

MAC-protocols

Kontrollierter Zugriff

Wahlfreier Zugriff

Collision avoidance

Collision resolution

Reservation-based

Token-based

Time-based

Master-Slave

Priority-based

probabilistic

dynamic

static

ATM

TDMA:

**TTP,
Maruti**

**Token-Ring
Token-Bus**

**Timed
Token
Protocol**

**CSMA/CA :
Collision Avoidance**

**IEEE 802.11
P-persistent CSMA**

VTCSMA

**ProfiBus DP
FIP
CAN-Open**

**CSMA/CA :
Consistent Arbitration**

CAN

**CSMA/CD :
Carrier Sense Multiple Access /
Collision Detection**

Ethernet

VTCSMA

Virtual Time CSMA

Verfahren zur Bestimmung einer globalen Prioritätsordnung von Nachrichten

M.L. Molle, L. Kleinrock: *Virtual Time CSMA: Why two clocks are better than one*
IEEE Trans. on Communications, COM 33(9), 1985

W. Zhao, J.A. Ramamritham: *Virtual Time CSMA Protocols for Hard Real-Time Communication*
IEEE Trans. on Softw. Engineering, SE 13(8), 1987

Annahmen:

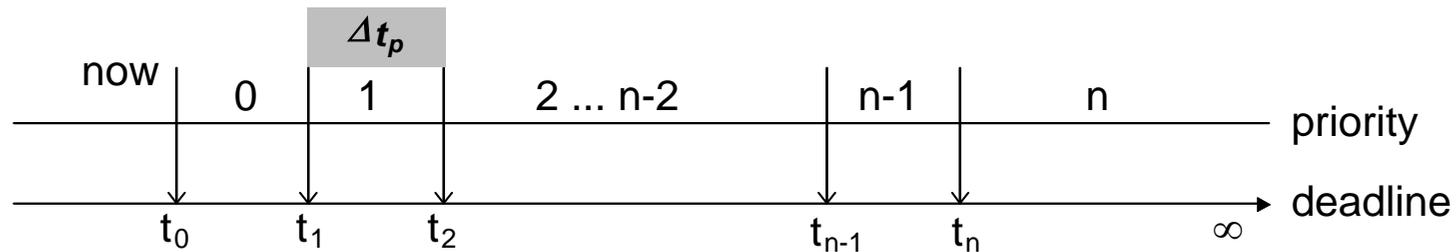
Folgende Informationen stehen jedem Knoten zur Verfügung:

- **Status des Kanals (frei oder besetzt)**
- **globale Zeit durch synchronisierte Uhren**
- **Lokale Prioritäten der sendebereiten Nachrichten**

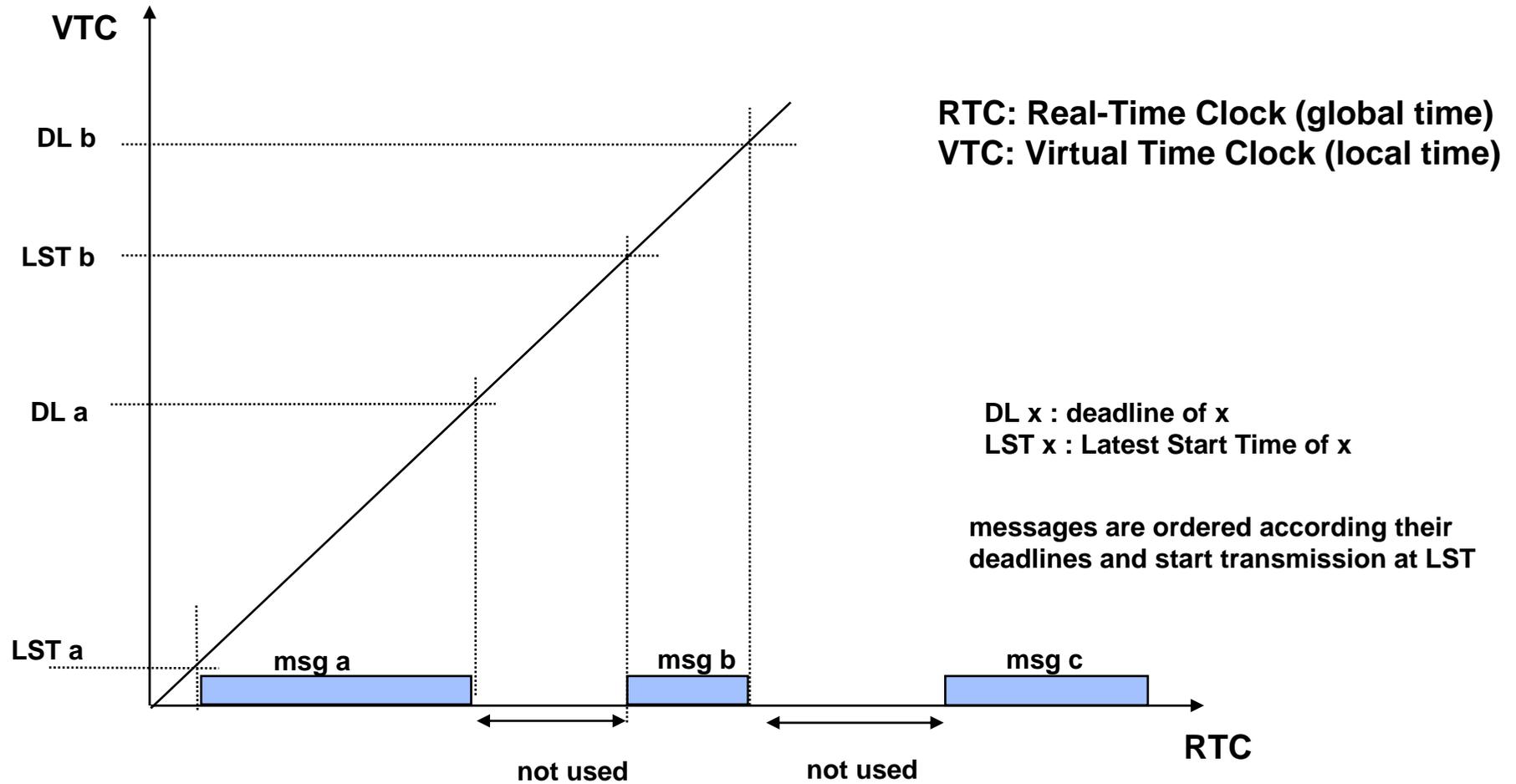
Grundidee des Verfahrens:

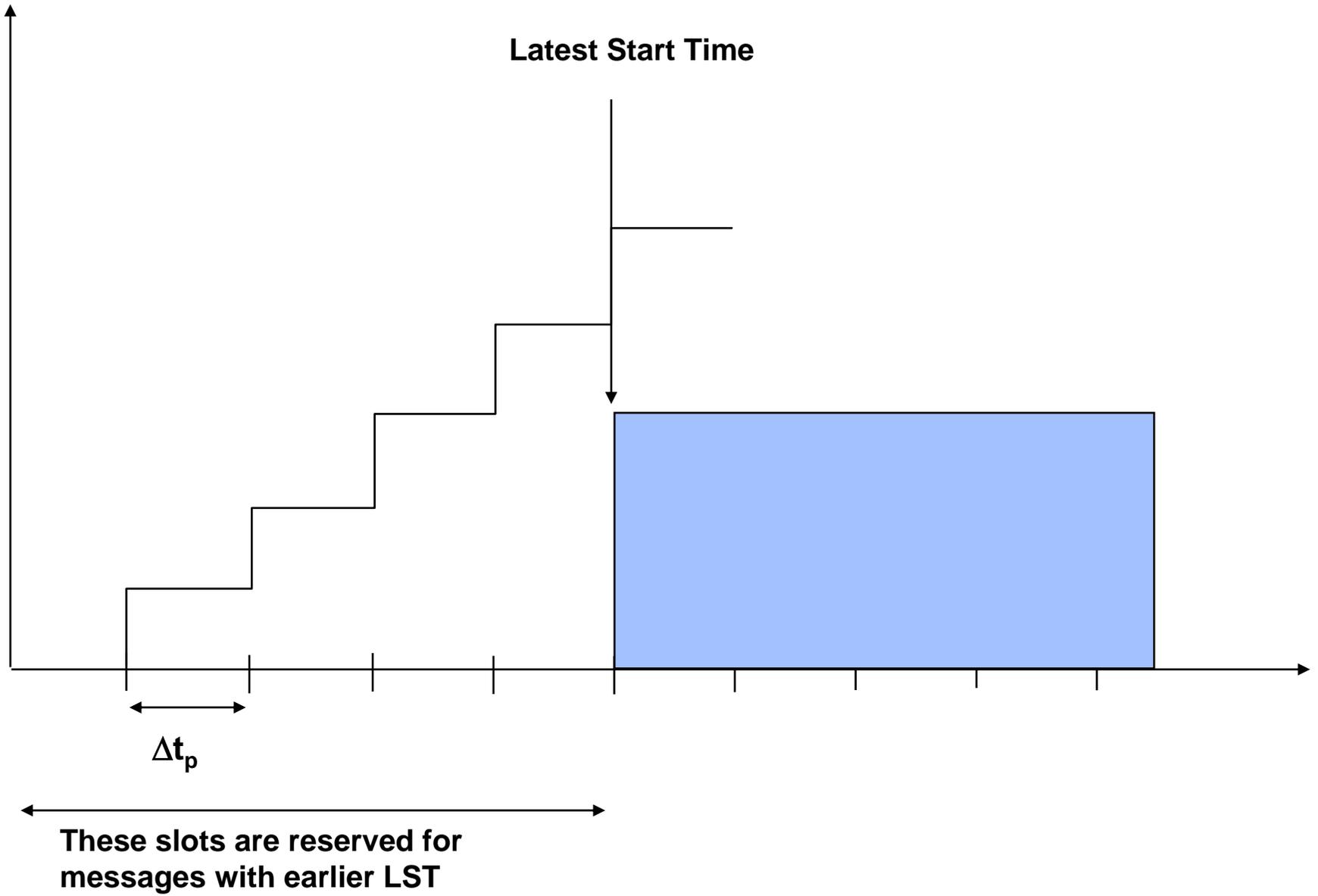
Berechnung der globalen Prioritätsordnung als Funktion der Zeit.

Abbildung von Deadlines auf Prioritäten

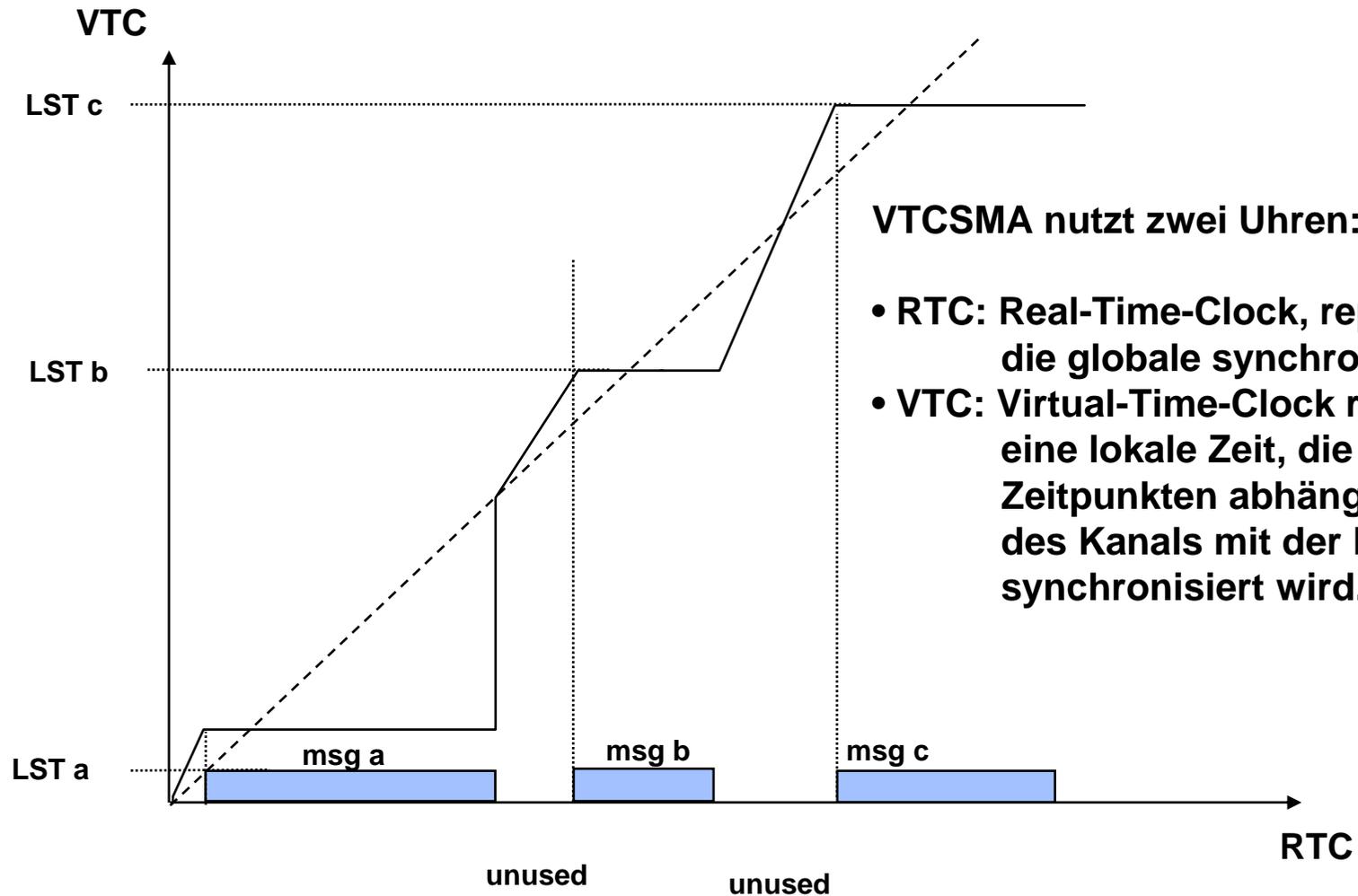


Virtual Time CSMA



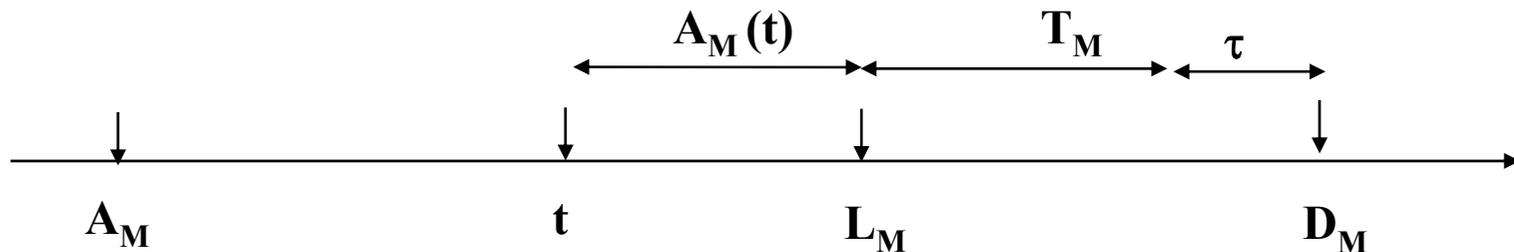


Virtual Time CSMA



Ausprägungen des VTCSMA-Algorithmus

- τ : Laufzeit von einem Ende des Netzwerkes zum anderen
- A_M : Nachricht wird in die lokale WS eingereicht (Arrival Time)
- T_M : Zeit, um die Nachricht zu übertragen
- D_M : Deadline, zu der die Nachricht im Empfänger ausgeliefert werden soll
- L_M : Spätester Zeitpunkt, an dem die Nachricht gesendet werden muß, um die Deadline einzuhalten.
D.h. $L_M = D_M - T_M - \tau$
- $A_M(t)$: Maximale Zeit, die eine Nachricht zum Zeitpunkt t noch verzögert werden darf ohne ihre Deadline zu verpassen.
D.h. $A_M(t) = D_M - T_M - \tau - t$



Ausprägungen des VTCSMA-Algorithmus

Wenn eine Nachricht in die Warteschlange eingereicht wird wird $VSX(M)$ wie folgt gesetzt:

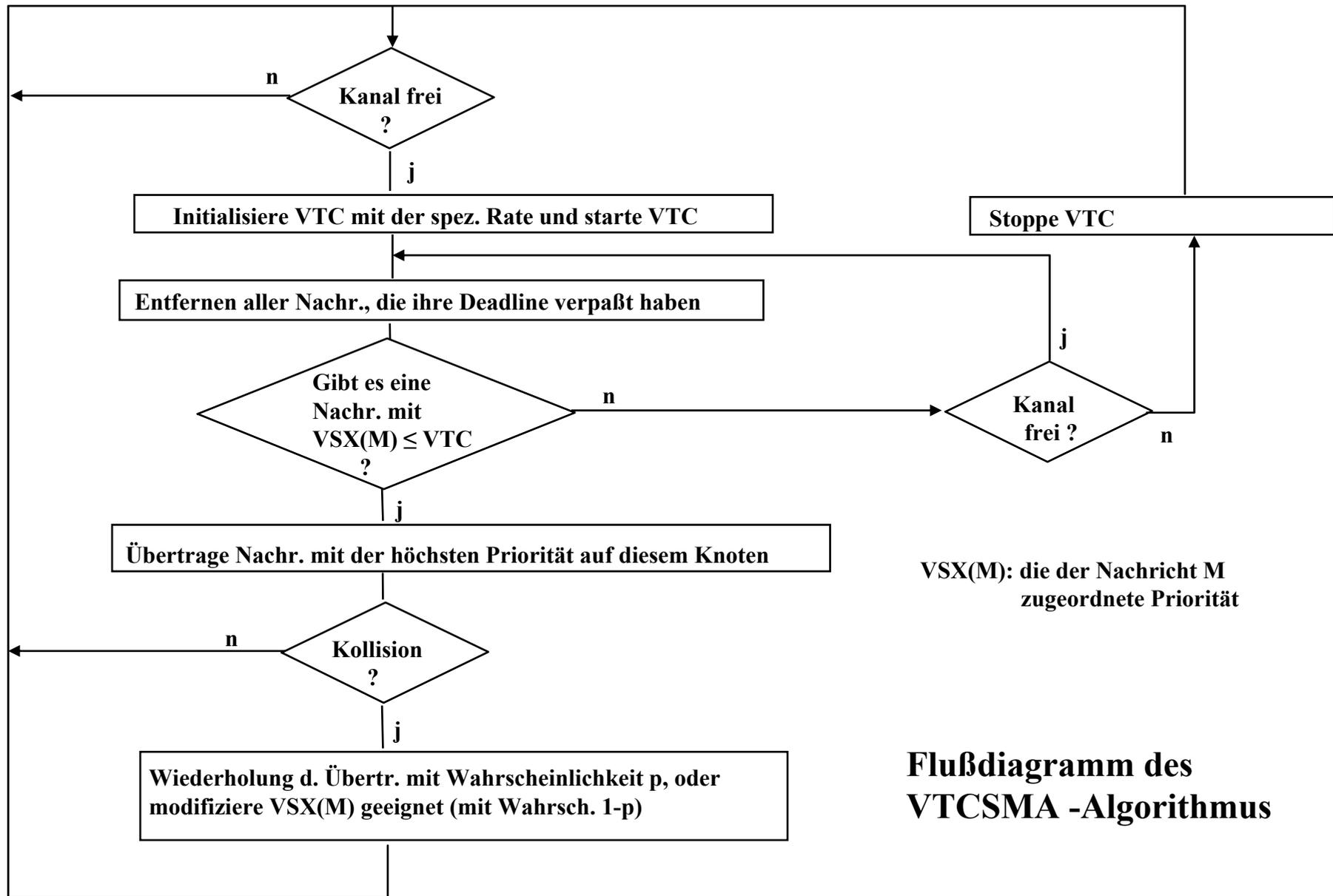
$$VSX(M) = \begin{cases} A_M : \text{für VTCSMA-A (Earliest Arrival First)} \\ T_M : \text{für VTCSMA-T (Minimum Transmission Time First)} \\ L_M : \text{für VTCSMA-L (Minimum Laxity First)} \\ D_M : \text{für VTCSMA-D (Earliest Deadline First)} \end{cases}$$

