Braided Ring

Ringing out Fault Tolerance. A New Ring Network for Superior Low-Cost Dependability

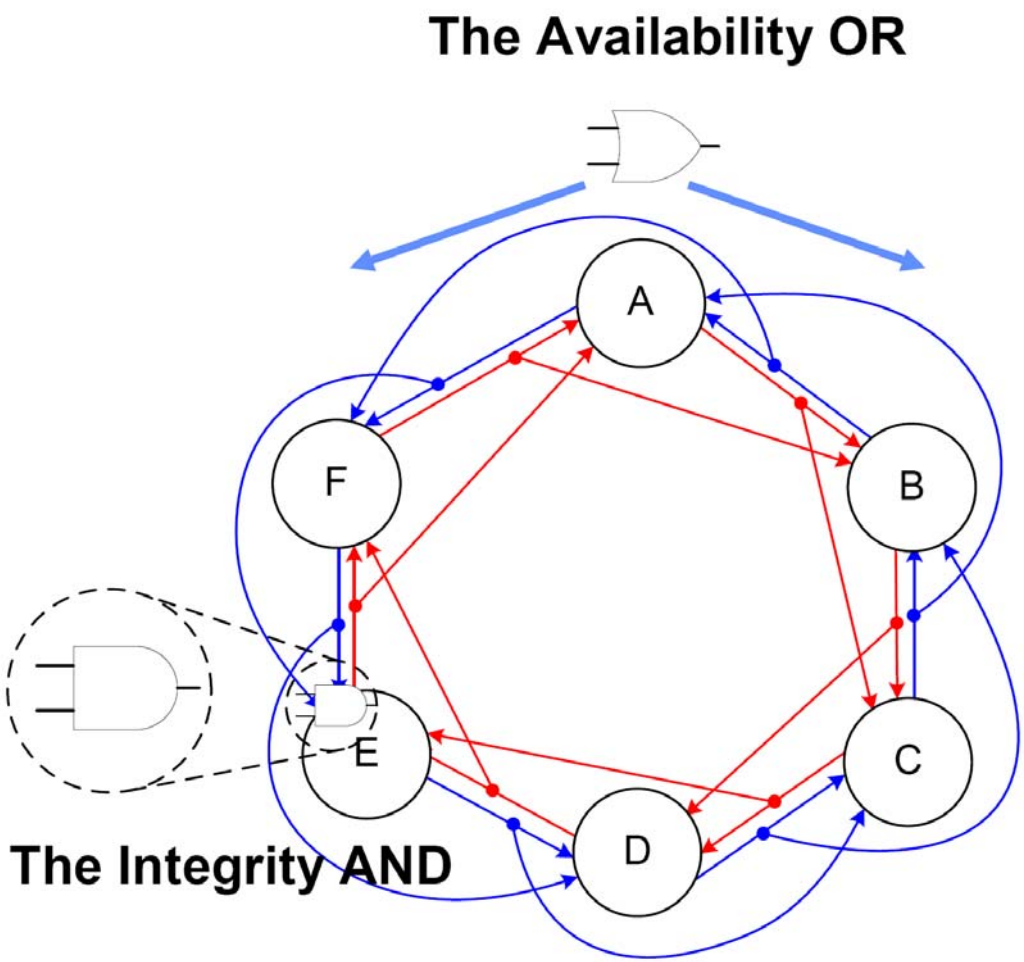
Brendan Hall, Honeywell International
Kevin Driscoll, Honeywell International
Michael Paulitsch, Honeywell International
Samar Dajani-Brown, Honeywell International

2005 International Conference on Dependable Systems and Networks (DSN'05) pp. 298-307

Braided Ring (Verschränkter Ring): Inspiriert von den Eigenschaften SafeBus

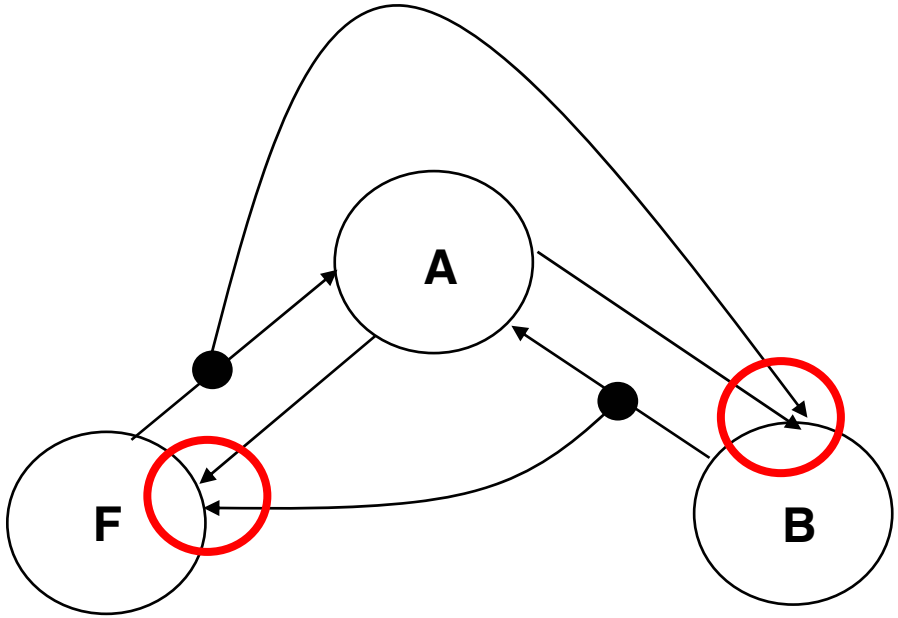
Ziele:

- **Höchste Integrität der Nachrichtenübertragung**
- **Tolerierung von Knotenausfällen und Verbindungsausfällen**
- **Schutz gegen byzantinische Fehler und Monopolisierung des Netzwerks**
- **Low cost Guardians**
- **Sicherer Start-up und Re-Integration von Knoten**
- **Integrität von Quelldaten und Unterstützung redundante Berechnungen**



