knoten mit einfacher Anbindung möglich
- bis zu 500 Knoten, max. 2km zwischen Knoten
- max. 200km Glasfaser, also 100km Entfernung

Wireless (1)

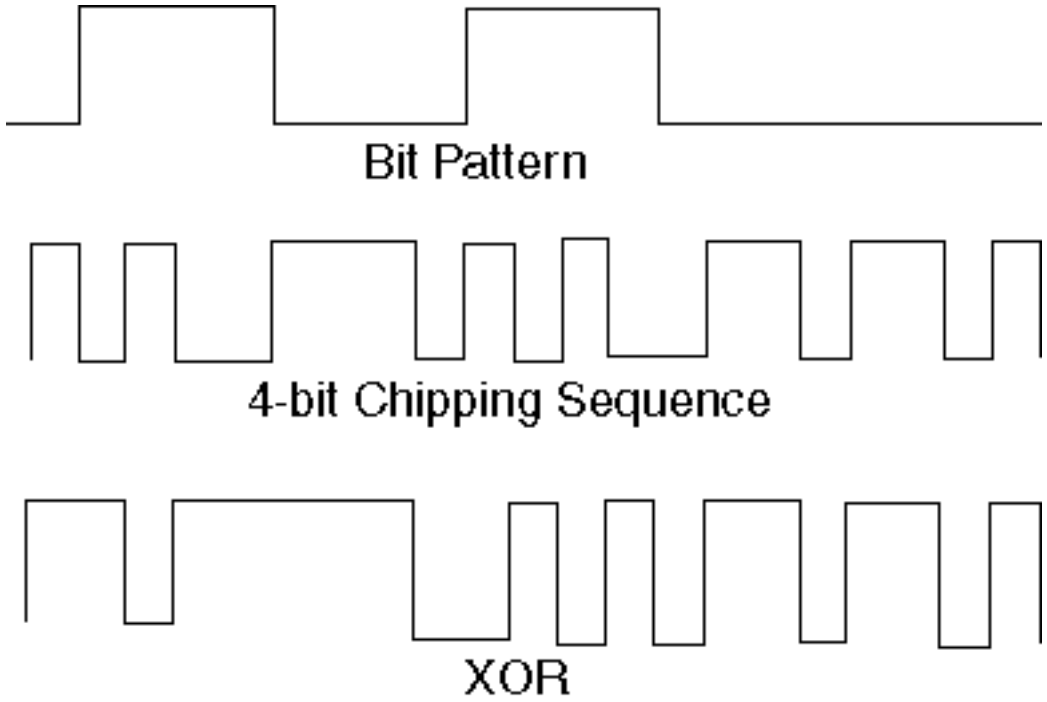
- IEEE 802.11 (und viele Erweiterungen)
- Übertragung via Funk, ursprünglich mit 1-2Mbps
- Erweiterung z.B. IEEE 802.11a: 5 GHz-Band, bis 54 Mbps
- oder z.B. IEEE 802.11b: 2,4 GHz-Band, bis 11 Mbps
  - 14 verschiedene Kanäle definiert zwischen 2.412 GHz und 2.484 GHz
  - davon USA: 1-11, Europa: 1-13, Frankreich: 10-13, Japan: 14

Wireless (2)

- Benutzt *spread spectrum*-Technologie
- Ziel: Interferenz mit anderen Geräten möglichst vermeiden
- z.B. *frequence hopping*: pseudozufälliges Wechseln der Sendefrequenz, Empfänger benutzt gleichen Algorithmus
- z.B. *direct sequence* mit n-bit *chipping code*
- d.h. XOR von Daten und „Zufallssequenz” mit höherer Baudrate als Daten

4-Bit-Chipping

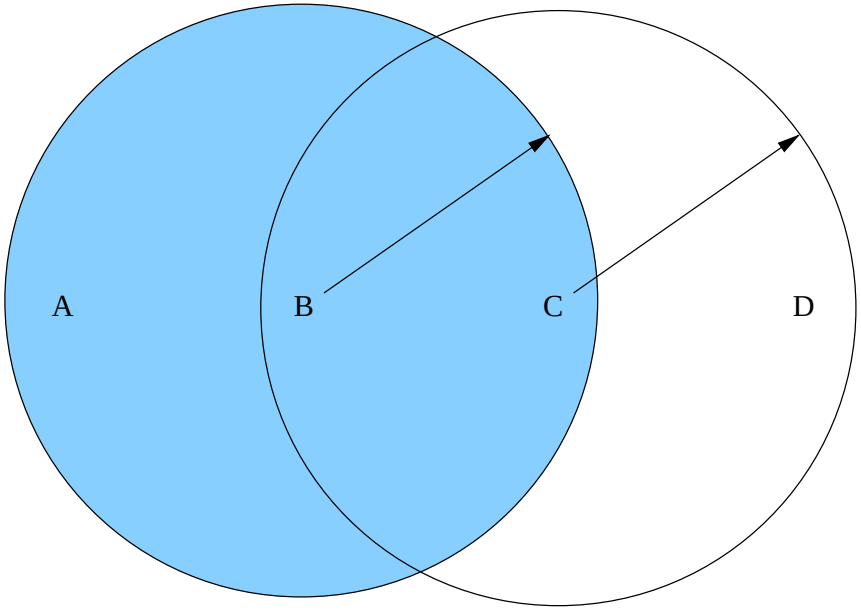
- 4-Bit-Chipping: „Zufallssequenz” wiederholt sich alle 4 Bits