Falls eine Kollision auftritt, wird mit der Wahrsch. p der Sendevorgang sofort wiederholt oder mit $1-p$ der Wert von $VSX(M)$ mit einer Zufallszahl aus folgenden Bereichen modifiziert :

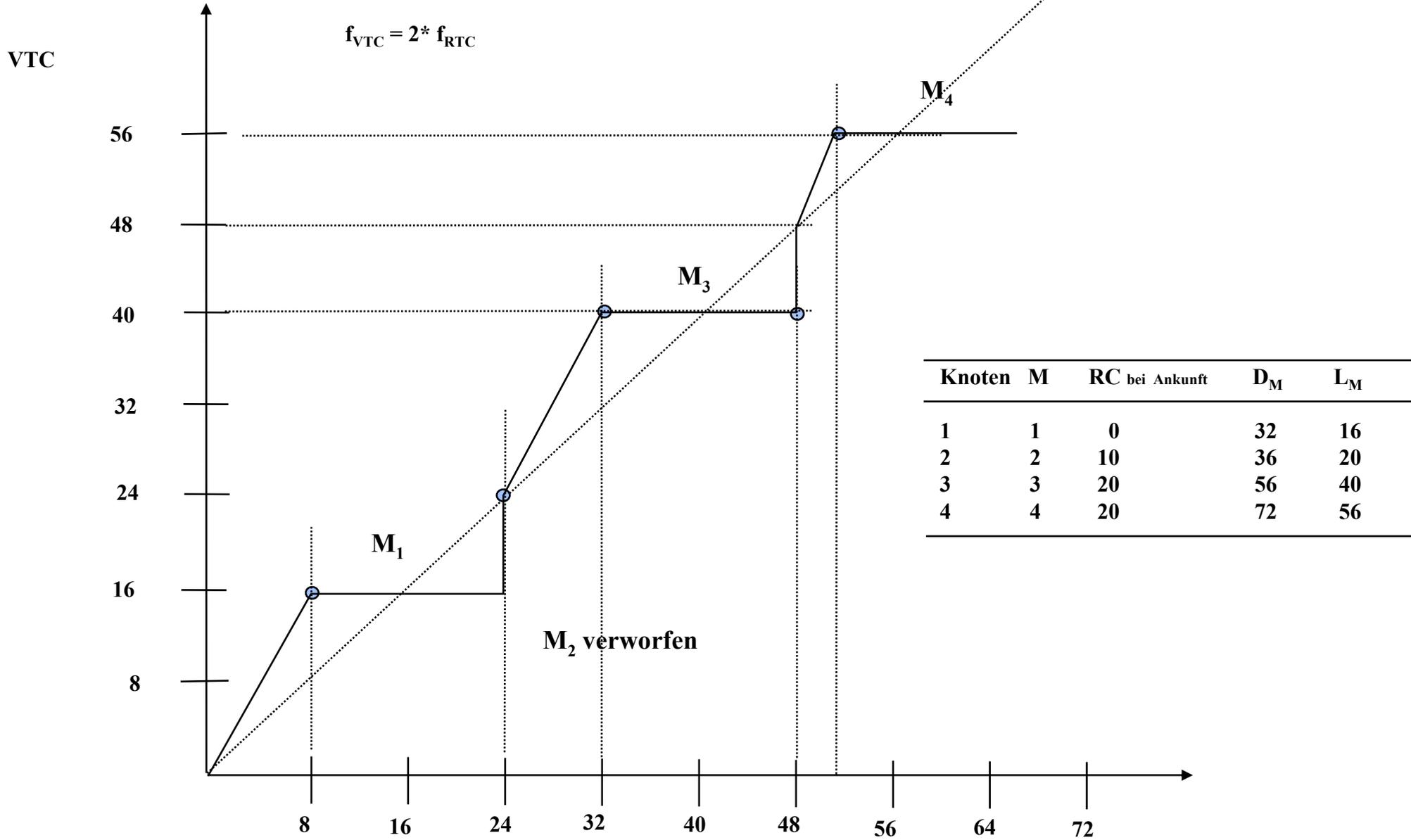
$$I = \begin{cases} (\text{Aktueller Wert von VTC}, L_M) & \text{für VTCSMA-A} \\ (0, T_M) & \text{für VTCSMA-T} \\ (\text{Aktueller Wert von RTC}, L_M) & \text{für VTCSMA-L} \\ (\text{Aktueller Wert von RTC}, D_M) & \text{für VTCSMA-D} \end{cases}$$

Wenn der Kanal frei ist, wird VTC mit folgenden Werten initialisiert:

$$VC = \begin{cases} \text{keine Änderung} & \text{für VTCSMA-A} \\ 0 & \text{für VTCSMA-T} \\ RTC & \text{für VTCSMA-L} \\ RTC & \text{für VTCSMA-D} \end{cases}$$



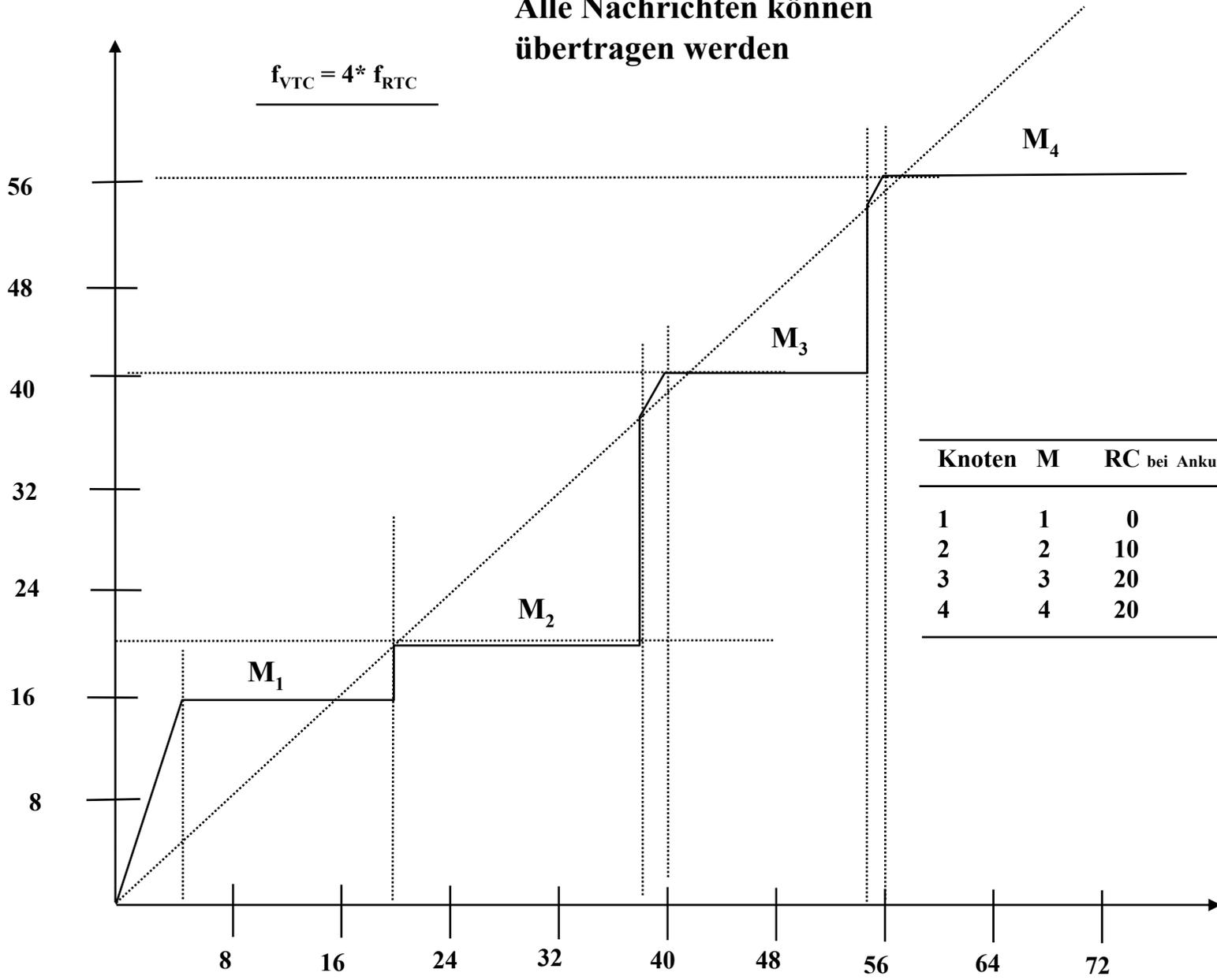
Beispiel für VTCSMA-L



Beispiel für VTCSMA-L
Alle Nachrichten können
übertragen werden

VTC

$$f_{VTC} = 4 * f_{RTC}$$



Knoten	M	RC bei Ankunft	D _M	L _M
1	1	0	32	16
2	2	10	36	20
3	3	20	56	40
4	4	20	72	56

VTCSMA Zusammenfassung

- **VTCSMA haben eine bessere Rate für verpaßte Deadlines verglichen mit CSMA.**
- **Die beste Rate wird mit dem VTCSMA-D Algorithmus erreicht.**
- **Die Leistungsfähigkeit ist eine Funktion der VTC-Frequenz
So lange die Laufzeiten nicht zu groß werden und das Netz nicht überlastet ist, wird eine Leistung nahe am Optimum erreicht.**
- **Der zusätzliche Aufwand der Uhrensynchronisation wird nicht berücksichtigt.**

Probleme:

- 1. Verlorene Bandbreite durch Wartezeiten.**
- 2. Zusätzlicher Aufwand durch dynamische Prioritäten.**

Nr	Größe	Jitter	Periode	Art	Deadline	Von	Zu
0	8	0.6	100.0	P	100.0	Battery	V/C
1	8	0.7	100.0	P	100.0	Battery	V/C

Der SAE Benchmark

Nr	Größe	Jitter	Periode	Art	Deadline	Von	Zu
2	8	1.0	1000.0	P	1000.0	Battery	V/C
3	8	0.8	100.0	P	100.0	Battery	V/C
4	8	1.1	1000.0	HP	1000.0	Battery	V/C
5	8	0.9	100.0	P	100.0	Battery	V/C
6	8	0.1	5.0	HP	5.0	Driver	V/C
7	8	0.1	5.0	HP	5.0	Brakes	V/C
8	8	0.1	5.0	HP	5.0	Brakes	V/C
9	8	0.2	100.0	P	100.0	Trans	V/C
10	8	0.1	5.0	P	5.0	Trans	V/C
11	8	0.8	100.0	HP	100.0	Brakes	V/C
12	1	1.2	1000.0	HP	1000.0	Battery	V/C
13	4	0.1	50.0	HS	5.0	Battery	V/C
14	1	0.2	50.0	S	20.0	Driver	V/C
15	1	0.3	50.0	S	20.0	Driver	V/C
16	2	0.4	50.0	S	20.0	Driver	V/C
17	1	0.3	20.0	S	20.0	Brakes	V/C
18	1	0.5	50.0	HS	20.0	Driver	V/C
19	3	0.6	50.0	S	20.0	Driver	V/C
20	2	0.3	1000.0	HS	1000.0	Trans	V/C
21	3	0.7	50.0	S	20.0	Driver	V/C
22	1	0.2	50.0	S	20.0	Battery	V/C
23	1	0.3	50.0	S	20.0	Battery	V/C
24	1	0.4	50.0	S	20.0	Battery	V/C
25	1	0.8	50.0	S	20.0	Driver	V/C
26	1	0.9	50.0	S	20.0	Driver	V/C
27	1	0.5	50.0	S	20.0	Battery	V/C

Nr	Größe	Jitter	Periode	Art	Deadline	Von	Zu
28	8	0.3	10.0	HP	10.0	V/C	Battery
29	8	0.4	10.0	HP	10.0	V/C	Battery
30	2	0.5	50.0	S	20.0	V/C	Trans
31	8	0.1	5.0	P	5.0	V/C	Battery
32	1	1.6	1000.0	HP	1000.0	V/C	Battery
33	8	0.6	50.0	S	20.0	V/C	Battery
34	1	0.7	50.0	S	20.0	V/C	Battery
35	2	1.7	1000.0	P	1000.0	V/C	Brakes
36	1	0.8	50.0	HS	20.0	V/C	Brakes
37	1	0.9	50.0	HS	20.0	V/C	Battery
38	7	1.0	50.0	HS	20.0	V/C	Ins.
39	1	1.1	50.0	S	20.0	V/C	I/MC
40	1	0.3	50.0	S	20.0	I/MC	V/C
41	8	0.2	5.0	P	5.0	V/C	I/MC
42	8	0.1	5.0	P	5.0	I/MC	V/C
43	1	1.2	50.0	S	20.0	V/C	I/MC
44	1	0.4	50.0	S	20.0	I/MC	V/C
45	1	1.3	50.0	S	20.0	V/C	I/MC
46	1	0.5	50.0	S	20.0	I/MC	V/C
47	1	1.4	50.0	S	20.0	V/C	I/MC
48	8	0.2	5.0	HP	5.0	I/MC	V/C
49	2	0.6	50.0	S	20.0	I/MC	V/C
50	1	0.7	50.0	A	20.0	I/MC	V/C
51	8	0.8	50.0	A	20.0	I/MC	V/C
52	1	1.5	50.0	S	20.0	V/C	I/MC

S: sporadische, P: periodisch, HS: HRTN sporadisch, HP: HRTN periodisch

SAE-Benchmark

of participants: 7
of messages: 53
Type of messages: per. (11), h-per.(12), spor. (24), h-spor.(5)
Size of messages (bytes): 1 (24), 2 (5), 3 (2), 4 (1), 8(20)
period (ms) : 5(8), 10(2), 20(1), 50(30), 100(6), 1000(6)

The benchmark generates a system load* for the WaveLan Modem of 464.000 bits/sec (16152 payload).

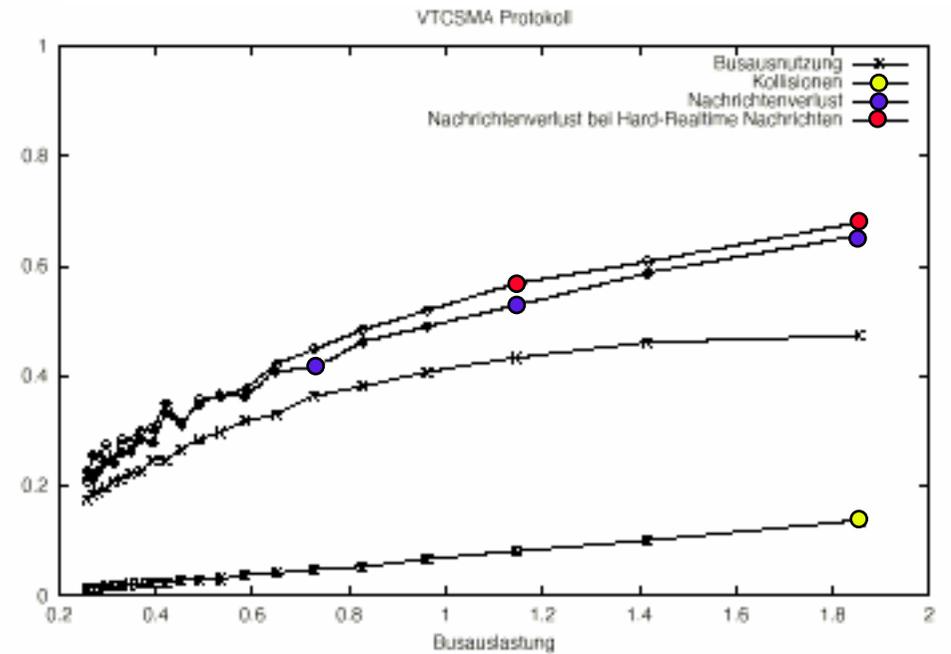
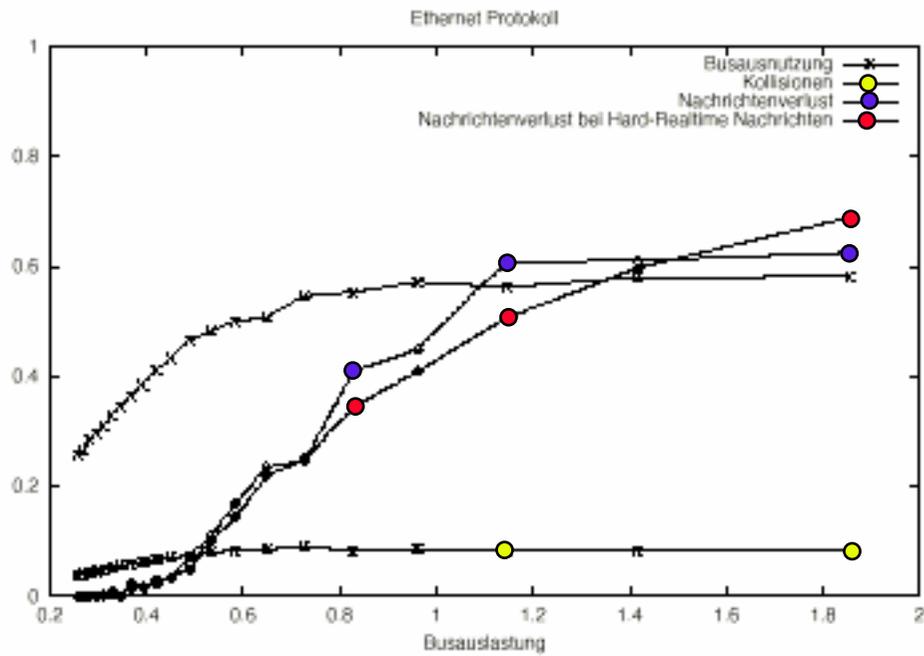
*system load= \sum (msg size/ period)

$$\text{Buslast} = \frac{\text{Bruttodatenaufkommen}}{\text{Zeit}}$$

$$\text{Busausnutzung} = \frac{\text{Zeit, in der Nachrichten gesendet werden}}{\text{Gesamtzeit}}$$

$$\text{Kollisionen} = \frac{\text{Zeit, die Bus durch Kollisionen belegt ist}}{\text{Gesamtzeit}}$$

$$\text{Nachrichtenverlust} = \frac{\text{Anzahl der verlorenen Nachrichten}}{\text{Anzahl der erzeugten Nachrichten}}$$



Was sind die Gründe für das schlechte Abschneiden von VTCSMA ?

Predictive **p-persistent** CSMA

LON: <http://echelon.com>

Prinzip:

- Jede Nachricht wird nicht sofort gesendet, sondern mit einer Wahrscheinlichkeit p im Zufallsintervall (entspricht einem durchnummerierten Zeitschlitz (Beta-2 Slot)) der Länge s .
- Wenn keine Nachrichten gesendet werden, d.h. eine geringe Last auf dem Netzwerk liegt, werden 16 Zufallsintervalle berücksichtigt. Die durchschnittliche Wartezeit beträgt damit: 8 s, die worst case Wartezeit: 16 s
- Steigt die Netzlast an, wird dynamisch die Anzahl der Zufallsintervalle um den Faktor n ($n = 1 \dots 63$) erhöht, so dass die Wahrscheinlichkeit von Kollisionen sinkt. n wird als „estimated channel backlog“ bezeichnet.
- Der Mechanismus zur Erhöhung der Zufallsintervalle nutzt die Tatsache aus, dass die meisten LON Pakete beantwortet werden. Da in jedem Paket die Anzahl der Antworten mitgeschickt wird können die Zielknoten den erwarteten Verkehr abschätzen.