Concept of the Braided Ring

Availability OR

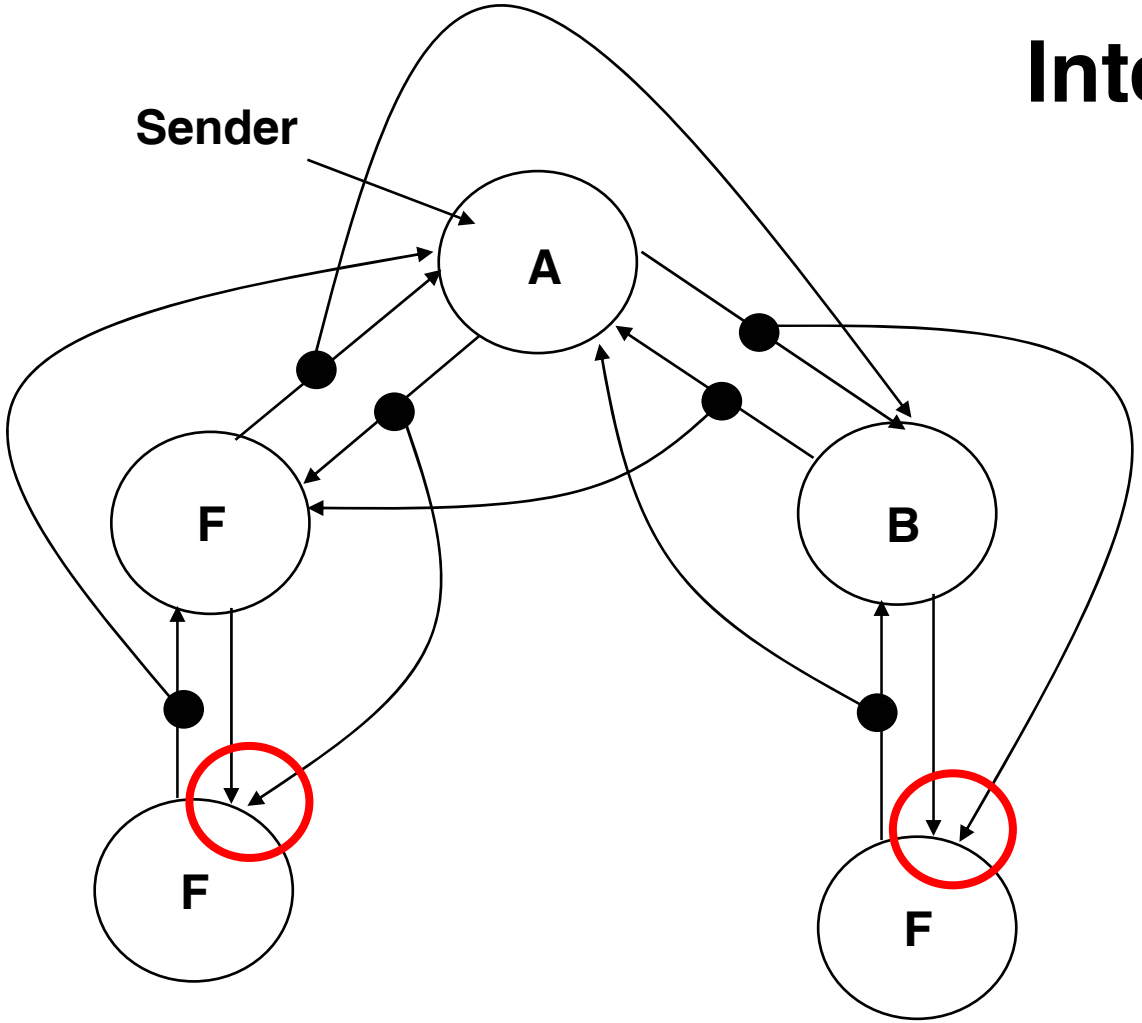


Availability OR toleriert:

- **Knotenausfälle (keine Weiterleitung)**

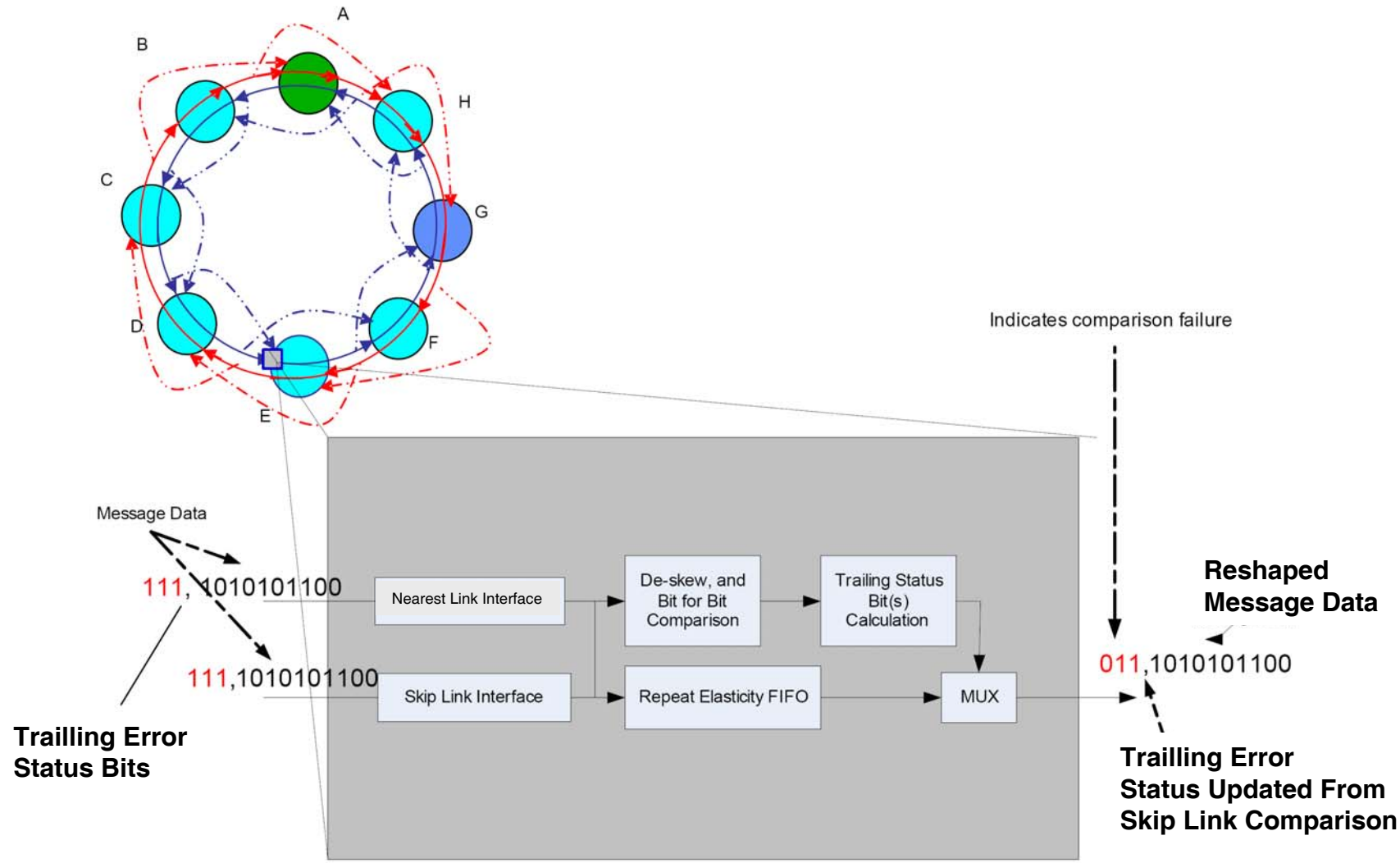
- **Verbindungsausfälle**
  - **beide Richtungen können genutzt werden**
  - **Babblin Idiot Fehler können maskiert werden**

Integrity AND



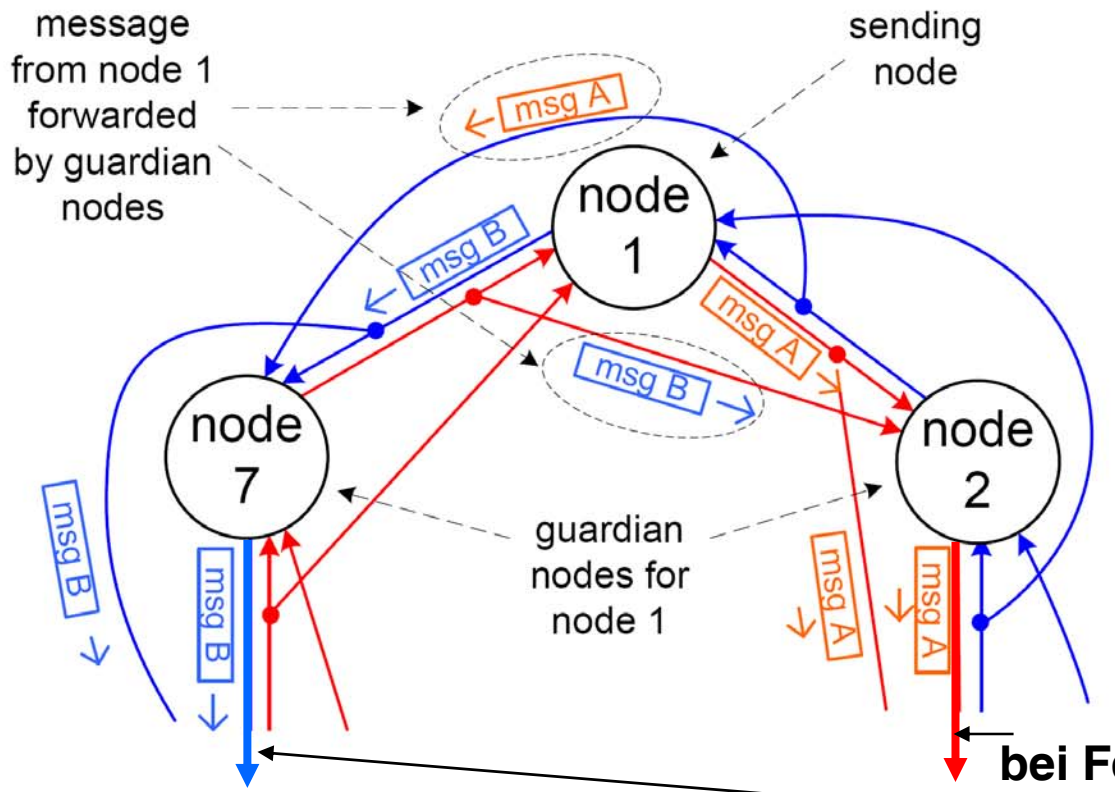
Erkennung von Nachrichtenverfälschungen

(byzantinische Fehler)