Grundidee

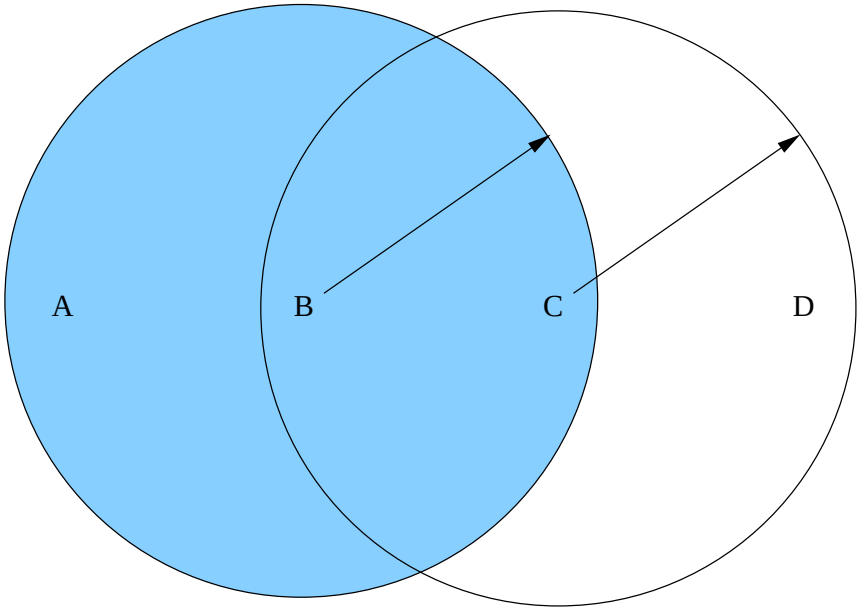
- Denkbar wäre, daß Ethernet-Algorithmus angewendet wird
- Grundidee bleibt:
- warten bis Übertragungskanal frei
- falls Kollision: Back off
- aber mit einigen Abwandlungen aufgrund von Problemen:

Probleme (1)



- A und C möchten mit B kommunizieren
- A und C wissen voneinander nichts
- Signale kollidieren bei B, aber weder A noch C bekommen dies mit
- A und C werden *versteckte Knoten (hidden nodes)* genannt

Probleme (2)



- B sendet an A
- C bekommt dieses mit
- C könnte dennoch gleichzeitig problemlos an D senden
- Dies ist das *exposed node problem*

MACA-Algorithmus (1)

- *Multiple Access with Collision Avoidance (MACA)*
- Idee: Vor den eigentlichen Daten werden Kontroll-Frames gesendet
- alle Knoten in Reichweite bekommen diese mit
- im *Request To Send (RTS)*-Frame ist Übertragungsdauer enthalten
- Zielknoten antwortet mit *Clear To Send (CTS)*-Frame und wiederholt Übertragungsdauer
- anschließend ist Übertragungsdauer allen Knoten in Reichweite bekannt

MACA-Algorithmus (2)

- Knoten, die CTS empfangen dürfen während Übertragungsdauer nicht senden
- Knoten, die RTS, nicht aber CTS empfangen dürfen gleichzeitig senden
- Nach der Übertragung der Daten sendet Empfänger ein ACK-Frame[2]
- andere Knoten müssen das ACK abwarten
- falls mehr als ein Knoten mit dem Senden von RTS beginnen und es zu einer Kollision kommt, dann werden sie kein CTS empfangen
- bei Kollisionen wird gemäß des bin. exp. Backoff-Algorithmus gewartet

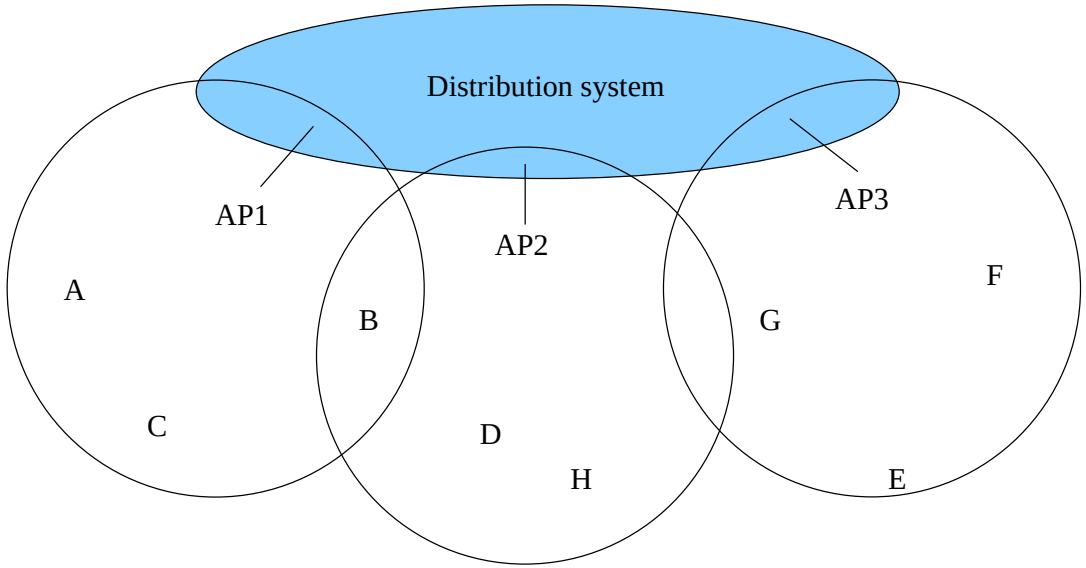
[2] Teil von MACAW: MACA for Wireless LANs