Channel Utilization versus Load for Random Access Protocols

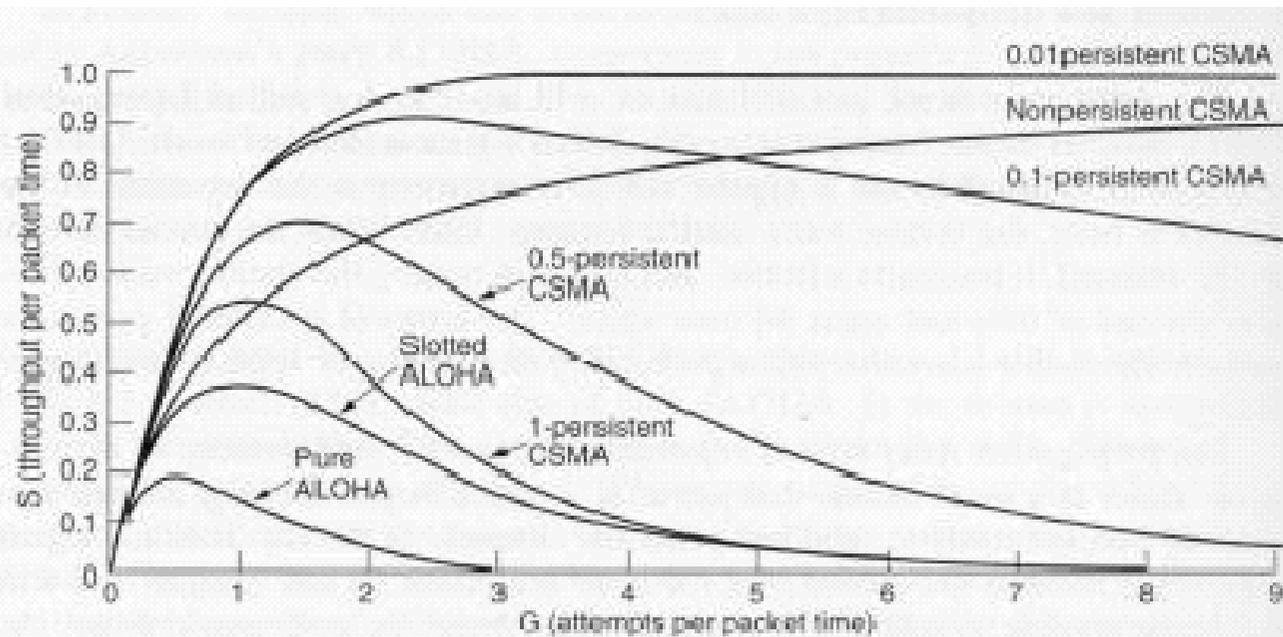
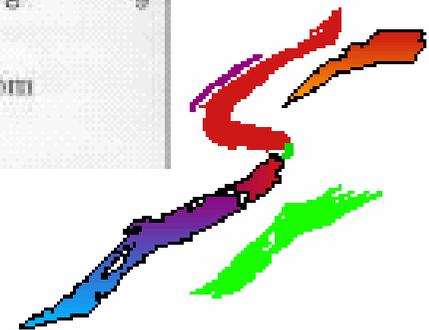
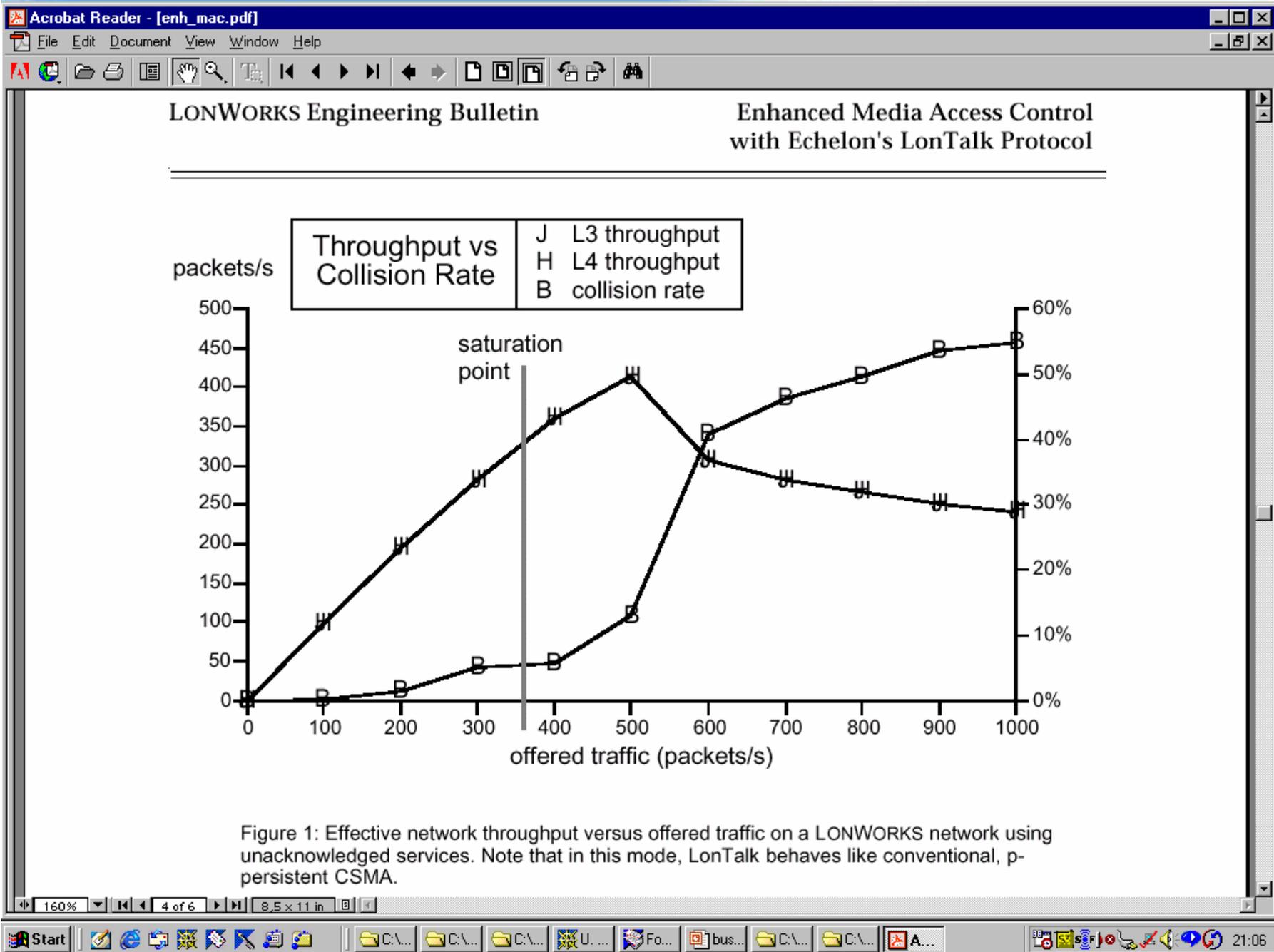


Fig. 4-4. Comparison of the channel utilization versus load for various random access protocols.





Das Window Protokoll

W. Zhao, J. A. Stankovic, K Ramamritham:

A Window Protocol for Transmission of Time-Constraint Messages

IEEE Trans. on Computers, 39(9) 1990

Annahmen:

Folgende Informationen stehen jedem Konten zur Verfügung:

- **Status des Kanals (frei oder besetzt)**
- **globale Zeit durch synchronisierte Uhren**
- **Die LTTT der sendebereiten Nachrichten (LTTT: Latest Time To Transmit)**

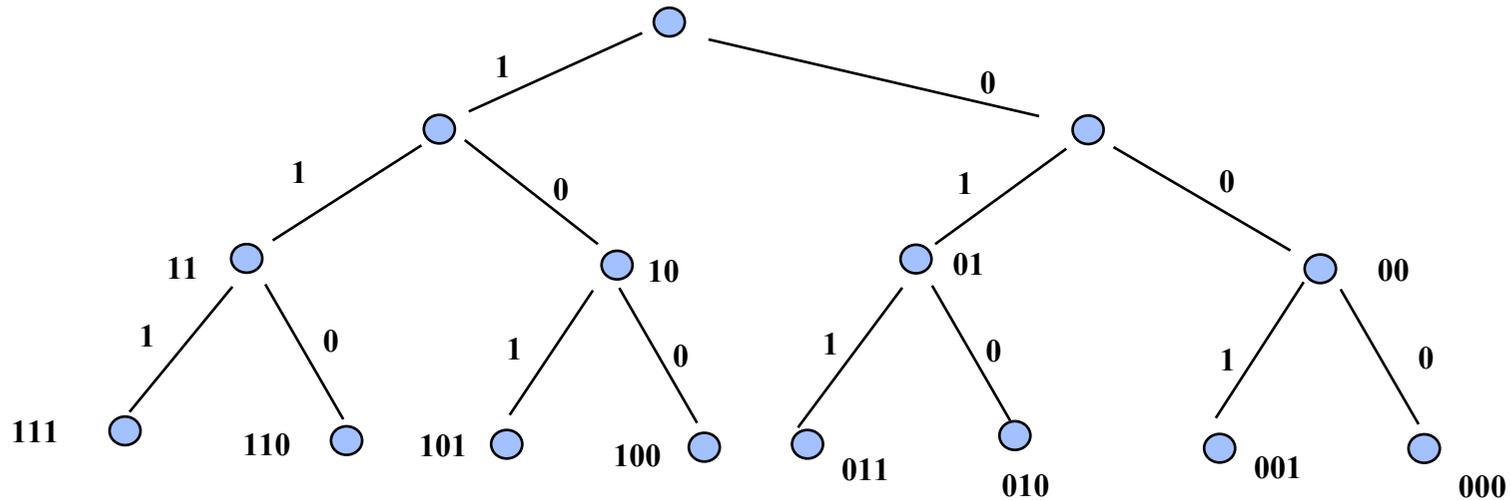
Grundidee des Algorithmus:

- **Teile die Zeit in Fenster fester Länge**
- **Falls der Bus frei ist sende die Nachricht mit der geringsten LTTT zu Beginn eines Fensters**
- **Falls eine Kollision auftritt, halbiere das Fenster. Falls die LTTT in das neue Fenster fällt, sende die Nachricht.**
- **Falls eine Nachricht nicht zu ihrer LTTT gesendet werden kann, wird sie verworfen.**

Prioritätsbasiertes Protokoll

T. Znati, L. M. Ni:

A prioritized multi-access protocol for distributed real-time applications,
Proc. 7th Int'l Conference on distributed comp.systems, September 1987



Prioritäten werden als in einem Baum angeordnet angesehen. Bei Kollisionen wird der jeweilige Baum halbiert, Nur die höher prioren Nachrichten werden gesendet. Entspricht dem Window Protocol bei entsprechender Transformation von Deadlines (LTTT) in Prioritäten.

Token basierte Protokolle

Token-Protokolle

Token Passing:

Mehrere Master rotieren ein Token.

Token Ring: physischer Ring (IEEE 802.5)

Token Bus: logischer Ring (IEEE 802.4)

Delegated Token:

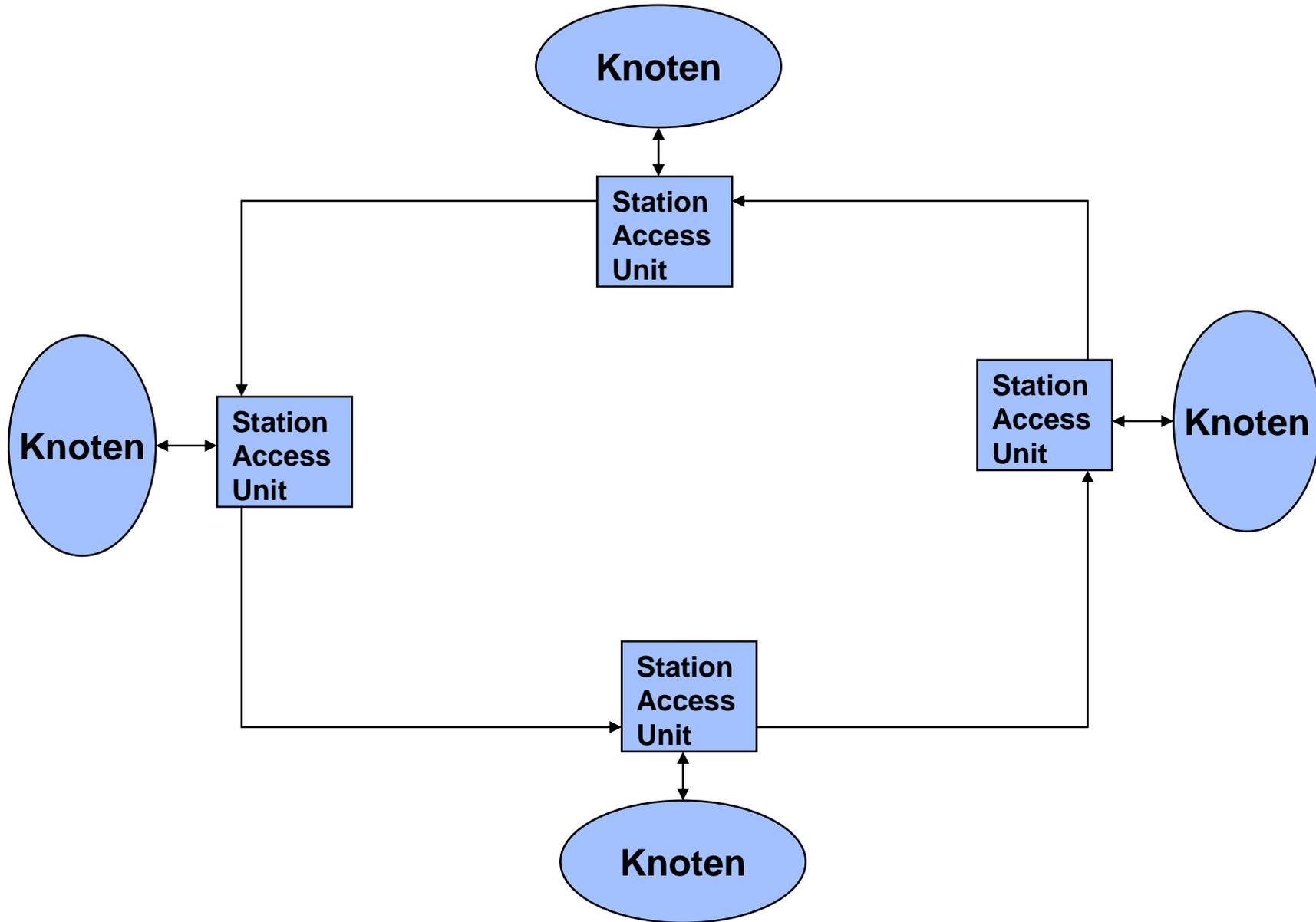
Verbindungsorientiert: zentraler Busarbiter vergibt Token für die Versendung einer oder mehrerer Nachrichten. Teilnehmer kann dann selbständig mit anderen Teilnehmern kommunizieren.

Nachrichtenorientiert: Busarbiter fordert über eine spezielle Nachricht, dem Token, das die ID der gewünschten Nachricht enthält, den zuständigen Teilnehmer zum Senden der Nachricht auf. Zentral gesteuerte Nachrichtenverteilung.

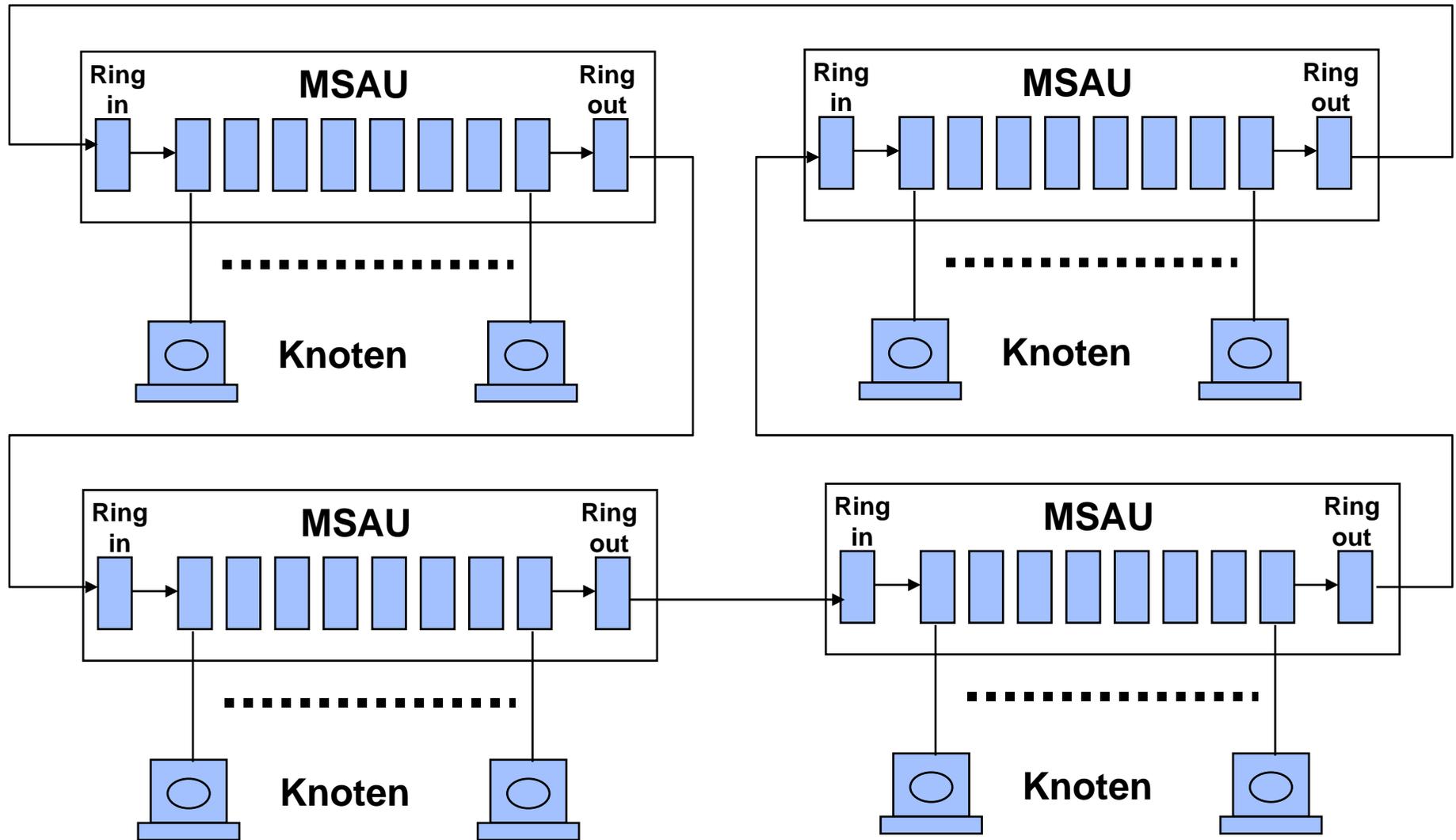
Beisp.: Verbindungsorientiert: Profibus (Token Passing (logischer Ring), Master und Slaves)

Nachrichtenorientiert: FIP (Factory Instrumentation Protocol)