Braided Ring Propagation and Status Generation and Appending

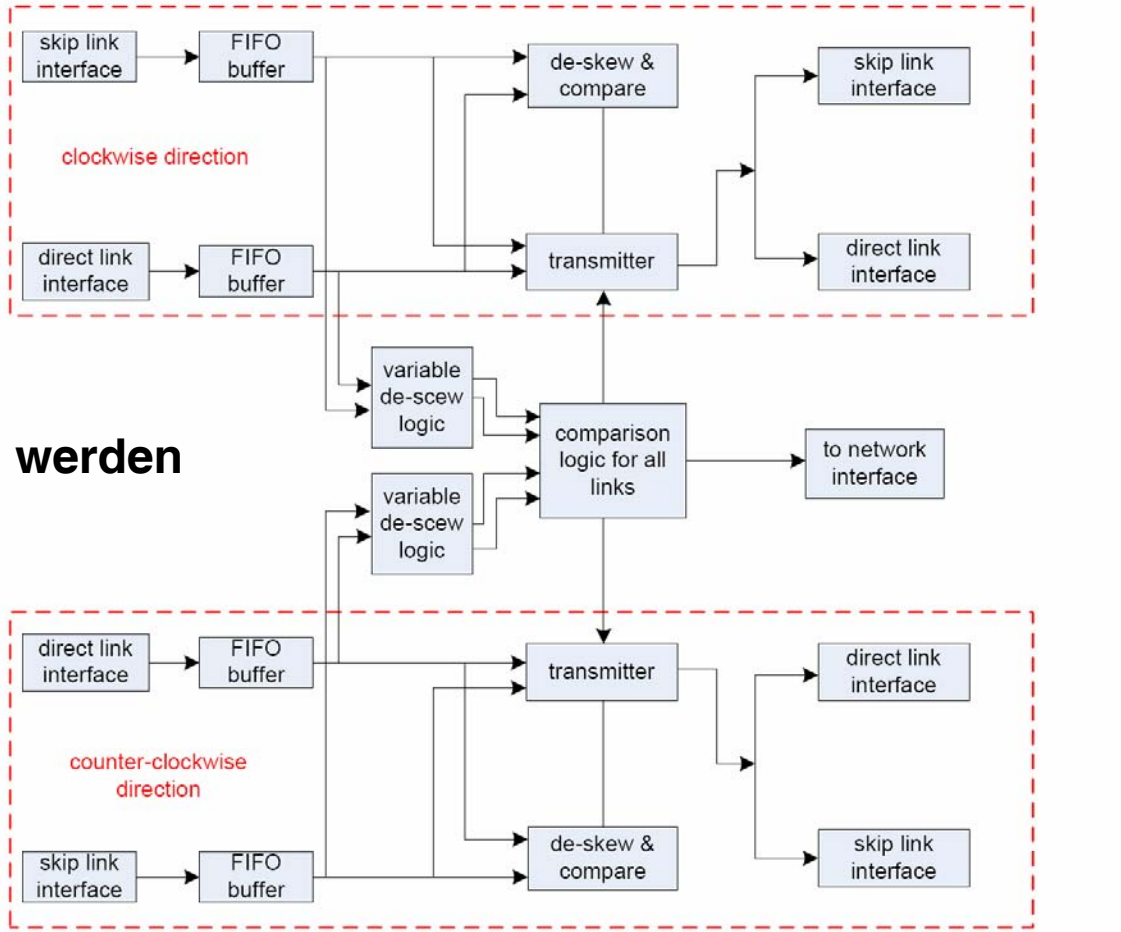
- **Bit-by-Bit Vergleich auf den eingehenden Links**
- **Alle Fehler, die von Nachbarn verursacht werden können erkannt werden**
- **Status des Vergleichs wird in den "trailing bits" transportiert**
  - **jeder Knoten fügt seinen Status der Nachricht hinzu**
    erlaubt präzise Fehlerlokalisatin
  - **"Aggreggated Error Status" :**
    ein Knoten kann den Status einer Nachricht von gültig --> ungültig ändern aber nicht umgekehrt.
  - **Alle durch einen weiterleitenden Knoten induzierten Fehler werden erkannt.**
  - **CRC wird eingesetzt, um Fehler auf den "direct Links" zu erkennen,**
  - **Zuverlässigkeit von 10-9 erfordert höchsten Schutz gegen alle möglichen "unglaublichen" Fehler wie Maskerade oder kontrollierte Datenkorruption**



Byzantine Transmission Detection

Guardians sorgen dafür, dass bei TDMA nur im dafür vorgesehenen Zeitschlitz gesendet wird.