Verteilungssystem

- bislang: ausreichend für ad hoc-Konfiguration
- aber nicht alle Knoten erreichbar
- zudem: Knoten dürfen sich frei bewegen
- daher weitere Strukturen definiert: ein Verteilungssystem
- einige Knoten sind feststehend: *Access Points (AP)*

Access Points (1)

- bilden „Basisstation”
- Knoten ordnen sich einer Basisstation zu (*assoziieren/associate*)
- z.B. Knoten A möchte mit Knoten E kommunizieren



Access Points (2)

- A sendet Frame an AP1
- dieser leitet es an AP3 weiter über das Verteilungssystem
- dieser sendet es an E
- woher AP1 weiß, daß es E über AP3 erreichen kann ist nicht Teil von IEEE 802.11
- dazu kann z.B. ein Bridging-Protokoll dienen

Access Points (3)

- Wie wählen aber die Knoten ihren Access Point?

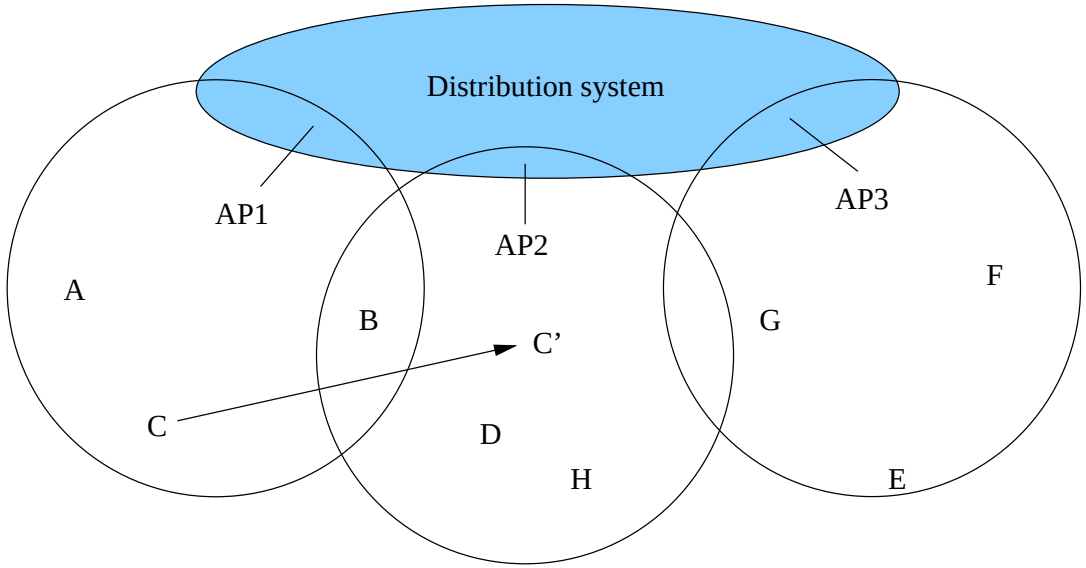
- IEEE 802.11 benutzt Methode *Scannen (scanning)*:

  1. der Knoten sendet ein *Sondierungs-Frame (probe frame)*
  2. alle APs in Reichweite antworten mit einem *Probe Response*
  3. der Knoten wählt einen AP aus und sendet ein *Association Request*
  4. der angesprochene AP antwortet mit *Association Response*

- zusätzlich: AP sendet regelmäßig *Signalisierung (beacon)*

Access Points (4)

- wenn Knoten „unglücklich” mit aktuellem AP wird er diese Prozedur einleiten
- z.B. wenn das Signal zu schwach wird
- in Schritt 4 wird der alte AP durch den neuen AP über das Verteilungssystem informiert



Wireless Frame Format

- Dauer
- Quell- und Zieladresse sowie zwei Zwischenadressen à 48 Bit
- bis zu 2312 Bytes Daten
- CRC-32
- 6-bit „Type”-Feld zeigt an, ob RTS, CTS, Daten oder Scanning-Paket
- zwei 1-Bit-Felder ToDS, FromDS