Token Protokolle

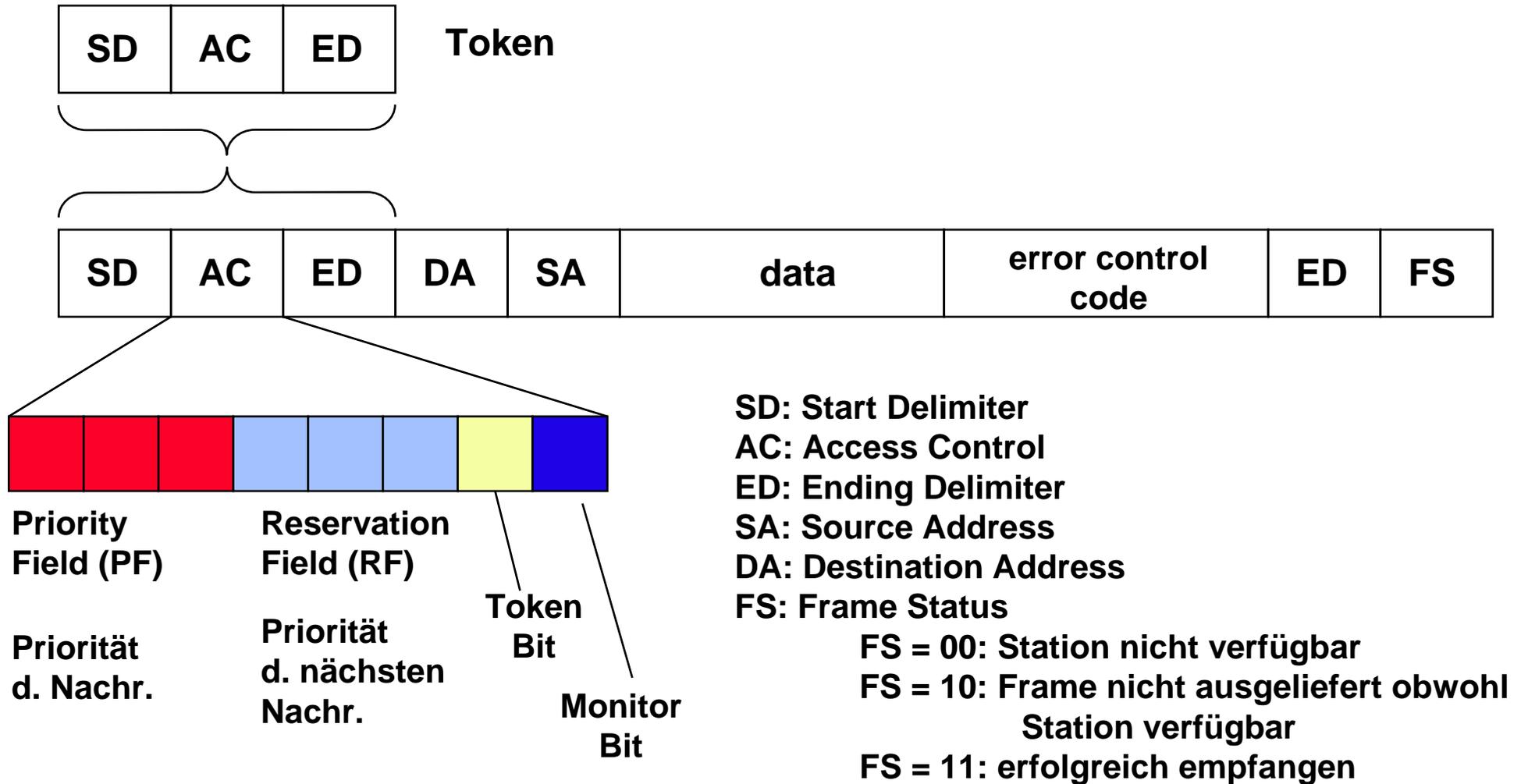


Token Netzwerk Topologien

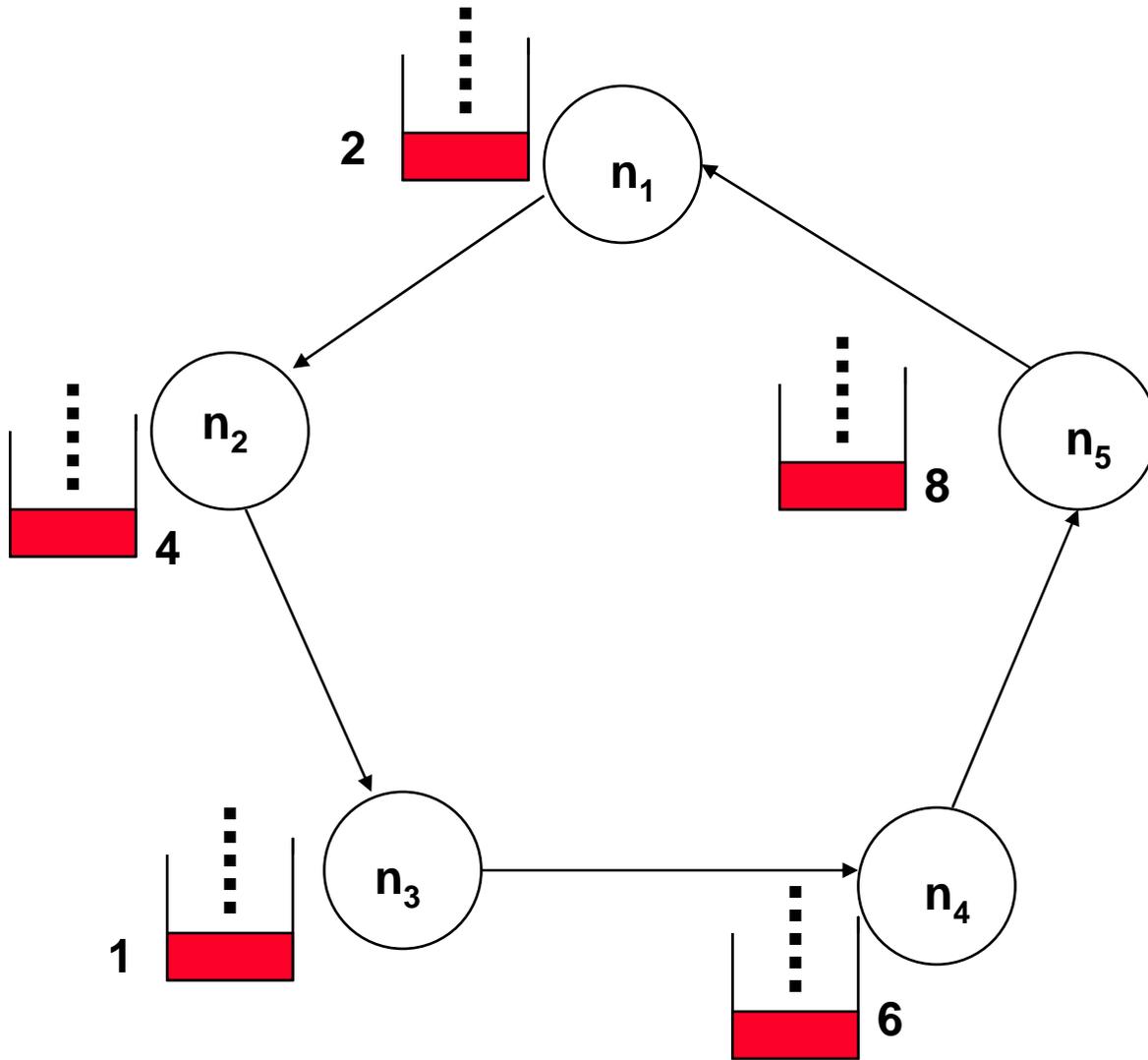


MSAU: Multi Station Access Unit

802.5 Token Ring Frame Format



Priorisierte Reservierung von Nachrichten



Runde n:

Knoten n_4 sendet Paket mit Priorität x. RF ist RF=6.

- n_1 ändert RF = 2
- n_2 und n_5 ändern nichts
- n_3 ändert RF in RF = 1

Runde n+1:

- n_4 generiert Token mit PF = 1 und sendet es auf den Ring
- keiner der Knoten ausser n_3 kann das Token nutzen, da die Priorität zu niedrig sind.
- n_3 erkennt, dass eine Nachricht mit entspr. Prio anliegt und fügt sie dem Token an. Gleichzeitig setzt n_3 das RF auf den ursprünglichen Wert (=2) zurück.

Token Ring/IEEE 802.5

Priority System

Token Ring networks use a sophisticated priority system that permits certain user-designated, high-priority stations to use the network more frequently. Token Ring frames have two fields that control priority: the *priority* field and the *reservation* field.

Only stations with a priority equal to or higher than the priority value contained in a token can seize that token. After the token is seized and changed to an information frame, only stations with a priority value higher than that of the transmitting station can reserve the token for the next pass around the network.

When the next token is generated, it includes the higher priority of the reserving station. Stations that raise a token's priority level must reinstate the previous priority after their transmission is complete.

Planbarkeitsanalyse

Schlüsselparameter ist $W_T = (n-1) D_B + T_{prop}$

W_T = Walktime: Zeit, die eine Runde im Token Ring dauert

n = Anzahl der Stationen im Ring

D_B = Station Delay: Verzögerungen der Nachricht in jeder Station

T_{prop} = Laufzeit der Nachricht auf dem Ring

- Nachrichten können nicht unterbrochen werden
- Wegen der Laufzeit im Ring kann die Priorität im RF-Feld veraltet sein
- Zusätzlich zur Nutzlast der Nachricht muss die Kontrollinformation übertragen werden

Das “Timed Token Protokoll”

Timed Token Protokoll

Eigenschaften:

- garantierte Übertragung von zyklischen HRT-Nachrichten
 - der Zyklus, in dem Stationen senden können, ist beschränkt
 - die Menge der Information (Bandbreite) ist bekannt und garantiert



Synchrone Nachrichten

- Übertragung von SRT- oder NRT-Nachrichten
die Übertragung von SRT- oder NRT-Nachrichten darf den Zyklus nicht verlängern



Asynchrone Nachrichten

Timed Token Protokoll

Was ist notwendig, um eine Garantie für synchrone Nachrichten geben zu können ?

- 1. Wie lange ist die Zeitspanne T , die eine Nachricht höchstens auf ihre Versendung warten muss?**

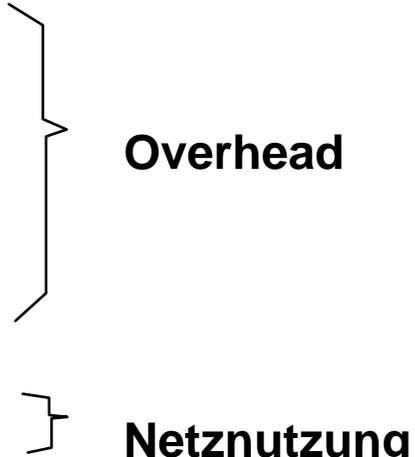
Bestimmung des Wertes T . Jeder Knoten sendet seine Anforderung, indem er ein T_i angibt. Das kleinste T_i bestimmt die Anforderungen und damit $T = \min (T_i)$

- 2. Wie groß ist die Menge der Information, die die Knoten in diesem Zyklus senden können?**

Target Token Rotationszeit (TTRT)

TTRT ist die Zeit, die vergeht, damit das Token einen vollständigen Zyklus im Ring absolviert.

Sie setzt sich zusammen aus:

- Verzögerung auf dem Medium für die Nachricht(en)
 - Token Übertragungszeit
 - Zeit, um das Token vom Netz zu holen
 - Latenz des Netzinterfaces
- 
- Overhead
- Netznutzung

Annahme: Overhead ist klein und vorhersagbar

Target Token Rotationszeit (TTRT)

TTRT ist keine worst-case Annahme, sondern eine systemabhängige Annahme, die synchrone und asynchrone Nachrichten einschliesst und z.B. auf Mittelwerten beruht. Daher kann $T = \text{TTRT}$ nicht garantiert werden. Es kann aber gezeigt werden, dass durch das Timed Token Protokoll die obere Schranke: $T = 2 \text{ TTRT}$ garantiert wird.

Trick: Wenn eine Nachricht alle $T = \min(T_i)$ Zeiteinheiten gesendet werden muss, wird

$\text{TTRT} = T/2$ gesetzt.

Prolog für das Timed Token Protokoll

1. Zyklus: Bestimmen von T
2. Zyklus: Bestimmen der Last für synchrone Nachrichten

Abschätzen der Last:

$t_p = TTRT - O$: Zeit, die für den Transfer von Nutzlast für einen Zyklus zur Verfügung steht.

B: Informationsmenge in Bit/sek (Bandbreite des Netzes)

Jede Station darf einen bestimmten Anteil f_i an der zur Verfügung stehenden Übertragungskapazität in jedem Zyklus nutzen ($\sum f_i = 1$).

Quota der synchronen Nachrichten der Station i: $Q = f_i B t_p$

Wie wird erreicht, dass trotz des asynchronen Nachrichtenverkehrs $T=2TTRT$ garantiert werden kann ?

Timed Token Protocol

Die Summe der Quota synchroner Nachrichten mit

$$\sum f_i t_p \leq TTRT \quad \sum f_i \leq TTRT$$

beschränkt ($\sum f_i \leq 1$!).

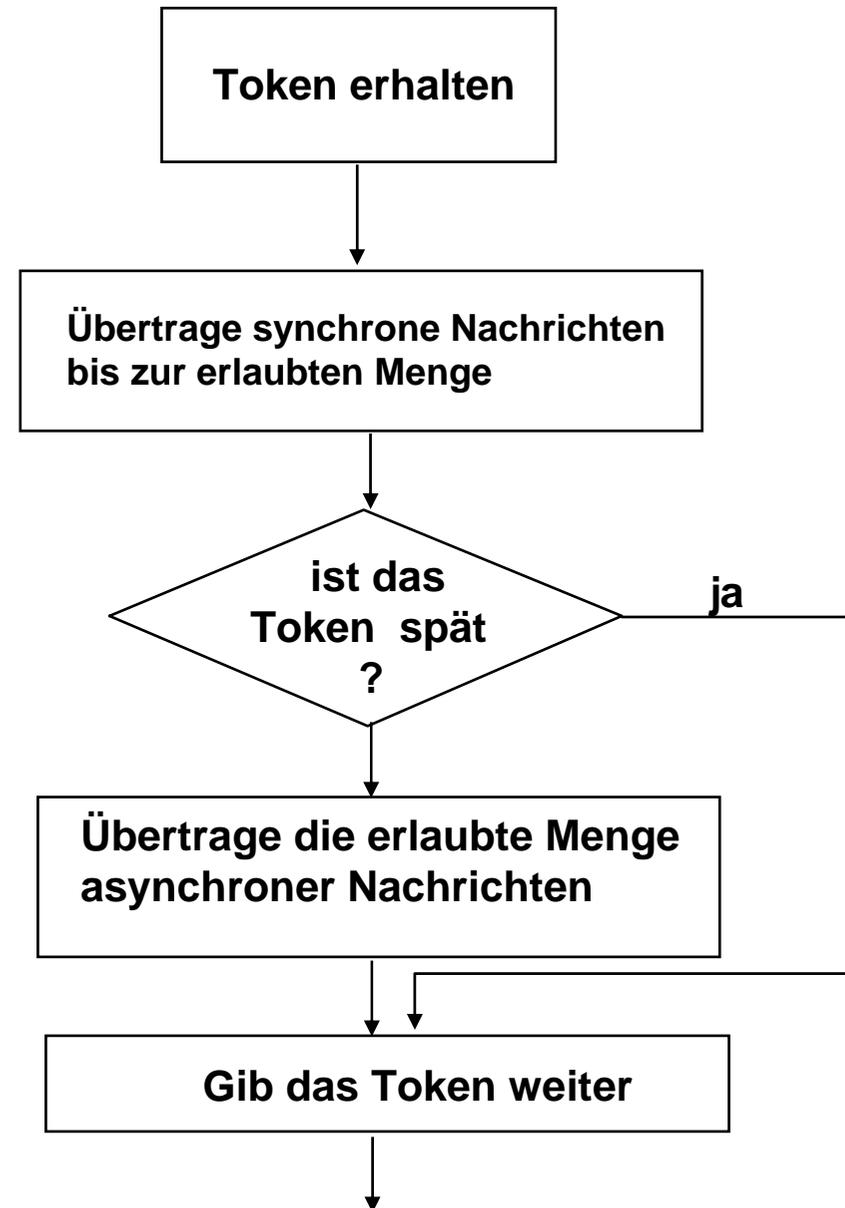
Wenn das Token ankommt, überprüft eine Station, wie viel Zeit seit seinem letzten „Besuch“ vergangen ist.

Def.:
Das Token ist:

früh, wenn: Zyklus $\leq TTRT$

spät, wenn: Zyklus $> TTRT$

gilt



Analyse des Timed Token Protokoll

Satz:

In einem fehlerfreien System ist die maximale Zeit, die zwischen zwei Besuchen des Tokens auf einer Station vergeht nicht höher als 2 TTRT.

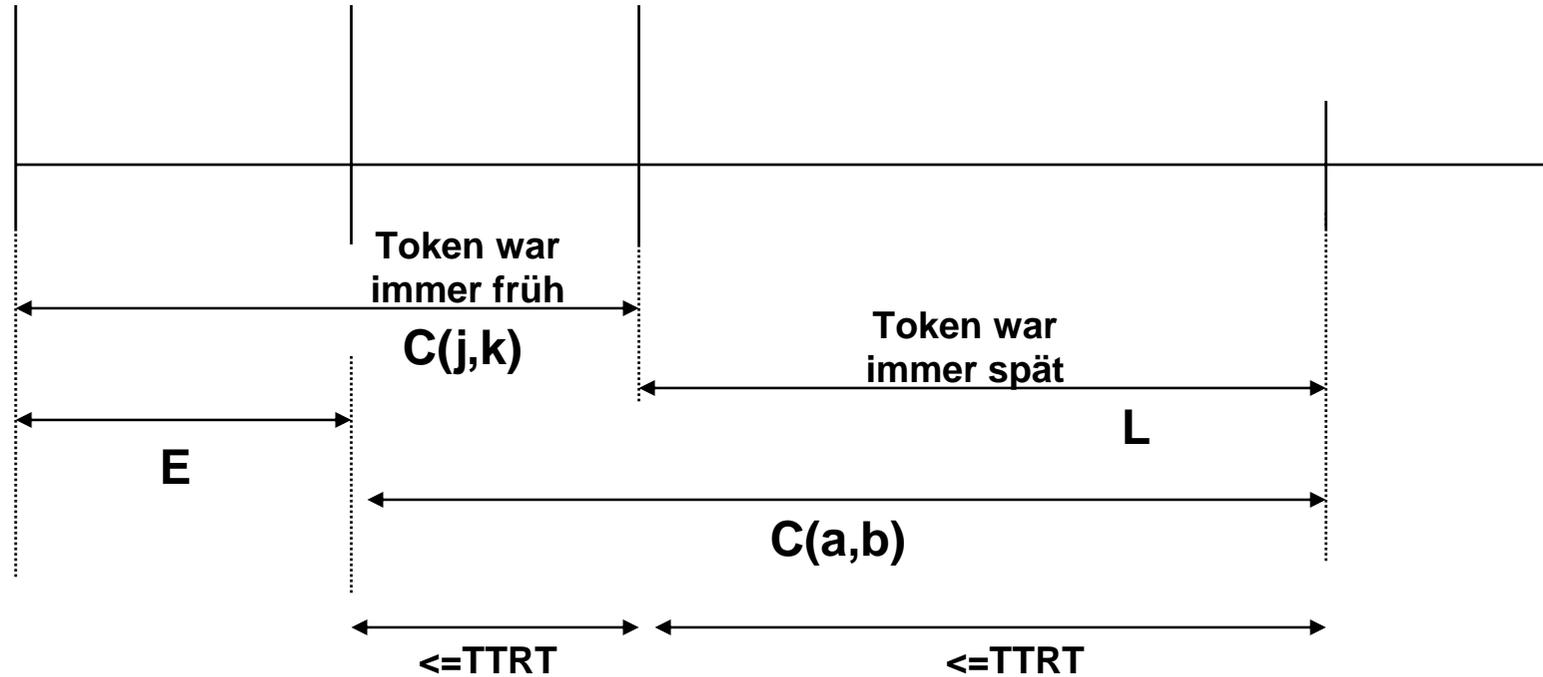
Beweisidee:

1. Fall: Das Token ist immer früh. Dann ist nach Def. von „früh“ nicht mehr als $TTRT < 2TTRT$ Zeit vergangen.
2. Fall: Das Token ist immer spät. Dann dürfen nur synchrone Nachrichten gesendet werden. Da die Summe der Quota synchroner Nachrichten mit $\sum f_i t_p \leq TTRT$ $\sum f_i \leq TTRT$ beschränkt ist ($\sum f_i \leq 1$), gilt die Annahme.

Analyse des Timed Token Protokoll

3. Fall:

$t(j-1,k)$ $t(a-1,b)$ $t(j,k)$ $t(a,b)$



$$C(j,k) \leq TTRT$$

$$C(a,b) = C(j,k) - E + L$$

$$\leq TTRT - E + \sum_{y,x = j,k \dots a,b} S(x,y)$$

$$\leq TTRT + \sum_{y,x = j,k \dots a,b} S(x,y)$$

$$\leq TTRT + TTRT = 2 TTRT$$

Da in L das Token immer spät ist,
werden nur synchrone Nachrichten
versendet.

$C(j,k)$: j-te Token-Zykluszeit für Knoten k; $\sum_{x,y = j,k \dots a,b} S(x,y)$ sind alle im Bereich L übertragenen synchronen Nachrichten.