4 Nachrichten werden verglichen !



Reconstitution Of Integrity

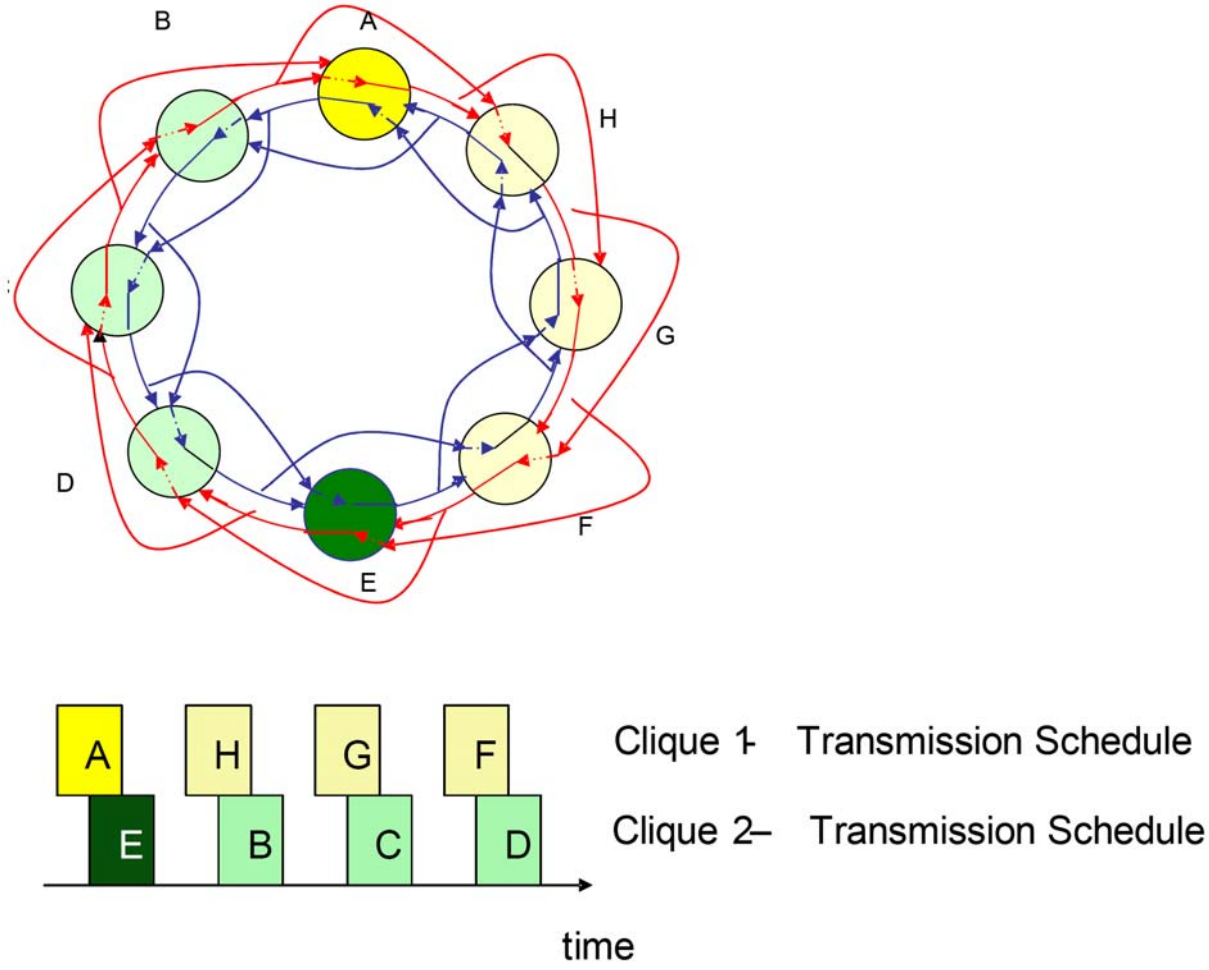
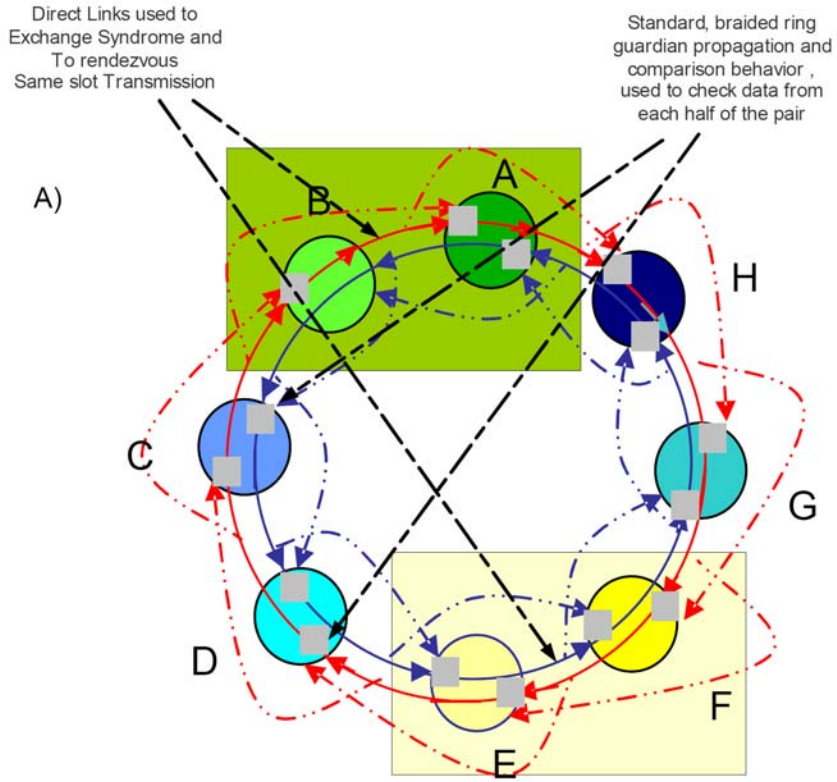
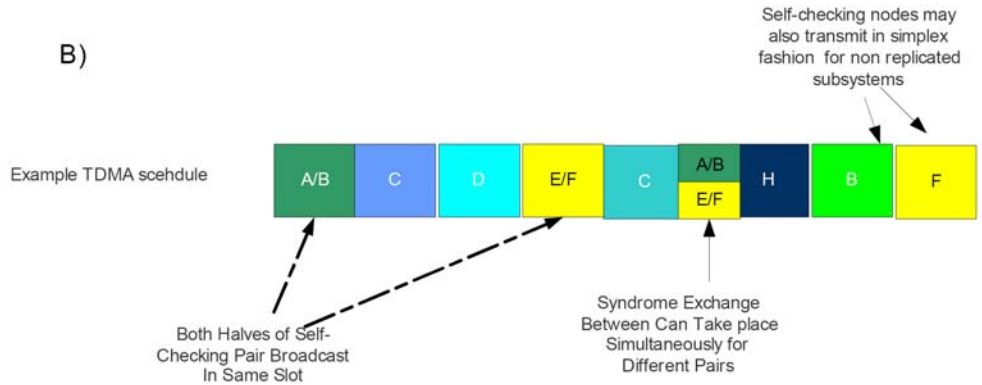
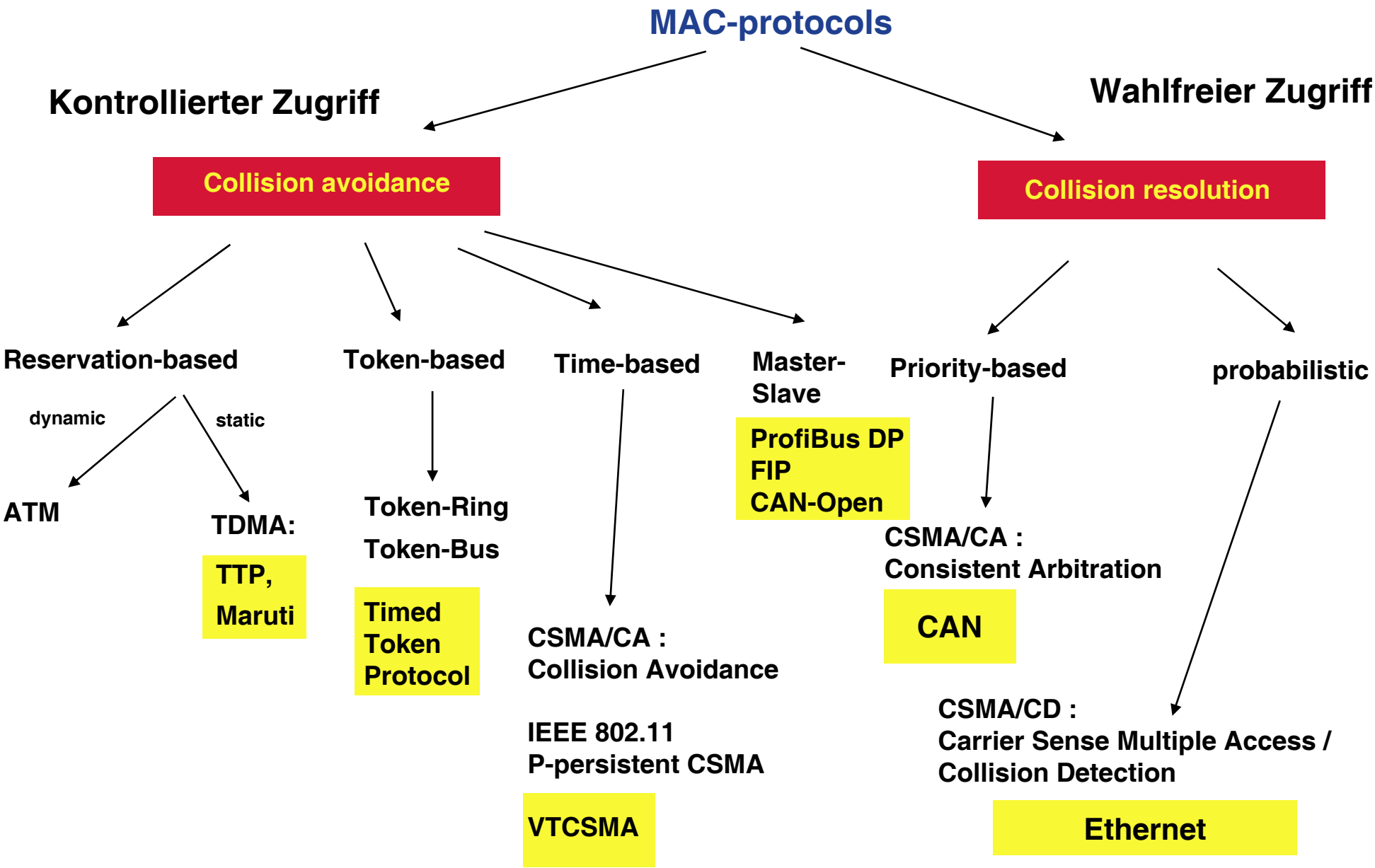