Controlled Access:

Master/Slave

all control information in one place
maximum of control
easy to change

Single point of failure
More communication requirements
Central bottleneck

Global Time

Easy temporal co-ordination
Minimal communication overhead

Global knowledge of the calendar
All nodes have to conform to global time
Only critical messages

Token-based

Decentralized mechanism
Integration of critical and non-critical messages

Latency of messages
Long recovery time

Predictability of various Networks*

Worst Case Times of Inaccessibility*		t_{inacc} (ms)	
ISO 8002/4 Token Bus (5 Mbps)		139.99	Token-based Protocols
ISO 8002/5 Token Ring (4 Mbps)		28278.30	
ISO 9314 FDDI (100 Mbps)		9457.33	
Profibus (500 kbps)		74.80	
CSMA/CD	unbounded		CSMA Protocols
CSMA/CA	stochastic		
CAN-Bus (1Mbps)		2.48	

The worst-case-delay of the Timed-Token-Protocol** is $2 \cdot TTRT$ (Target Token Rotating Time)

* P. Verissimo, J. Ruffino, L. Ming: "How hard is hard real-time communication on field-busses?"

Advantages of CSMA-Networks for Real-Time Applications

- Low Latency of Messages
- Less “global” failures
- Short Recovery Time

Disadvantages:

- Collisions due to Decentralized Arbitration Mechanism

Problem: Worst Case Usable Bandwidth = 0

This may be just the case in safety critical situations!

Zusammenfassung : “Contention”- Protokolle

- **Verfahren funktionieren nur bei niedriger Netzlast zufriedenstellend**
- **Bei hoher Netzlast steigt die Zahl der Kollisionen an und damit der Anteil der erfolglos gesendeten Bits**
- **Da jede Nachricht nur als Ganzes gesendet werden kann, müssen diese Bits erneut gesendet werden.**
- **Bei hoher Netzlast kommt es zum Thrashing, d.h. die Bandbreite geht gegen 0**
- **Je früher eine Kollision erkannt wird, desto weniger Bandbreite geht verloren**

Wünschenswerte Eigenschaften eines Netzwerks:

Broadcast:

alle korrekte Knoten, die ein ungestörtes Telegramm (Frame) empfangen, empfangen dasselbe Telegramm.

Fehlererkennung:

korrekte Knoten erkennen jede Verfälschung eines empfangenen Telegramms, die vom Netzwerk verursacht wird.

Ordnung:

jeweils zwei Telegramme, die von zwei korrekten Knoten empfangen werden, werden in derselben Reihenfolge empfangen.

Full Duplex:

gesendete Telegramme werden vom sendenden Knoten empfangen.

Beschränkte Anzahl von Omissions (k):

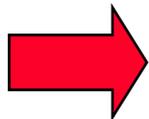
in einem bekannten Zeitintervall treten Fehler in höchstens k Telegrammen auf.

Gleichmäßigkeit (Tightness):

Knoten, die ein unverfälschtes Telegramm empfangen, empfangen es zu Zeitpunkten, die höchstens T_{tight} auseinanderliegen.

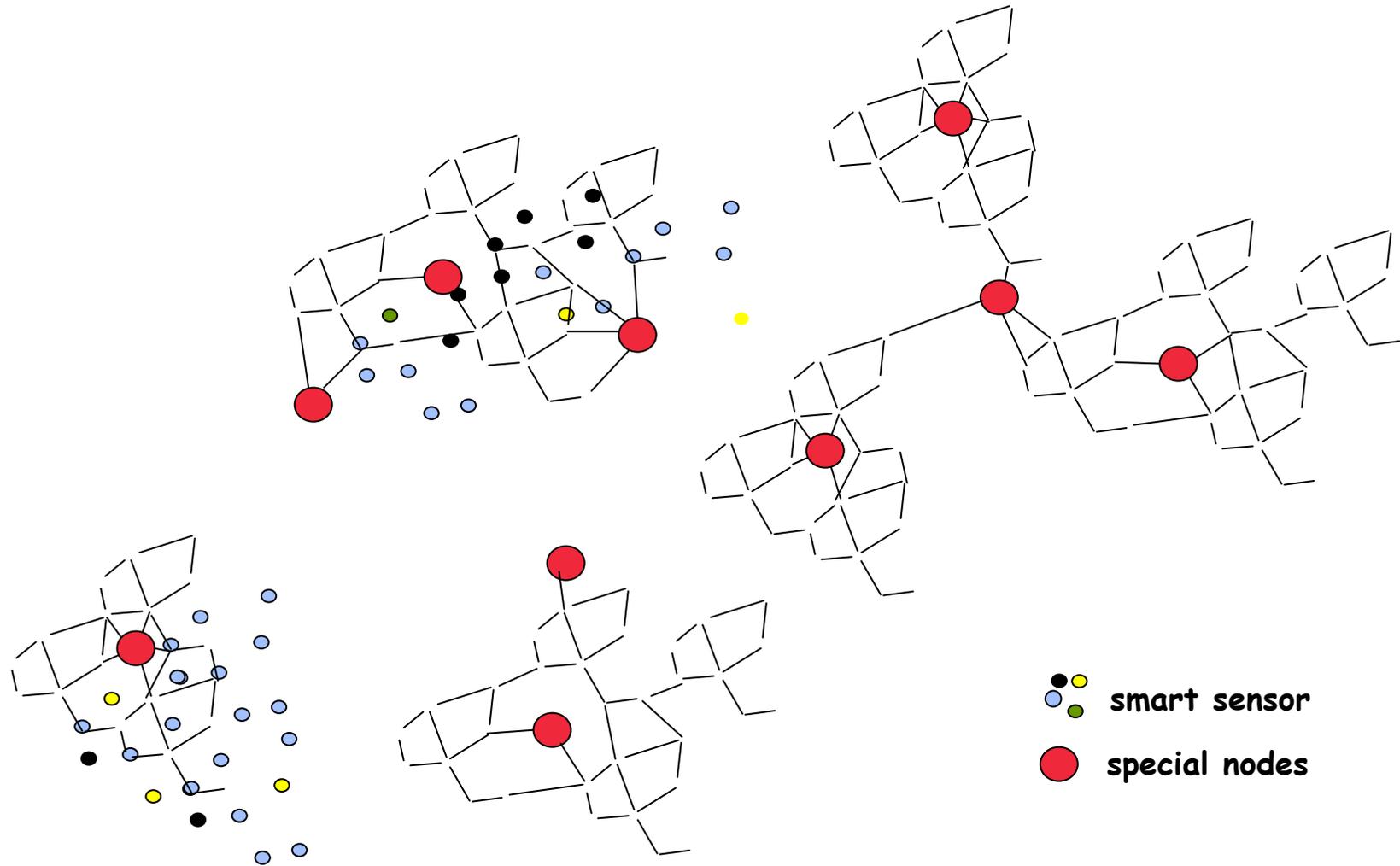
Beschränkte Nachrichtenverzögerung:

Jedes Telegramm wird vom Netzwerk mit einer beschränkten Zeitverzögerung versendet, von der Sendeaufforderung an gerechnet .



Eigenschaften eines Abstrakten Netzwerks

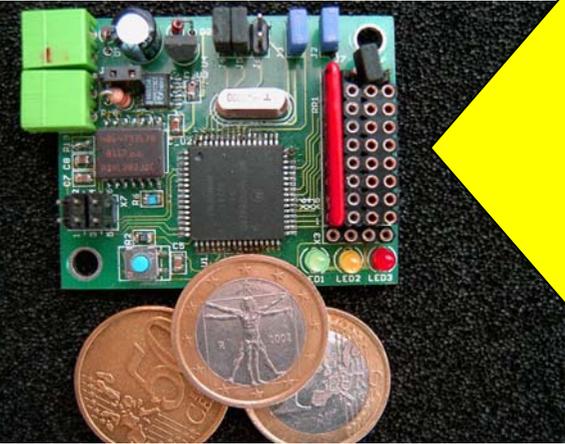
Sensornets for a wired physical world



Hardware for Sensor Networks

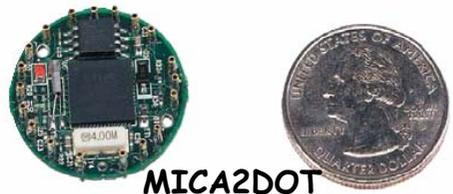
Developed Sensors at CORE

- infrared motion detector
- infrared distance sensor
- acceleration sensor
- embedded gyro
- weather station
- magnetic field detector
- in-house location system

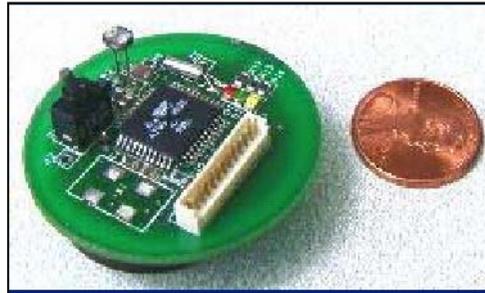


any-board, CORE, Ulm

a mica mote,
Berkeley, Crossbow



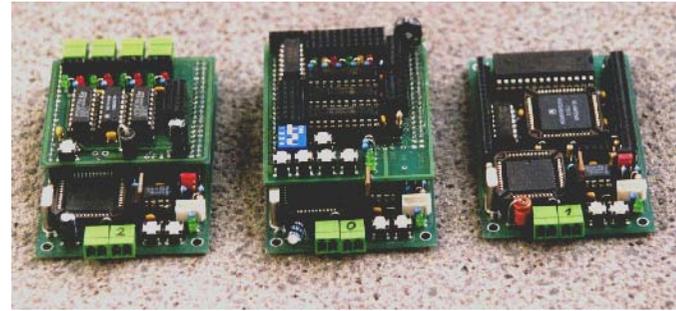
MICA2DOT



WeC „Smart Rock“ UCB

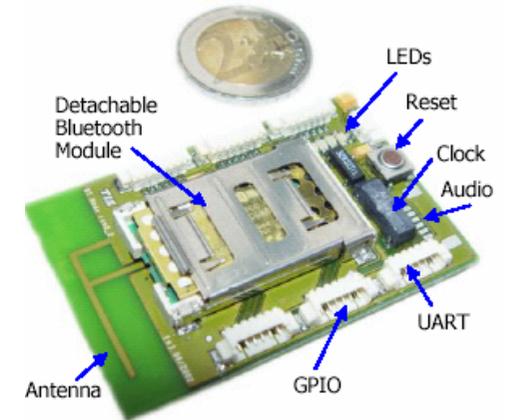


The EYES prototype



68HC11 CAN-Sensor Boards, CORE, Ulm

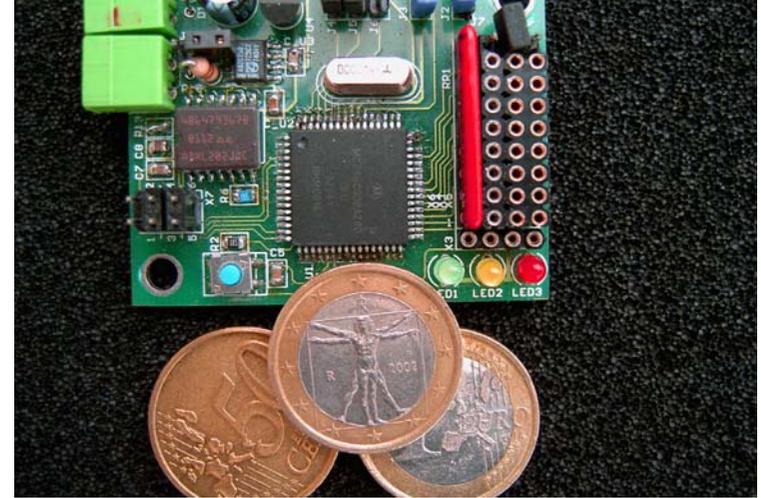
Smart-its: ETH Zurich,



cpu, memory
on back side

Tiny Properties

Designed for experimentation:
Basic Board + Piggyback extension



Basic board:

- Processor 68HC908AZ60 (60k Flash, 2k RAM)
- Power regulator (linear or switched) 6-14 V
- LEDs for checks, configuration jumpers
- CAN-Bus Network Interface
- Sockets for AD, C&C, digital I/O
- Sockets for asynch. and synch serial comms.

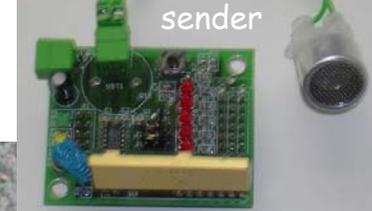
Power consumption:

- Processor ~ 250 mW @ 16MHz
- Radio link (Easy Radio, 19kbit/sec): ~150mW(transmit), ~75mW(idle)
- 9V Block (565 mAh): ~ 8h@continuous operation, ~30 days@10ms/sec

the TINY family



Magnetic field detector



weather station:
pressure, temperature,
humidity, light intensity

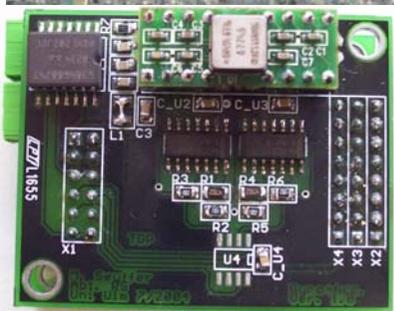
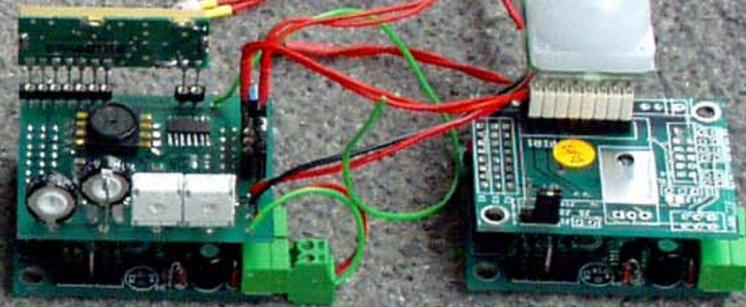
PIR: Passive InfraRed
motion detector



Wireless modem for
PC serial line



TINY with on-board
acceleration/
inclination sensor



gyro



TINY basic board: 68HC908 AZ60: 60 k Flash, 2 k RAM,
CAN, serial, extension ports for piggy-back boards.



Low Power Wireless
comm. board.

TINY Family: Motorola 68HC908AZ60A, CAN-Bus, Serial Line, Port for Extension Boards, phys. size: 50x40 mm, power consumption: approx. 55 mA @ 16MHz, Power input: 6-12 V, linear (< 100mA) or switched power regulator. Available sensor/actuator extension boards: PIR, weather, distance, acceleration, DC motor control, radio transceiver

Eigenschaften:

Multi-hop
Unbekannte Topologie
Selbstorganisierend
Energieeffizient

Großes Problem: Kollisionen sind teuer, weil sie erst am Ende einer Übertragung erkannt werden können.

MAC-Protokoll sollte Kollisionen vermeiden!

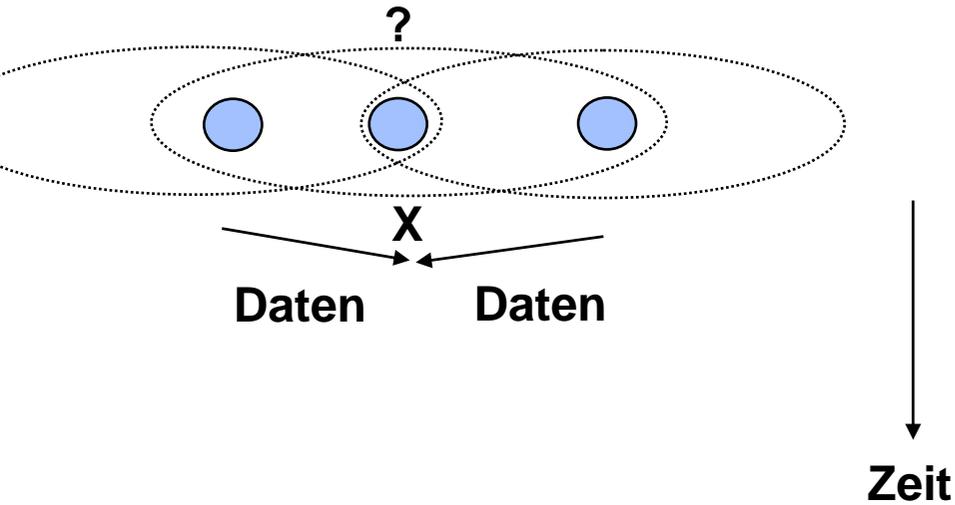
- **Zentrale Instanz**
- **Plan**
- **mehrere Kanäle**
- **dezentrale Arbitrierungsmechanismus**

Zusätzliches Problem: Aktives Mithören ist nicht energieeffizient.

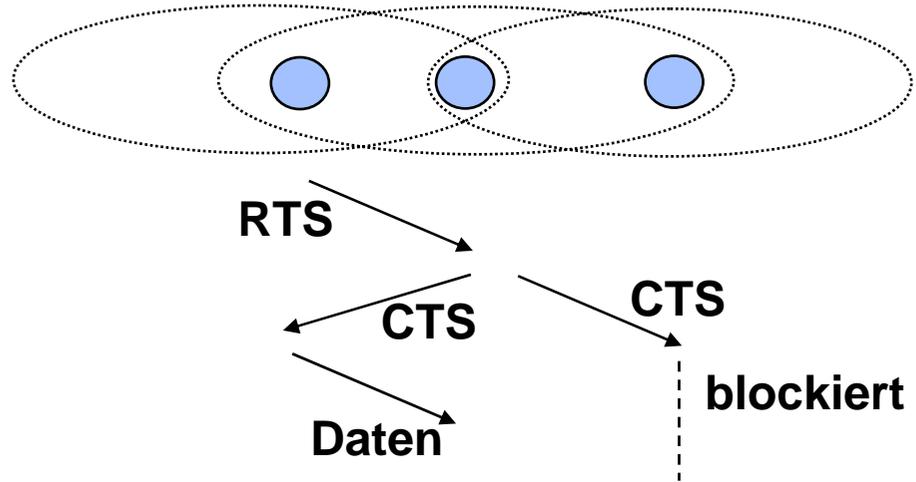
- **gelegentliches, periodische Überprüfung des Kanals**
- **festgelegte Schlaf- und Wachperioden**
- **zusätzlicher ultra-low-power Weckkanal**

mehr Probleme

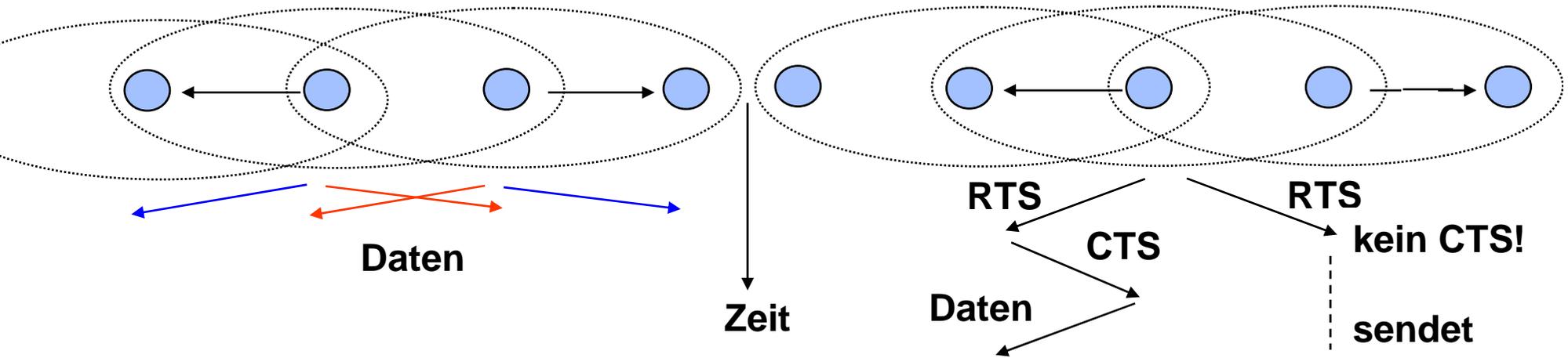
Hidden Terminal Problem



Multiple Access with Collision Avoidance (MACA)



Exposed Terminal Problem

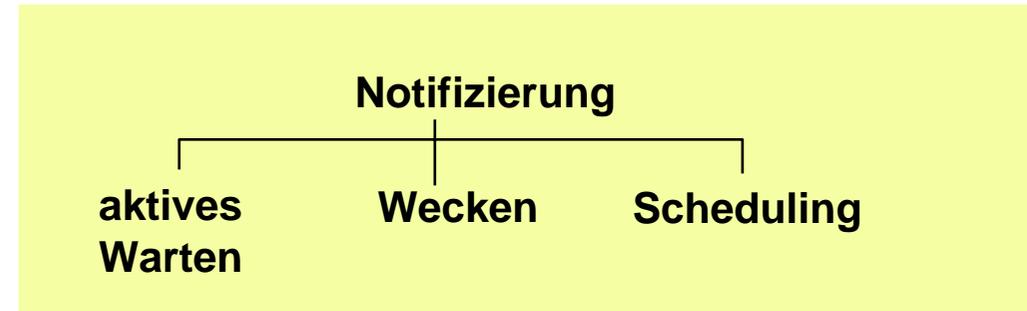
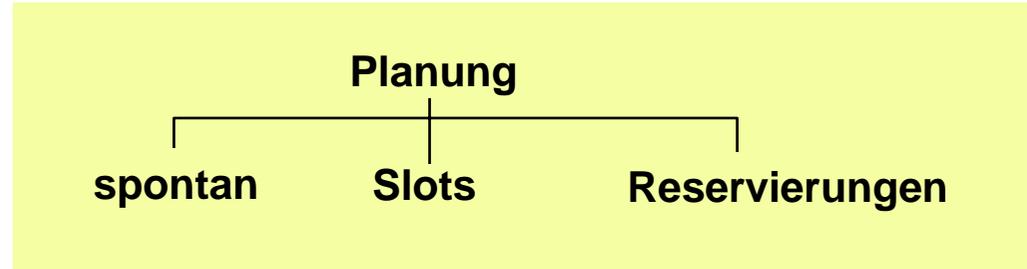
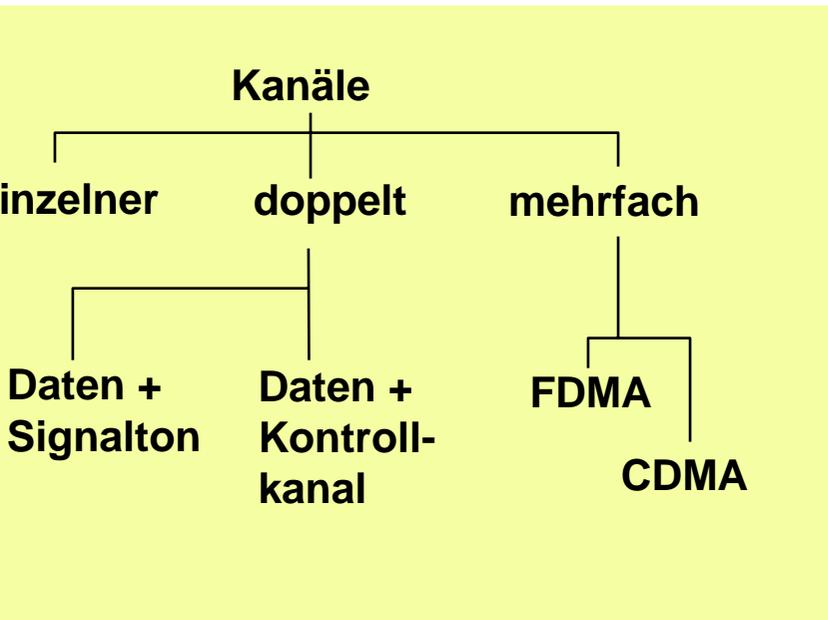


Zielkonflikte und Entwurfsentscheidungen

Einen oder mehrere physische Übertragungskanäle

Geplante oder ungeplante Übertragung

Notifizierungsart



MAC-Prinzipien

Contention-Based

Slot-Based

Reservation-Based

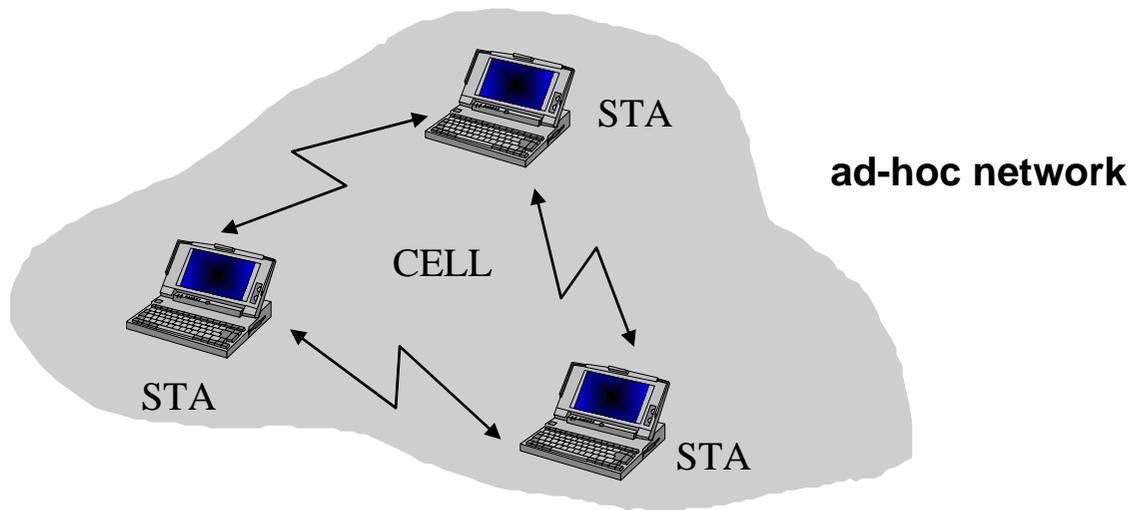
MAC-Prinzipien

	Sendebedingung	Startzeit	Kanäle
einfaches) Aloha	Datenverfügbarkeit	beliebig	1
Slotted Aloha	Zeitschlitz,	Anfang Zeitschlitz	1
MACA	RTS/CTS	dyn. Reservierung	1
MACAW	MACA + Acknowledge	wie MACA	1
CSMA	Medium nicht belegt	beliebig	1
CSMA/CA	Medium nicht belegt	nach Wartezeit,	1
		nach dyn. Reserv.	
TDMA	nach Plan	festgelegter Rahmen	1
FDMA	verschiedene Frequ.	beliebig	m
CDMA	versch. orth. Codes	beliebig	m

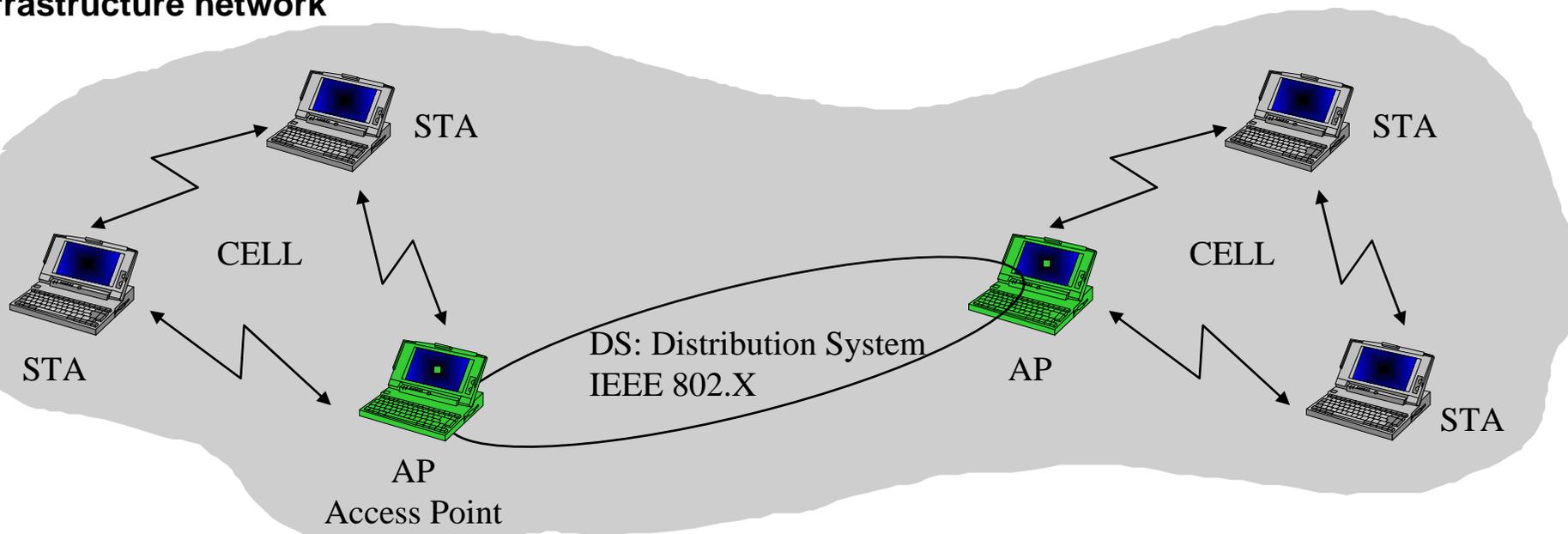
Organisation: beliebig, Zeitschlitz, exklusiver Zeitrahmen

Notifizierung: durch Mithören, Wecken, Planen

Network Types

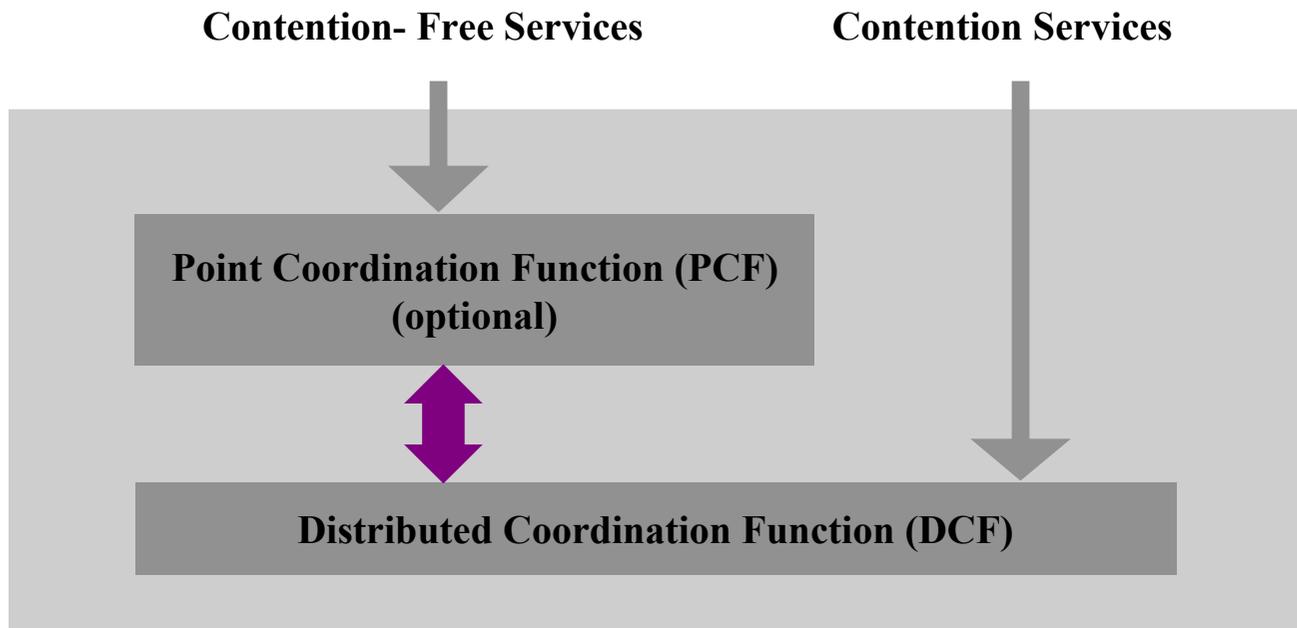


infrastructure network

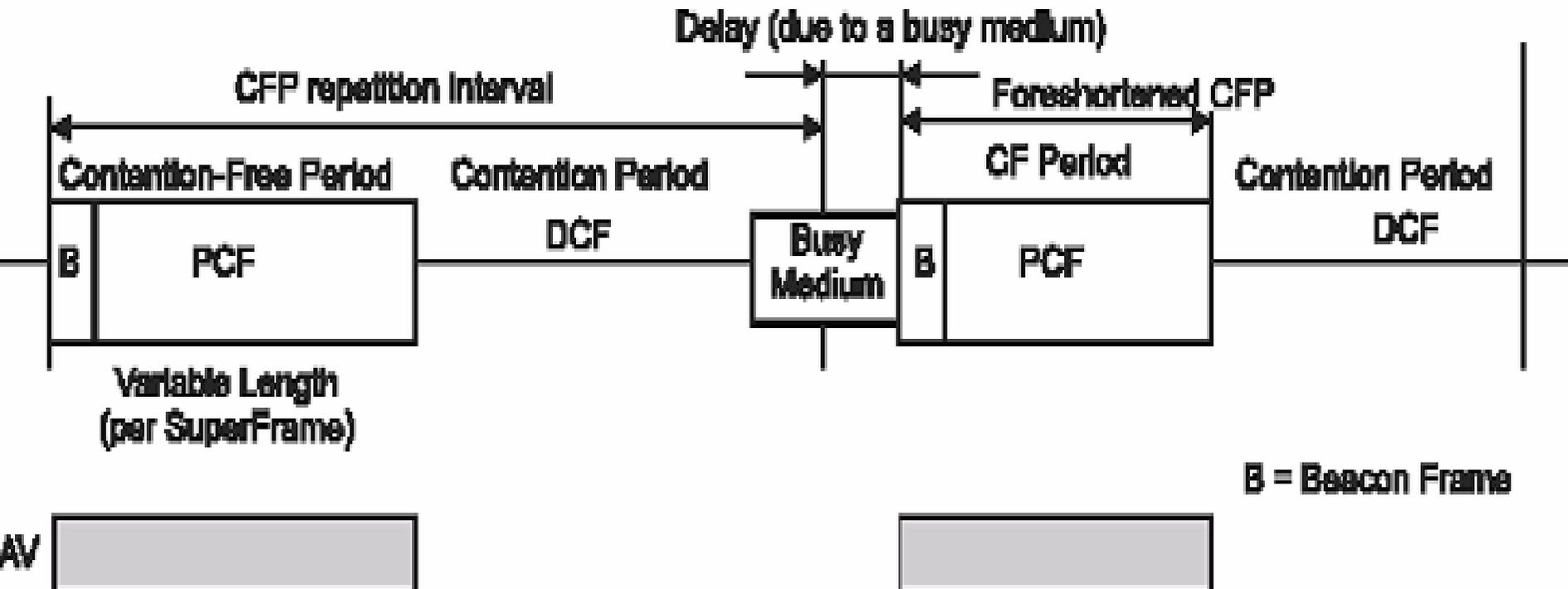


IEEE 802.11 MAC Layer

MAC Architektur:



Alternation of PCF and DCF

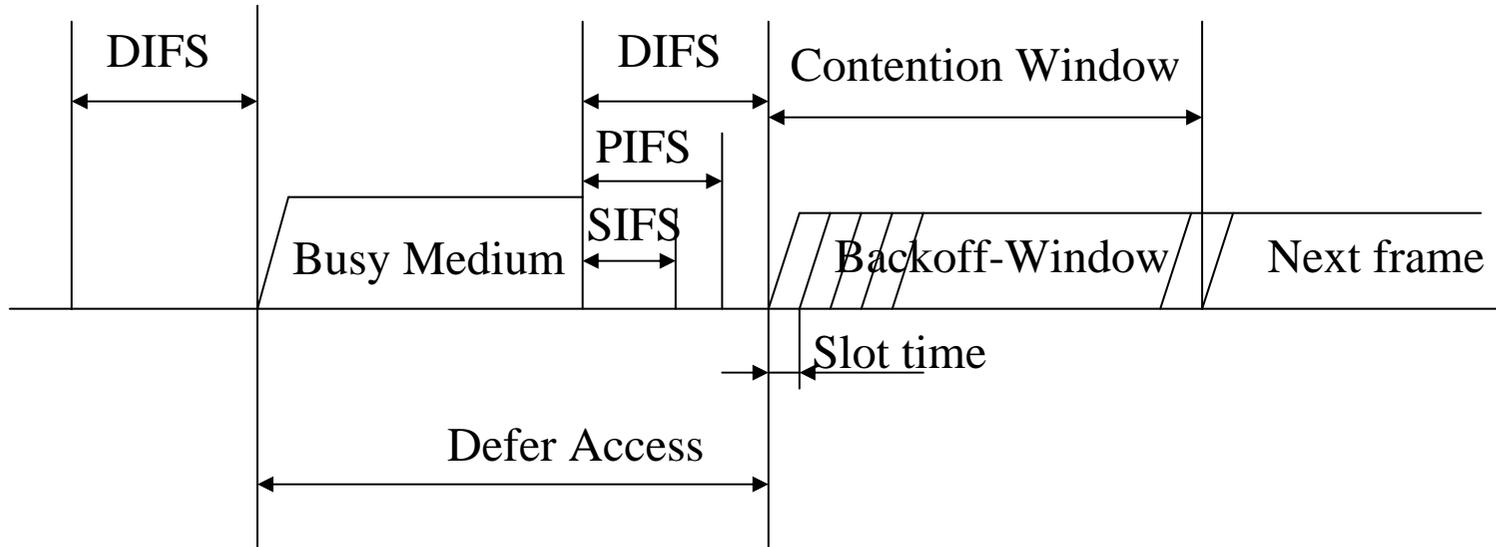


Distributed Coordination Function (DCF)

- CSMA/CA Protocol
- Collision Avoidance by random backoff procedure (p-persistent)
- All Frames are acknowledged, lost Frames are resend
- Priority Access by Interframe Space (IFS)

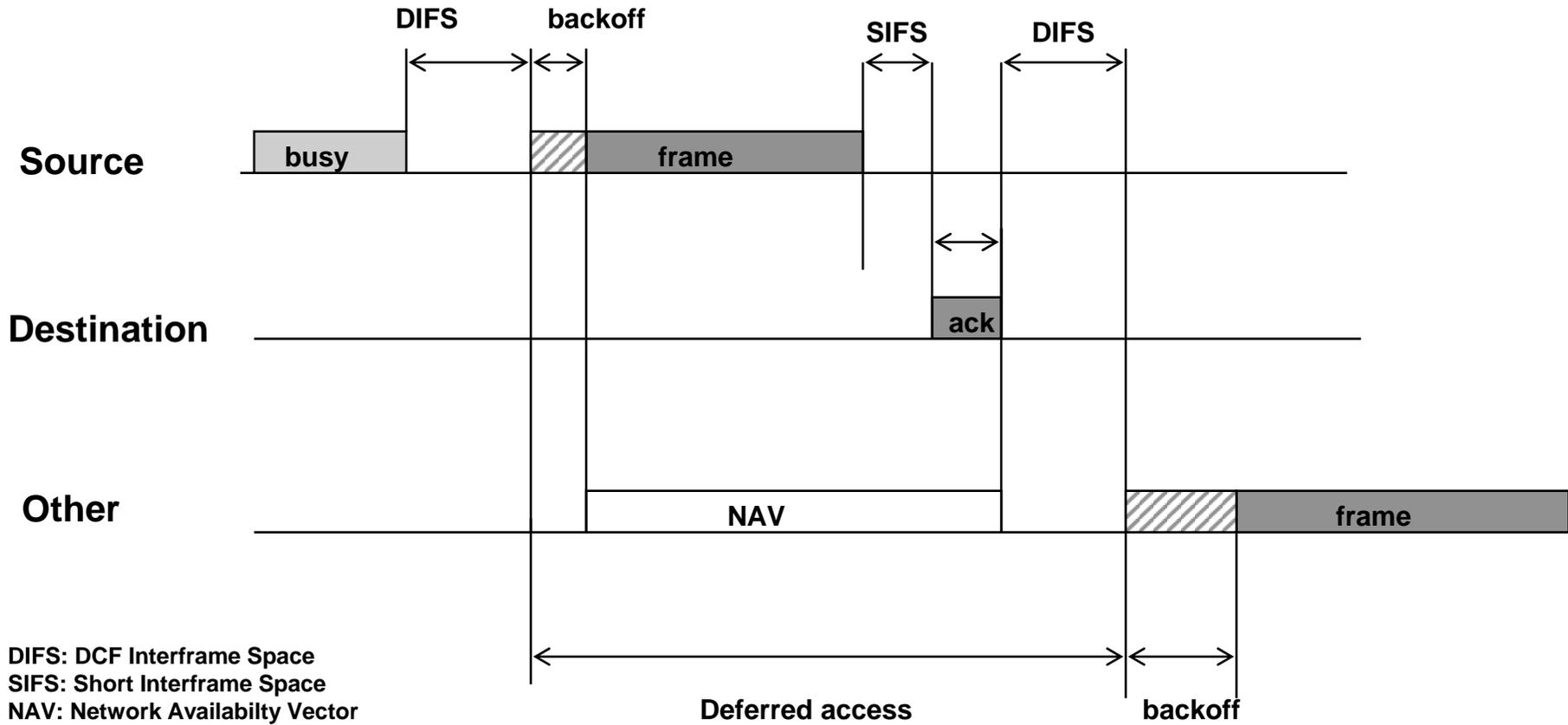
=> fair arbitration but no real-time support