Figure 6. Pathological Clique Formation

Wechselseitige Überwachung und Replikations-Aktualisierung






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

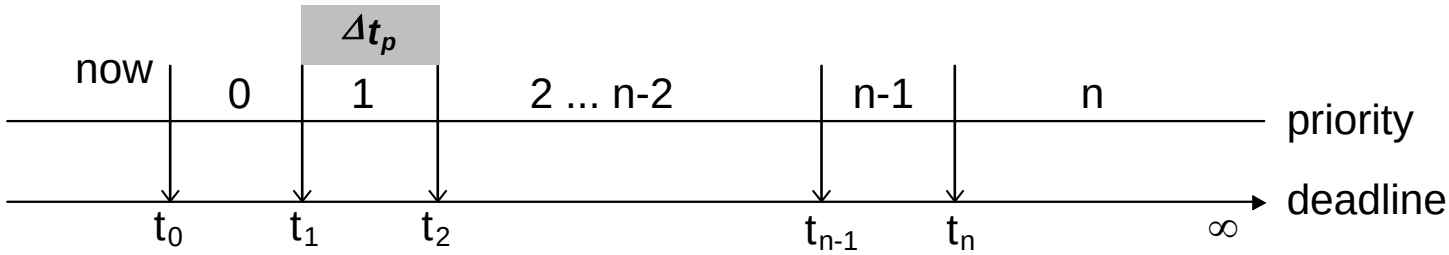
W. Zhao, J.A. Ramamritham: ***Virtual Time CSMA Protocols for Hard Real-Time Comunication***
IEEE Trans. on Softw. Engineering, SE 13(8), 1987

Annahmen:
Folgende Informationen stehen jedem Konten zur Verfügung:
- **Status des Kanals (frei oder besetzt)**
- **globale Zeit durch synchronisierte Uhren**
- **Lokale Prioritäten der sendebereiten Nachrichten**

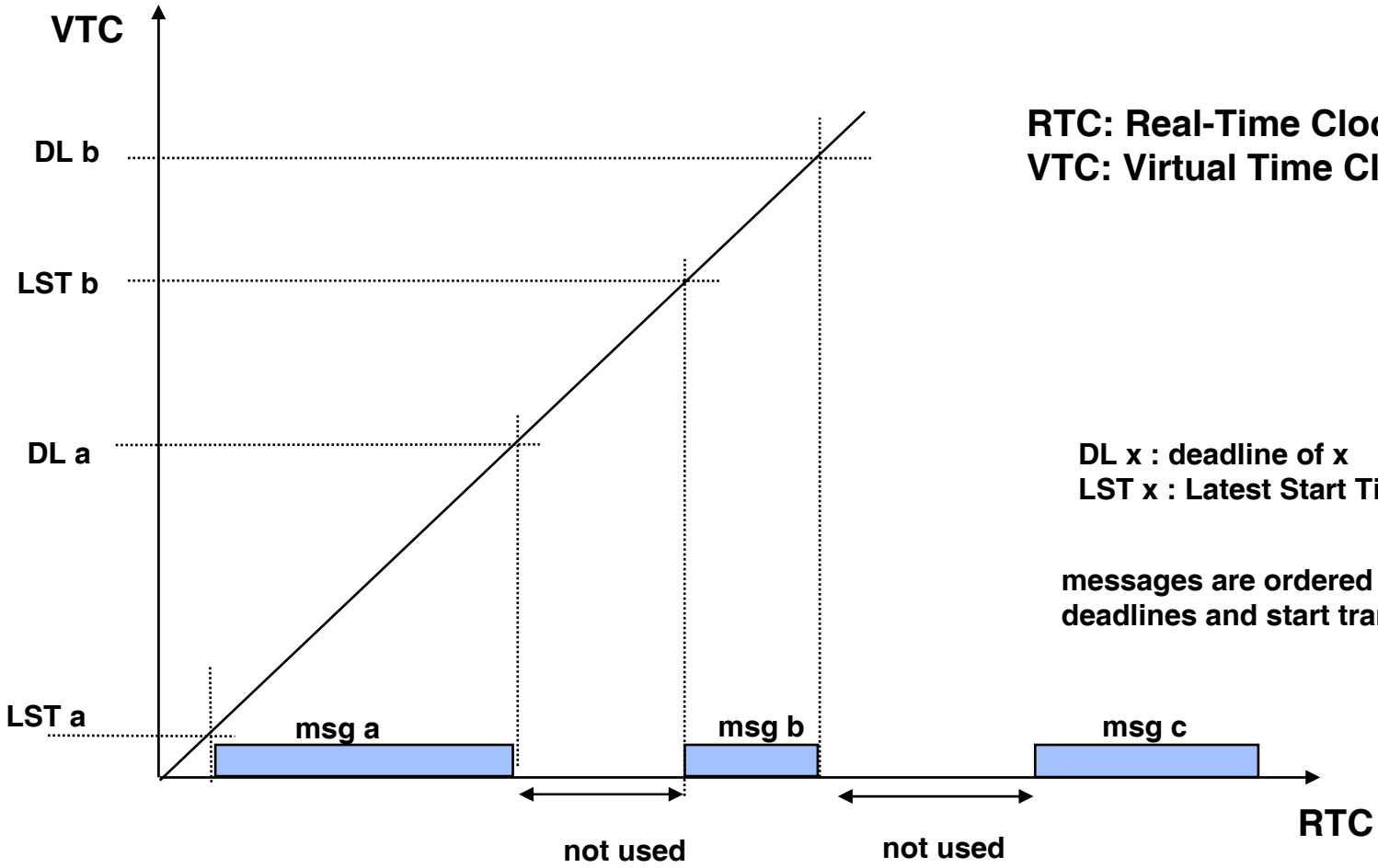
Grundidee des Verfahrens:

Berechnung der globalen Prioritätsordnung als Funktion der Zeit.