Relationship of different IFSs in 802.11



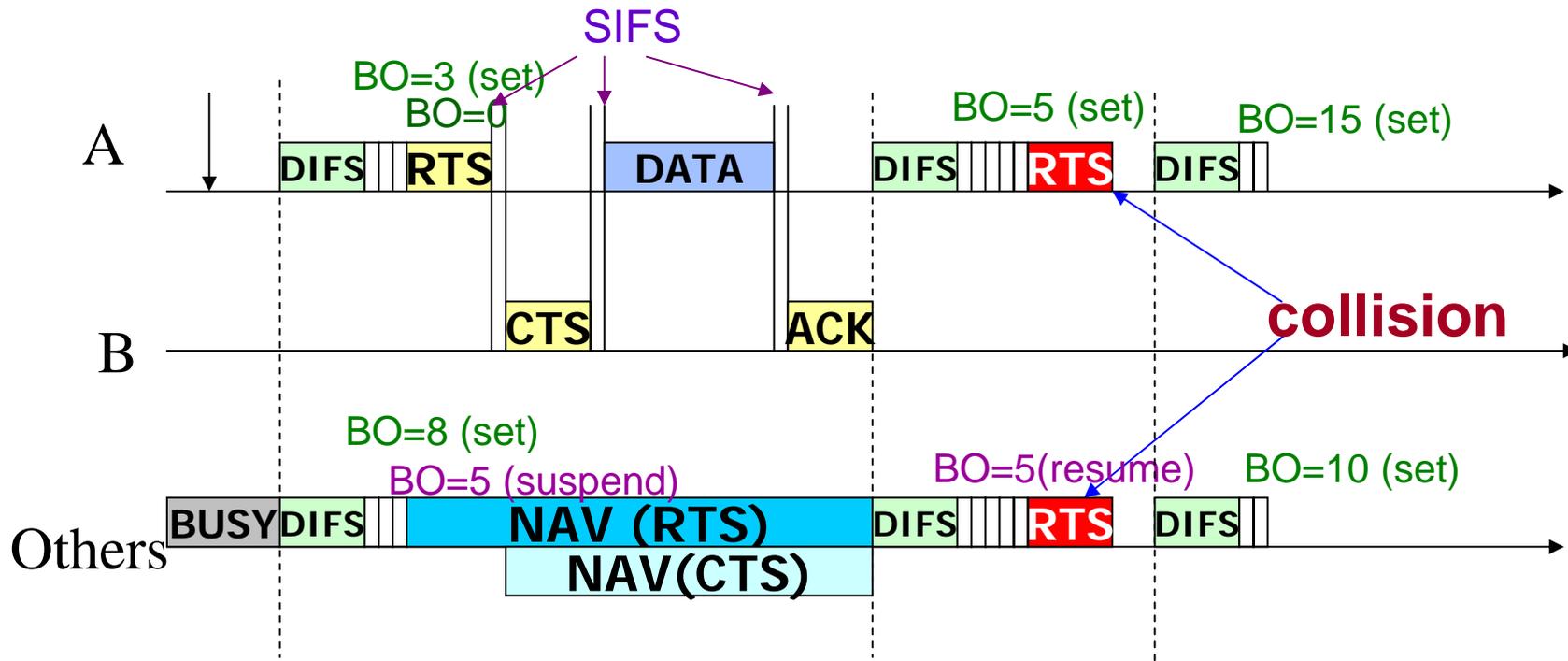
DIFS: DCF Interframe Space
PIFS: PCF Interframe Space
SIFS: Short Interframe Space

Basic DCF transmission protocol



Properties: Point-to-point, every frame is acknowledged

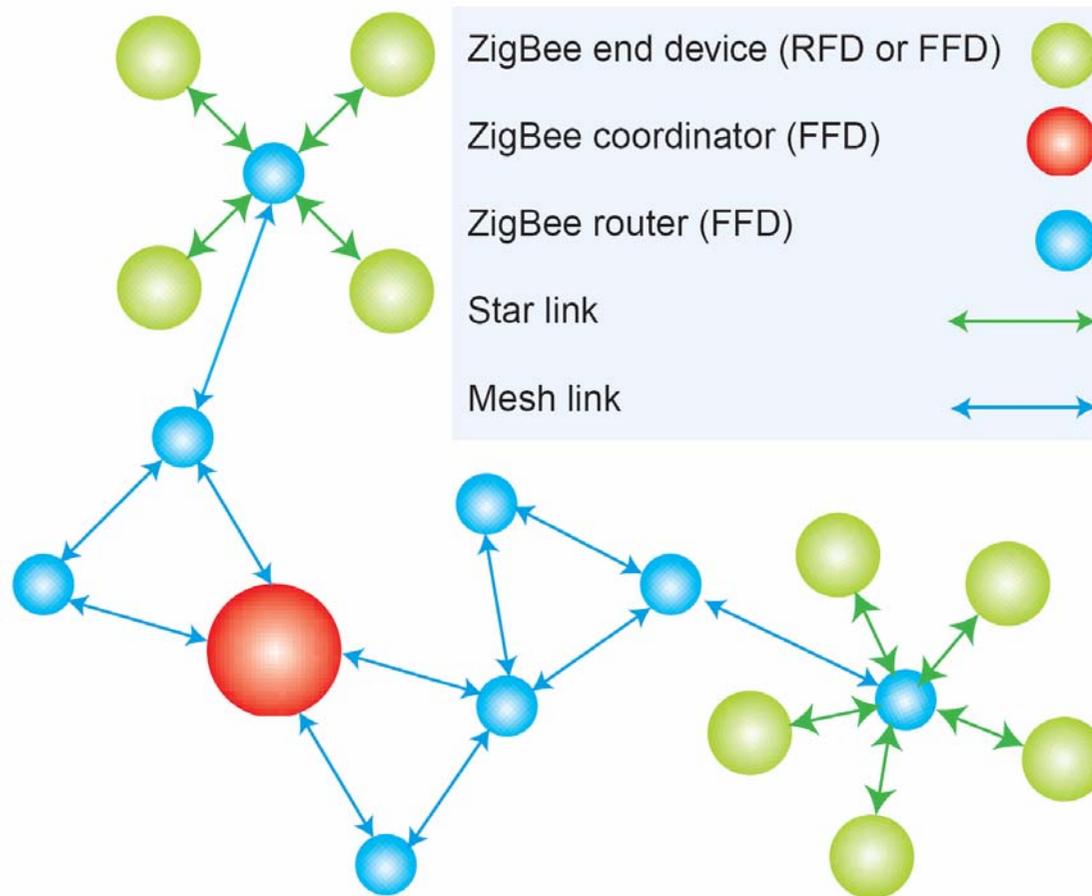
Example of 802.11 RTS/CTS/DATA/ACK Scheme



BO: backoff

ZigBee

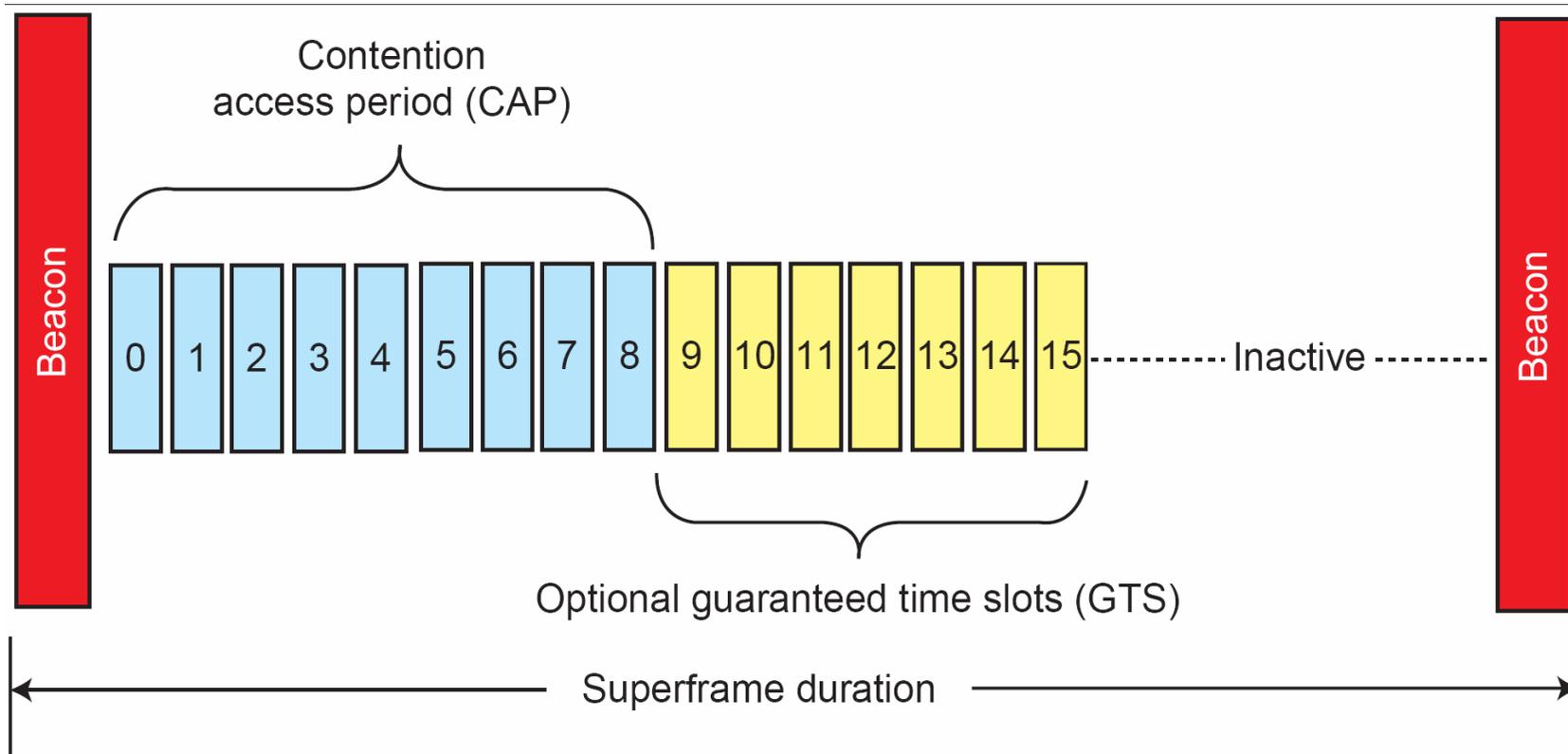
(based on the IEEE 802.15.4 standard)



Basic MAC Operation

Contention Period: CSMA/CA based on a slotted protocol

Contention Free Period: Reserved slots



Kenngrößen für drahtlose Übertragung

	EasyRadio	RFMonolithics TR 1001	ChipCon CC1000	Lucent WLAN PC "Silver"
Frequenz	868 MHz	868 MHz	868 MHz	2,4 GHz
Bitrate (Kbps)	19	115,2	76,8	11.000
Energieverbrauch				
senden (mA)	17	12,0	25,4	284,0
empfangen(mA)	8	3,8	11,8	190,0
standby(µA)	--	0,7	30,0	10.000,0
Umschaltzeiten(µs)				
standby-transmit		16	2000	
receive-transmit		12	270	
standby-receive		518	2000	
transmit-receive		12	250	
transmit-standby		10		
receive-standby		10		

Quellen für erhöhten Energieaufwand:

- Aktives Warten:** wenn ein Knoten nicht weiss, wann er eine Nachricht erwarten kann, muss er ständig auf Empfang bleiben.
- Mithören:** (Overhearing) Ein Knoten empfängt Nachrichten, die nicht für ihn bestimmt sind. Besser: während dieser Zeit abschalten.
- Kollisionen:** Energie, die durch Senden während einer Kollision aufgebracht wird, ist verloren. Das entsprechende Paket muss erneut gesendet werden. Kollisionen können nicht während des Sendens erkannt werden.
- Protokollaufwand:** Jede zusätzliche Massnahme wie RTS/CTS oder ein Acknowledge erhöhen den Protokollaufwand gegenüber den Nutzdaten.
- Dynamische Fluktuationen:** Ungleich verteilte Last auf dem Netz erhöht die Wahrscheinlichkeit von Kollisionen unter Hoch- oder Überlastsituationen (Thrashing).

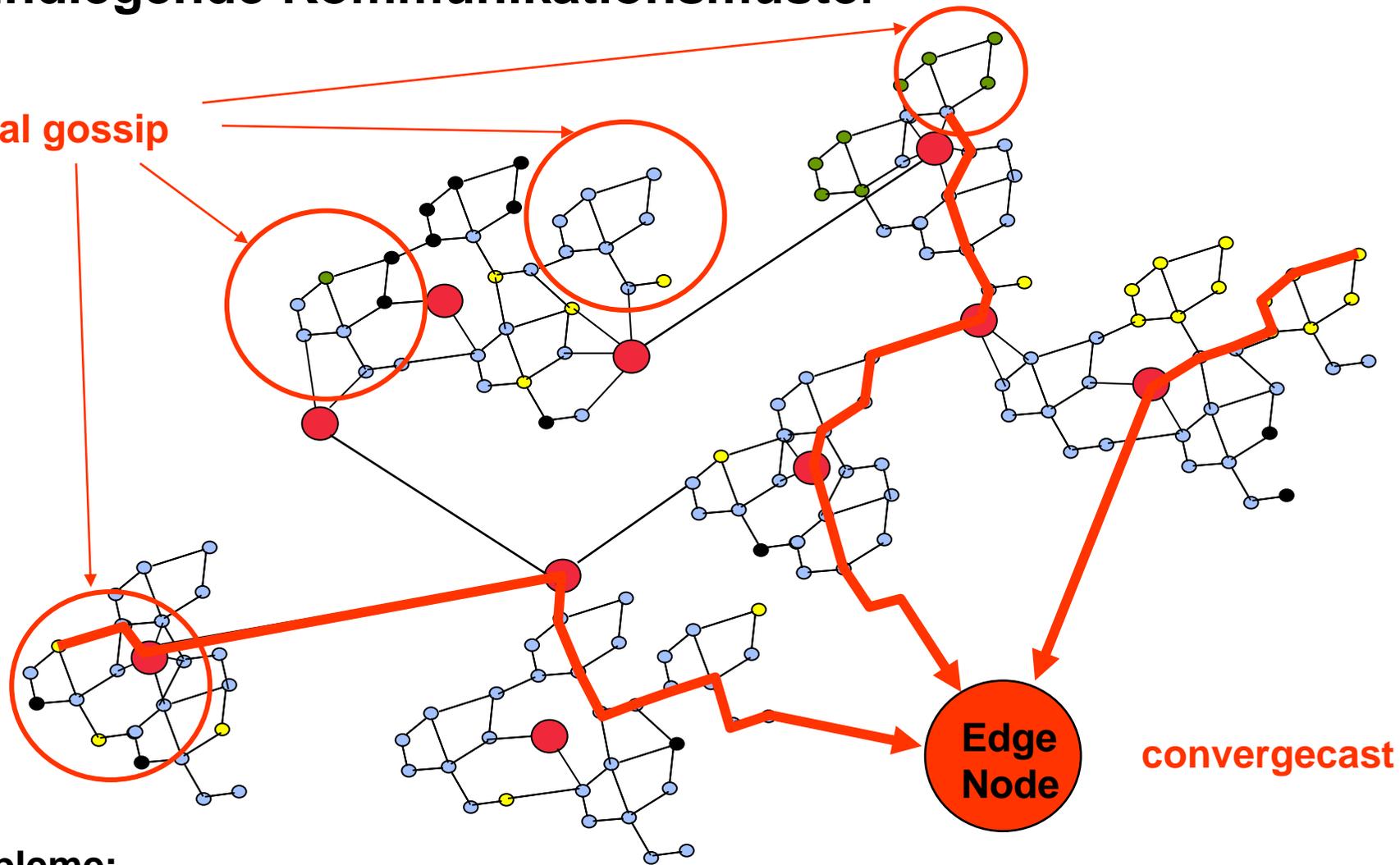
Größtes Problem:

idle listening

**Energiesparende Protokoll versuchen die
Zeit des inaktiven Mithörens zu minimieren!**

Grundlegende Kommunikationsmuster

local gossip



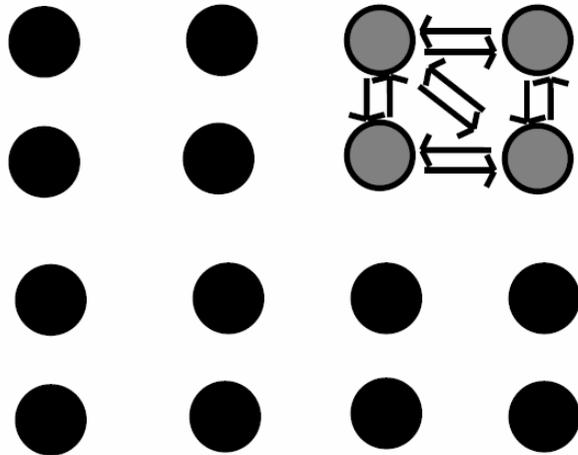
Probleme:

Faire Verteilung der Last

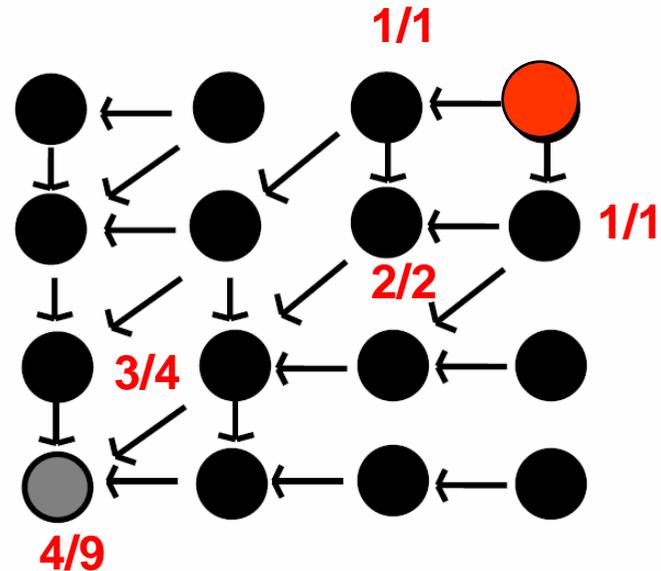
Optimales Routing

Vermeidung von Partitionen

Grundlegende Kommunikationsmuster



local gossip

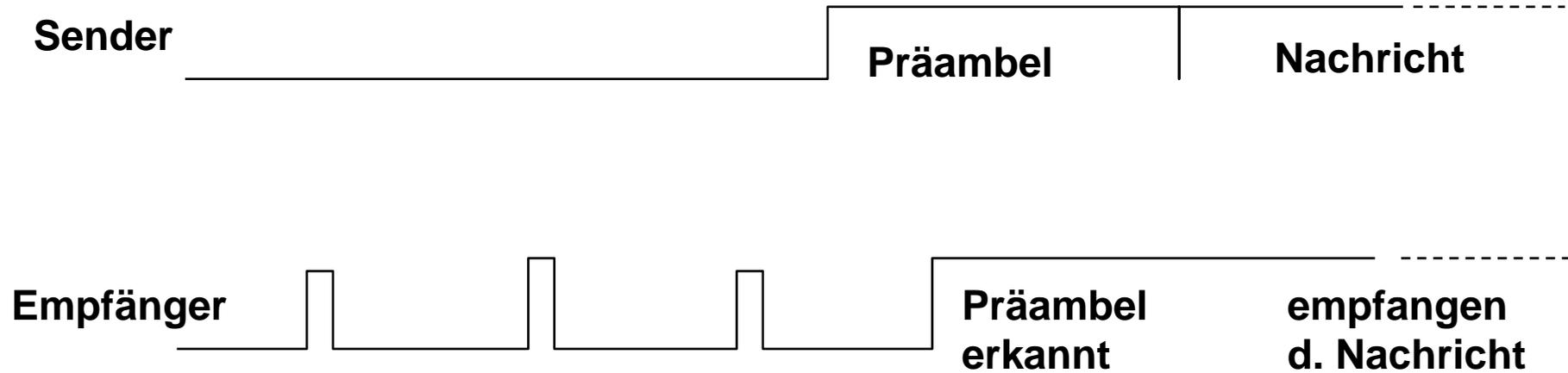


convergecast

a/ b: a= min # of hops
b= # of different routes

Variations: Low Power Listening

.)