Abbildung von Deadlines auf Prioritäten



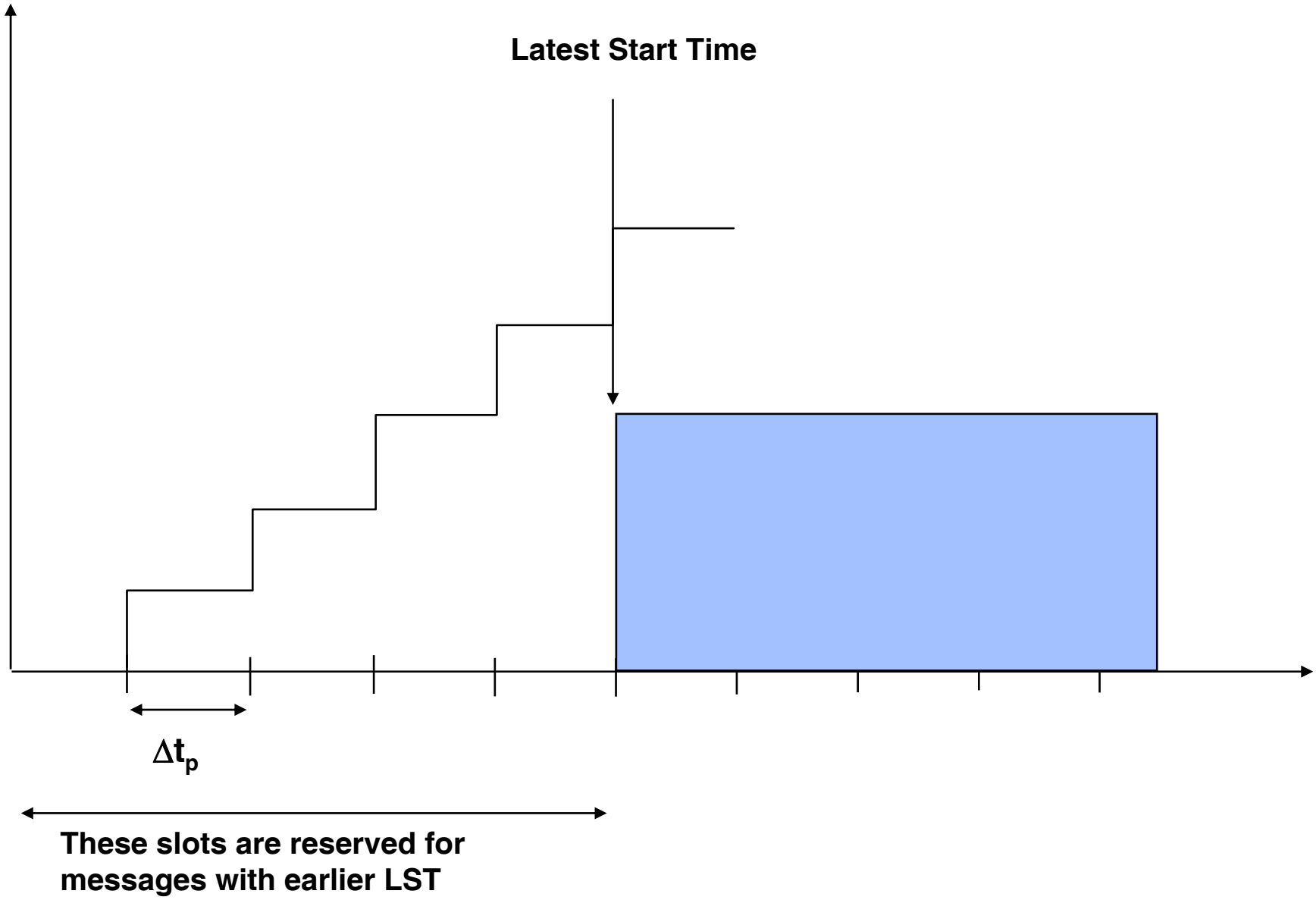
Virtual Time CSMA



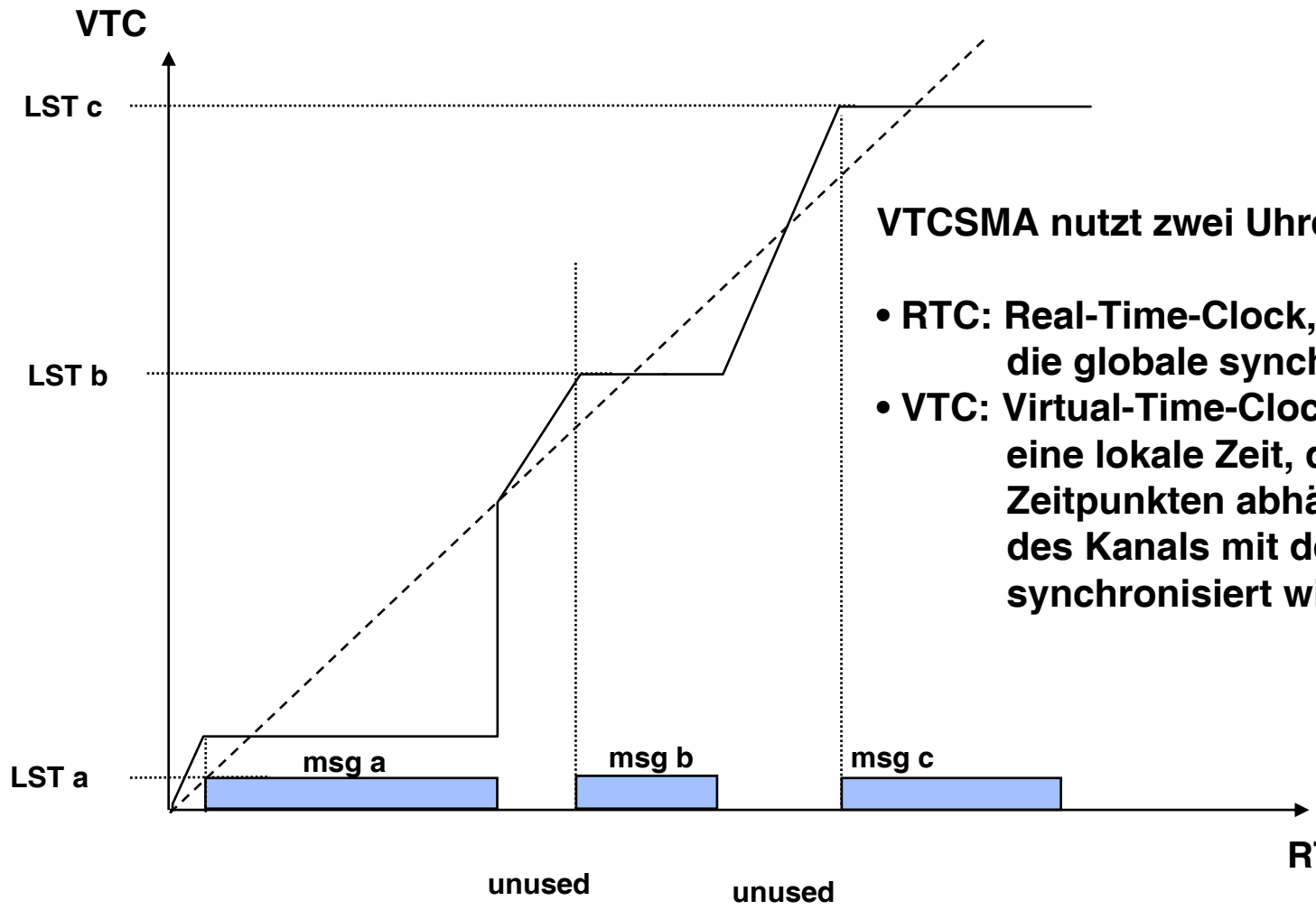
RTC: Real-Time Clock (global time)
VTC: Virtual Time Clock (local time)

DL x : deadline of x
LST x : Latest Start Time of x

messages are ordered according their deadlines and start transmission at LST


Latest Start Time
Δt_p
These slots are reserved for
messages with earlier LST

Virtual Time CSMA



VTCSMA nutzt zwei Uhren:

- **RTC: Real-Time-Clock, repräsentiert die globale synchronisierte Zeit**
- **VTC: Virtual-Time-Clock repräsentiert eine lokale Zeit, die zu bestimmten Zeitpunkten abhängig vom Status des Kanals mit der RTC synchronisiert wird.**

Ausprägungen des VTCSMA-Algortihmus

τ : Laufzeit von einem Ende des Netzwerkes zum anderen

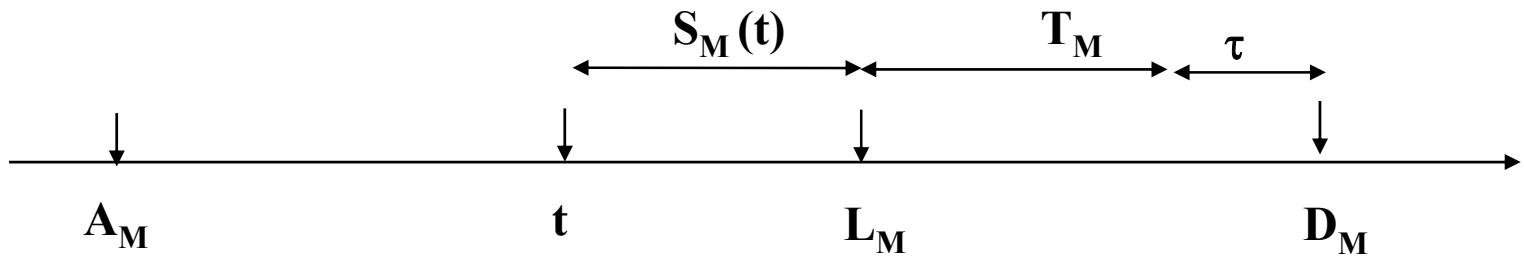
A_M : Nachricht wird in die lokale WS eingereiht (Arrival Time)