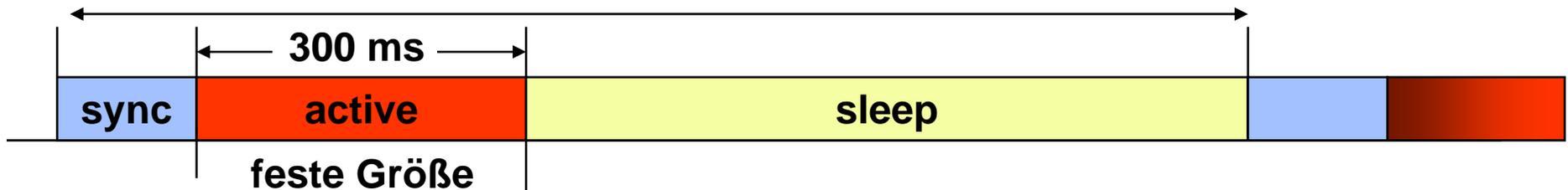
2.) Sender weiss, wann der Empfänger bereit ist. Zeitliche Synchronisation!

J. Hill, D. Culler: MICA: A wireless platform for deeply embedded networks. IEEE Micro 22(6), Nov. 2002

A. El-Hoiyi: Aloha with preamble sampling for sporadic traffic in ad-hoc wireless sensor networks, IEEE Int. Conf. on Comm. (ICC) New York, Apr. 2002

Slotted Protocols

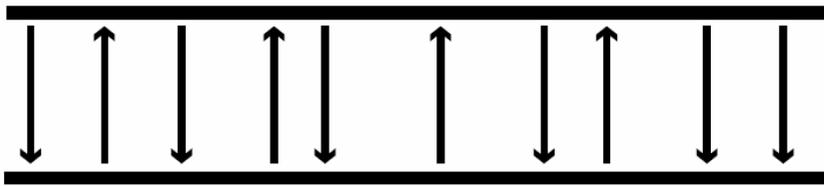
Slot Länge: beliebig aber fest (0,5 - 1 sec)



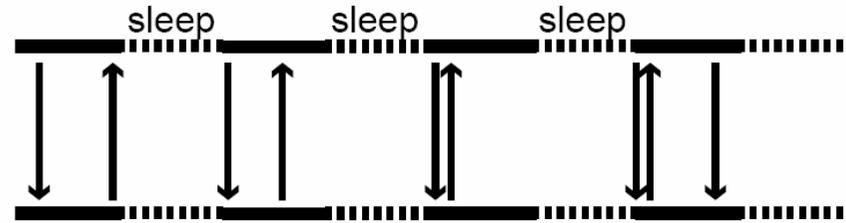
Example: **S-MAC (Sensor -MAC)** (Ye, Heideman, Estrin)

Knoten sind in (virtuellen) Clustern organisiert, die ein gemeinsames Slot-Format annehmen.

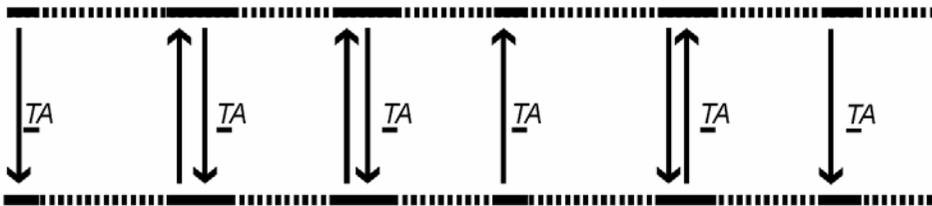
Variation **T-MAC (Time-out MAC)**: Adaptive Festlegung des Verhältnisses von active und sleep Intervall. Falls nach einem kurzen Intervall nichts gesendet wird, kann sofort in Sleep übergegangen werden.



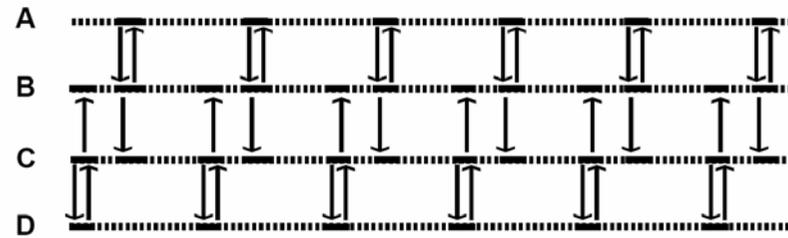
CSMA



S-MAC

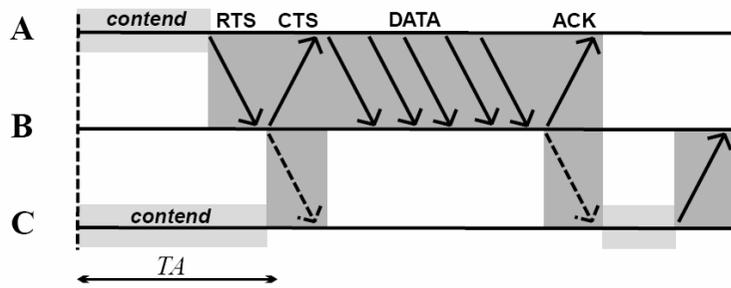


Basiszyklus

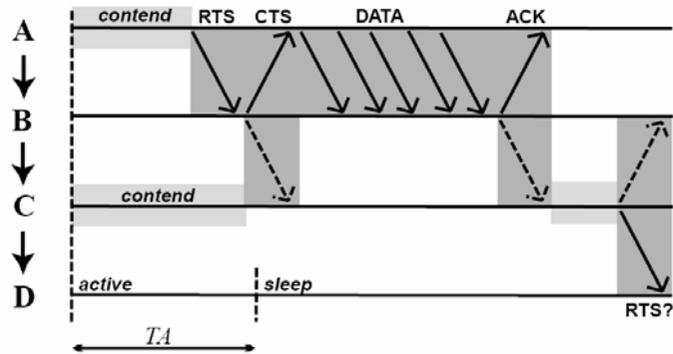


Kommunikation zwischen virtuellen "Cluster"

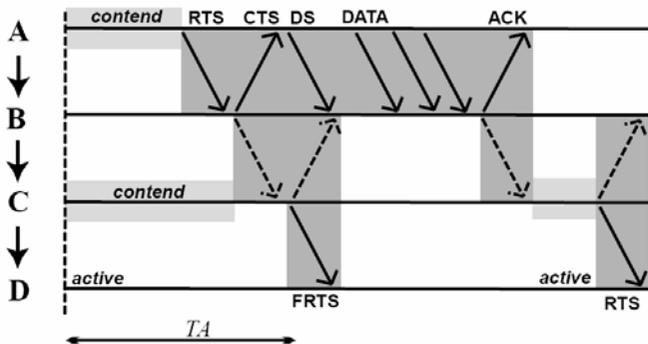
T-MAC



Basic Cycle



Early Sleeping Problem.
Node C goes to sleep before node B can send the RTS.



Future Requests to Send (FRTS)

As the FRTS packet would disturb the data packet that follows the CTS, the data packet must be postponed for the duration of the FRTS packet. To prevent any other node from taking the channel during this time, the node that sent the initial RTS (node A in Figure 3.5) transmits a small Data-Send (DS) packet. After the DS packet, it must immediately send the normal data packet. Since the FRTS packet has the same size as a DS packet, it will collide with the DS packet, but not with the following data packet. The DS packet is lost, but that is no problem: it contains no useful information.

Wie kommt die Information von der Quelle zum Interessenten?

Problem: Bei der Installation des Netzes ist die Topologie nicht bekannt!

Mögliche Lösungen:

Topologie-Erkennung

Inhalts-Basiertes Routing

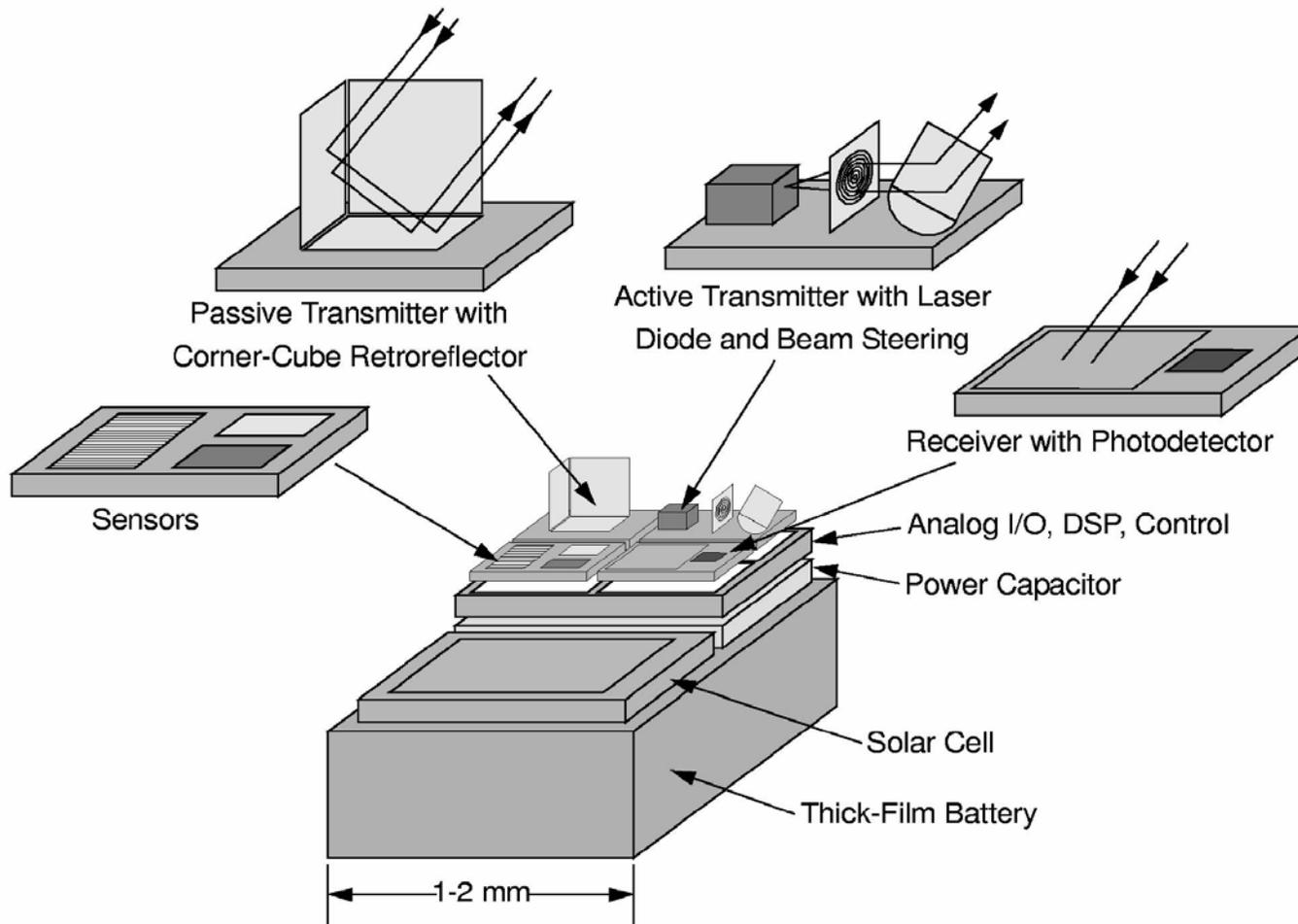
Directed Diffusion

Intelligenter Staub



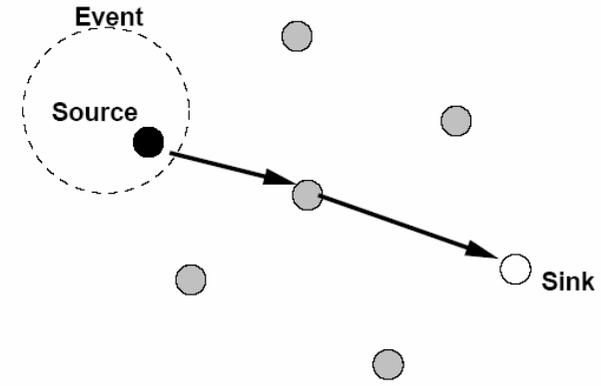
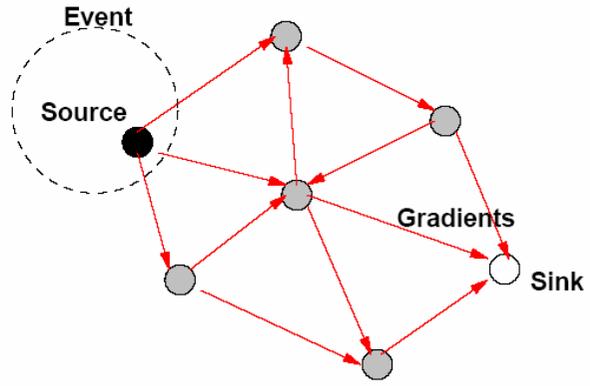
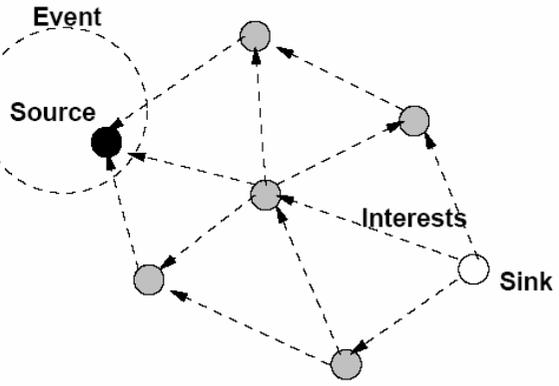
image source: Hitachi

"Smart Dust"



Sensor, Kontrollhardware, Batterie, DSP, passive Kommunikation,

Directed Diffusion (Intanagonwiwat, Govindan, Estrin)



Task: activity of a sensor node

```

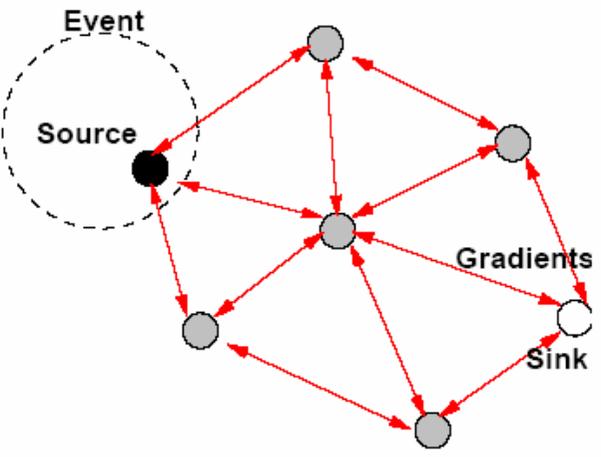
type = four-legged animal // detect animal location
interval = 20 ms // send back events every 20 ms
duration = 10 seconds // .. for the next 10 seconds
location = [-100, 100, 200, 400] // from sensors within rectangle
    
```

Event: information description

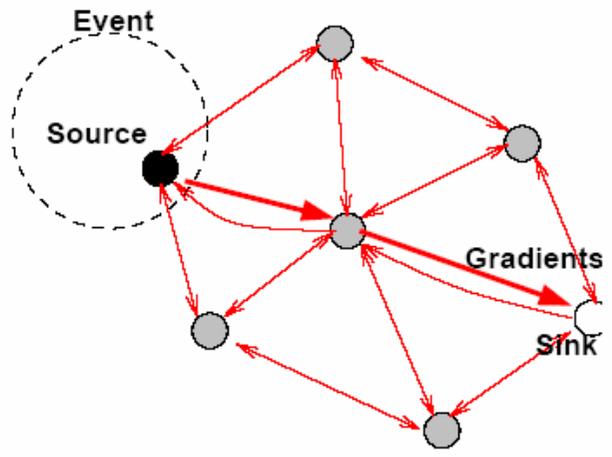
```

type = four-legged animal // type of animal seen
instance = elephant // instance of this type
location = [125, 220] // node location
intensity = 0.6 // signal amplitude measure
confidence = 0.85 // confidence in the match
timestamp = 01:20:40 // event generation time
    
```

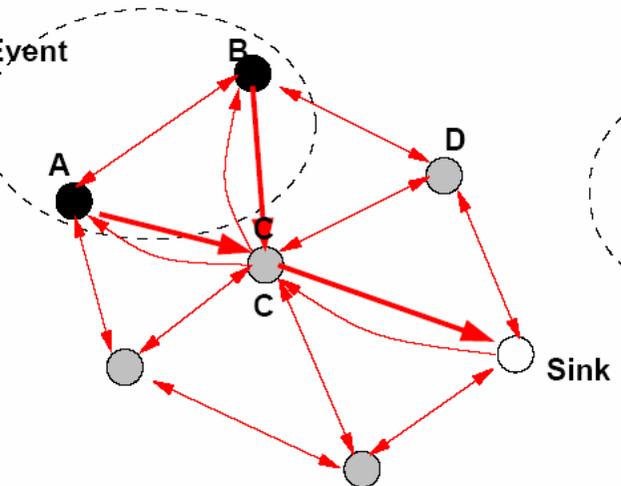
Aspekte des "Directed Diffusion"



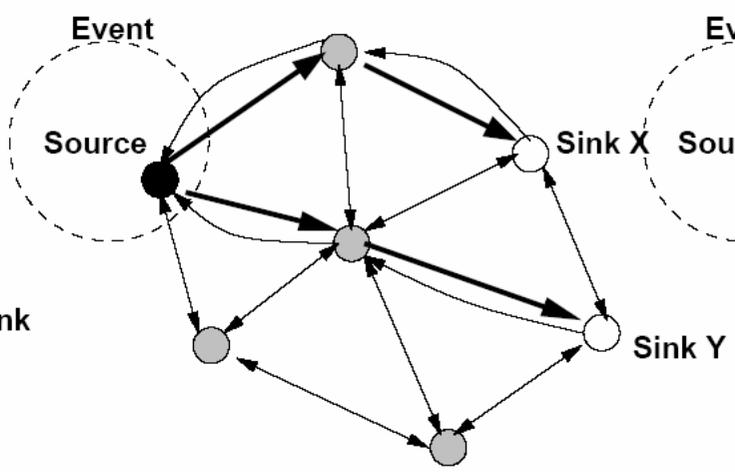
Gradient Establishment



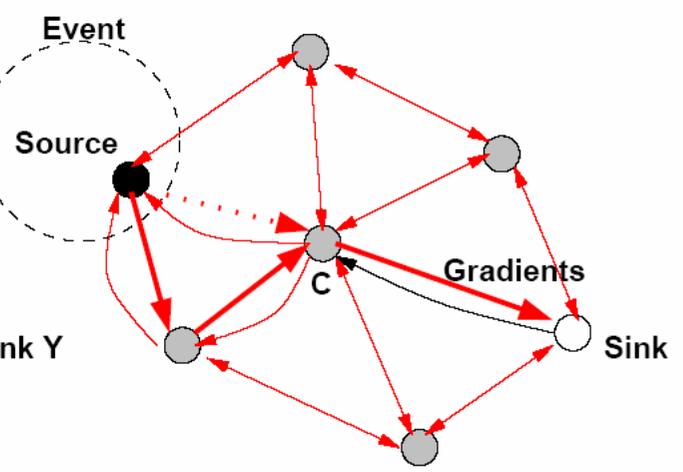
Reinforcement



Multiple Sources



Multiple Sinks



Repair