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. $L_M = D_M - T_M - \tau$

$S_M(t)$: Maximale Zeit, die eine Nachricht zum Zeitpunkt t noch verzögert werden darf ohne ihre Deadline zu verpassen.
D.h. $S_M(t) = D_M - T_M - \tau - t$



Ausprägungen des VTCSMA-Algortihmus

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

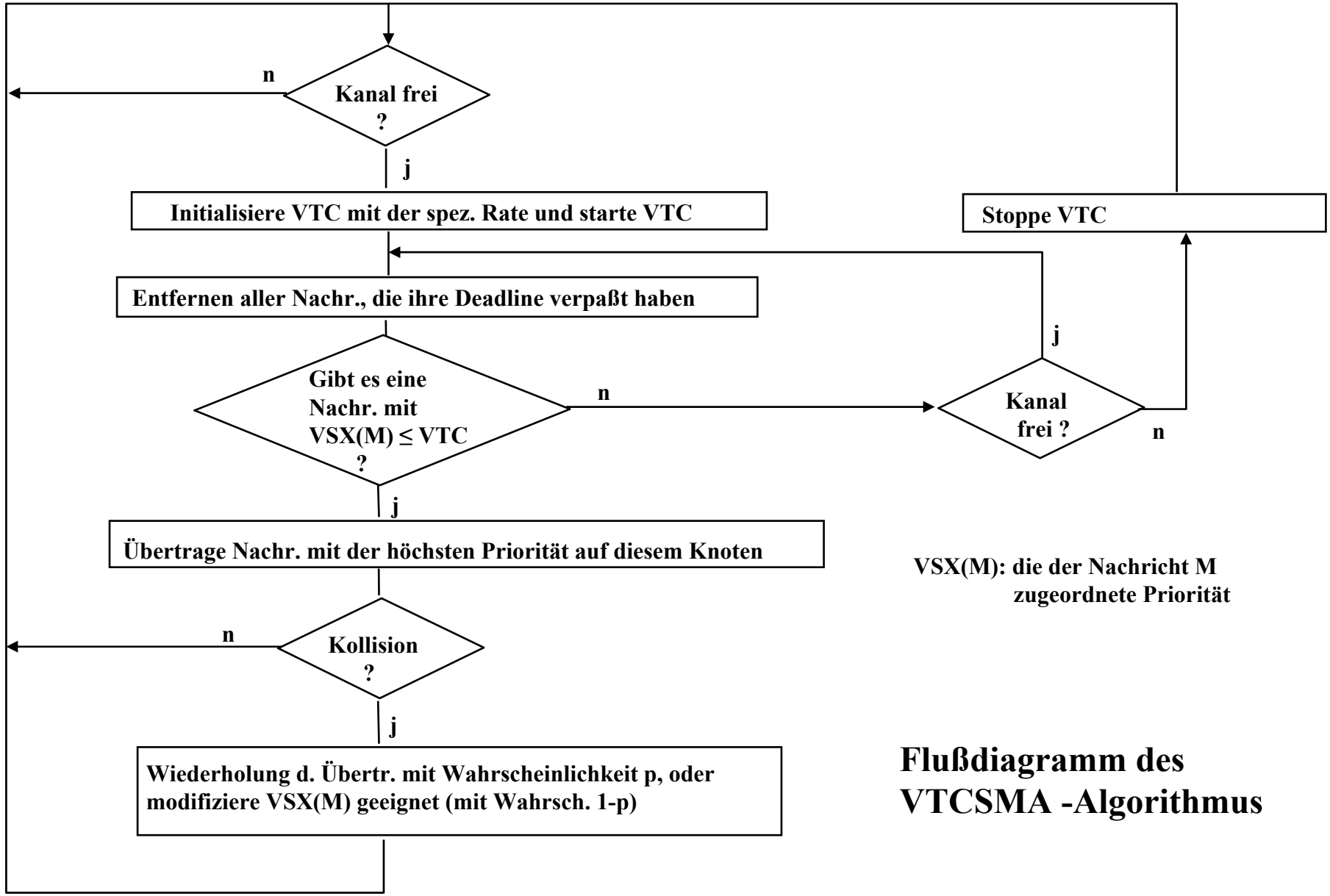
$$VSX(M) = \begin{cases} A_M : \text{für VTCSMA-A (Earliest Arrival First)} \\ T_M : \text{für VTCSMA-T (Minimum Transmission Time First)} \\ L_M : \text{für VTCSMA-L (Minimum Laxity First)} \\ D_M : \text{für VTCSMA-D (Earliest Deadline First)} \end{cases}$$

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 :

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

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

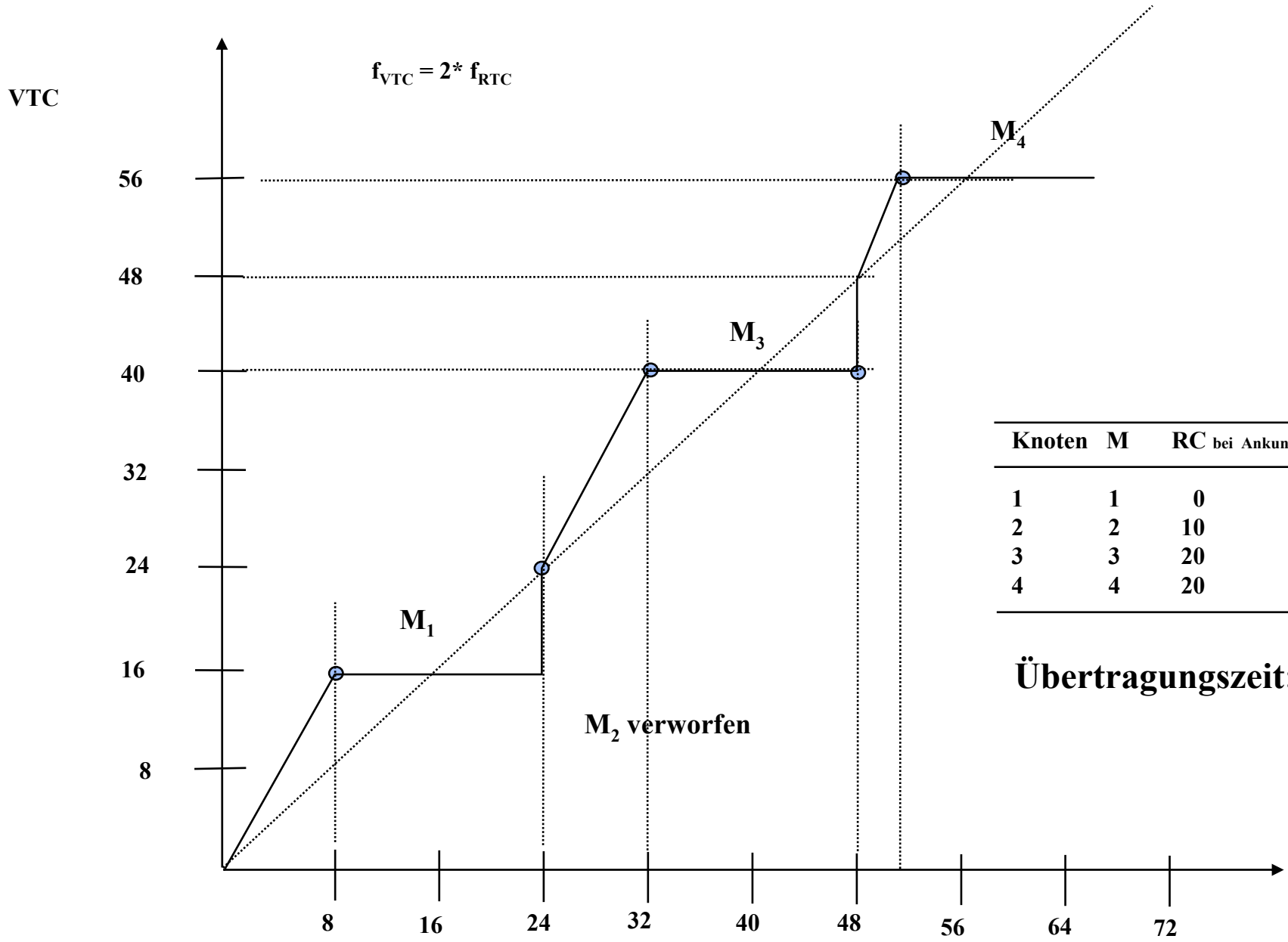
$$VC = \begin{cases} \text{keine Änderung} & \text{für VTCSMA-A} \\ 0 & \text{für VTCSMA-T} \\ \text{RTC} & \text{für VTCSMA-L} \\ \text{RTC} & \text{für VTCSMA-D} \end{cases}$$



VSX(M): die der Nachricht M zugeordnete Priorität

Flußdiagramm des VTCSMA -Algorithmus

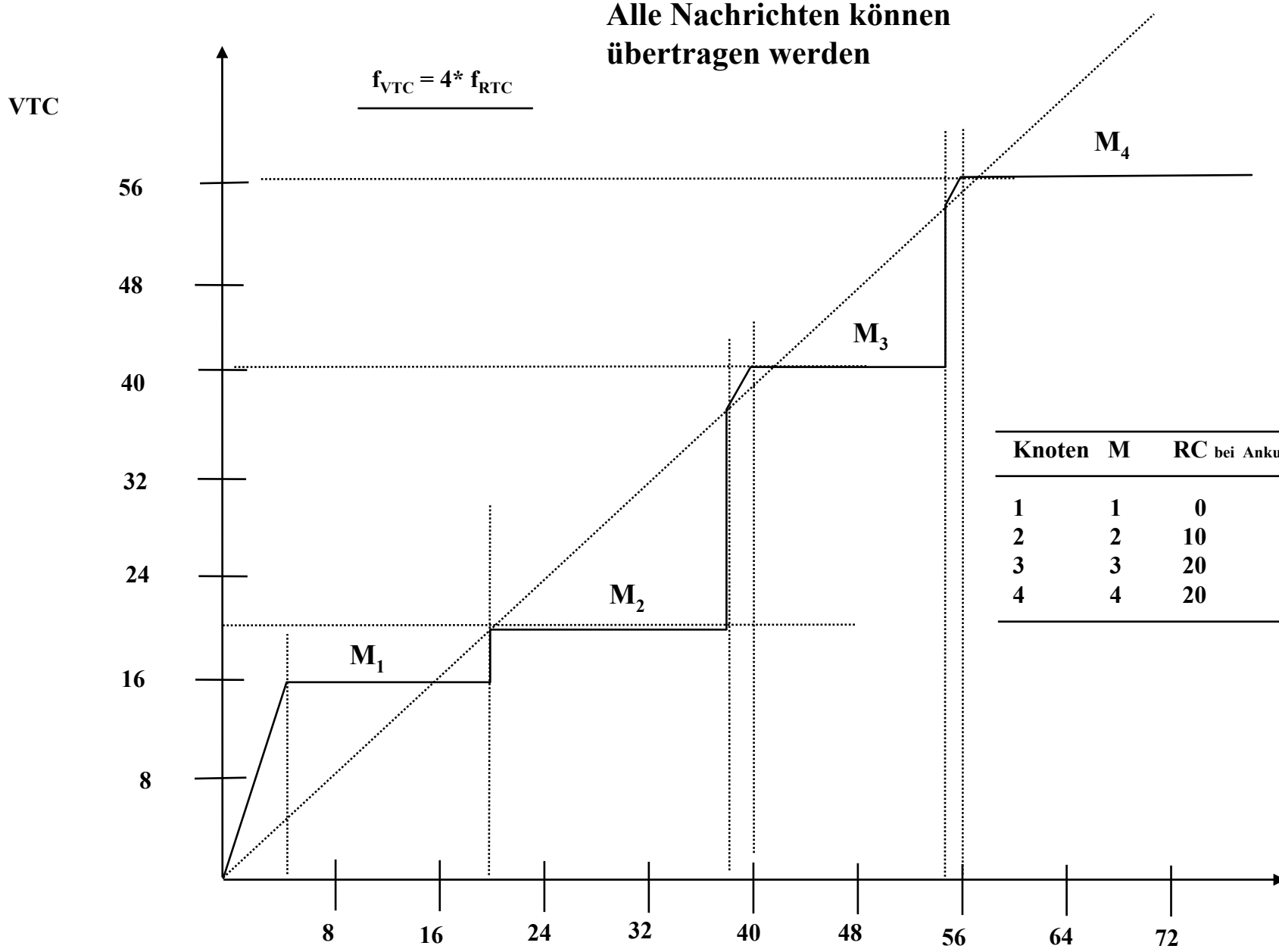
Beispiel für VTCSMA-L



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

Übertragungszeit: 16

Beispiel für VTCSMA-L
Alle Nachrichten können übertragen werden



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 Uhrensysnchronisation wird nicht berücksichtigt.**

Probleme:

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

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...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, das 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

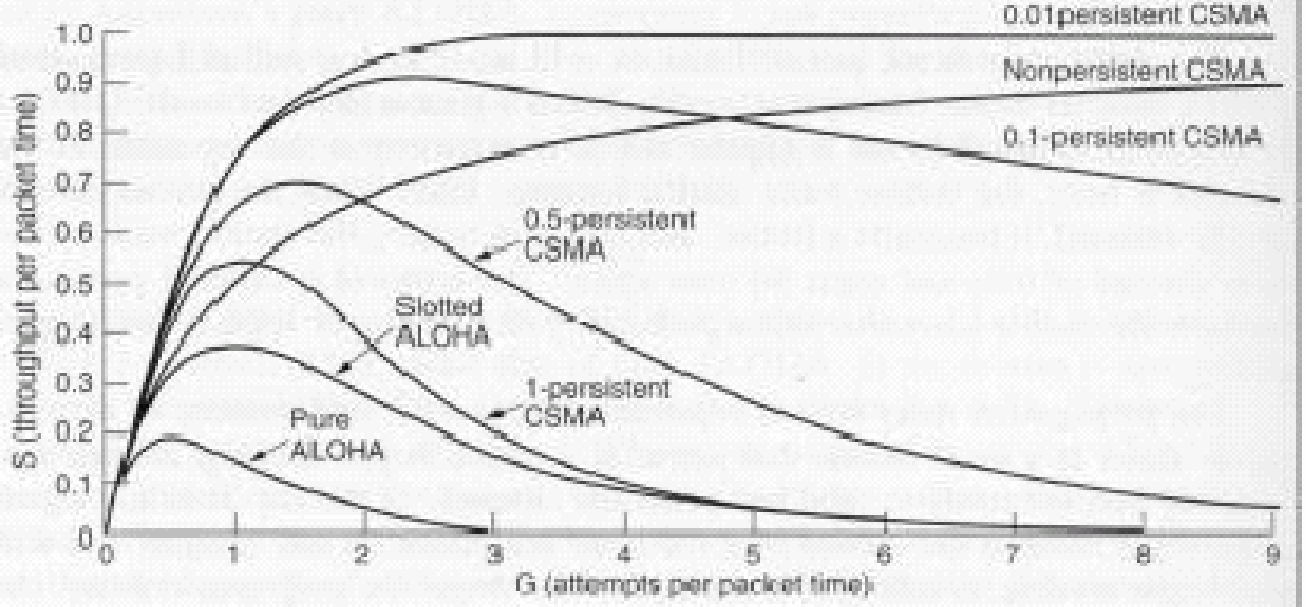
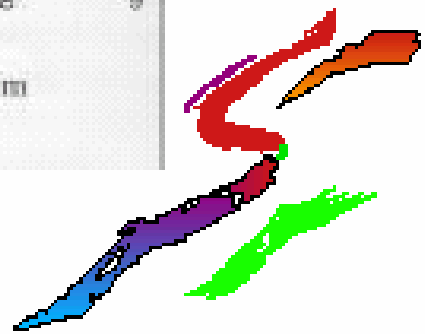


Fig. 4-4. Comparison of the channel utilization versus load for various random access protocols.



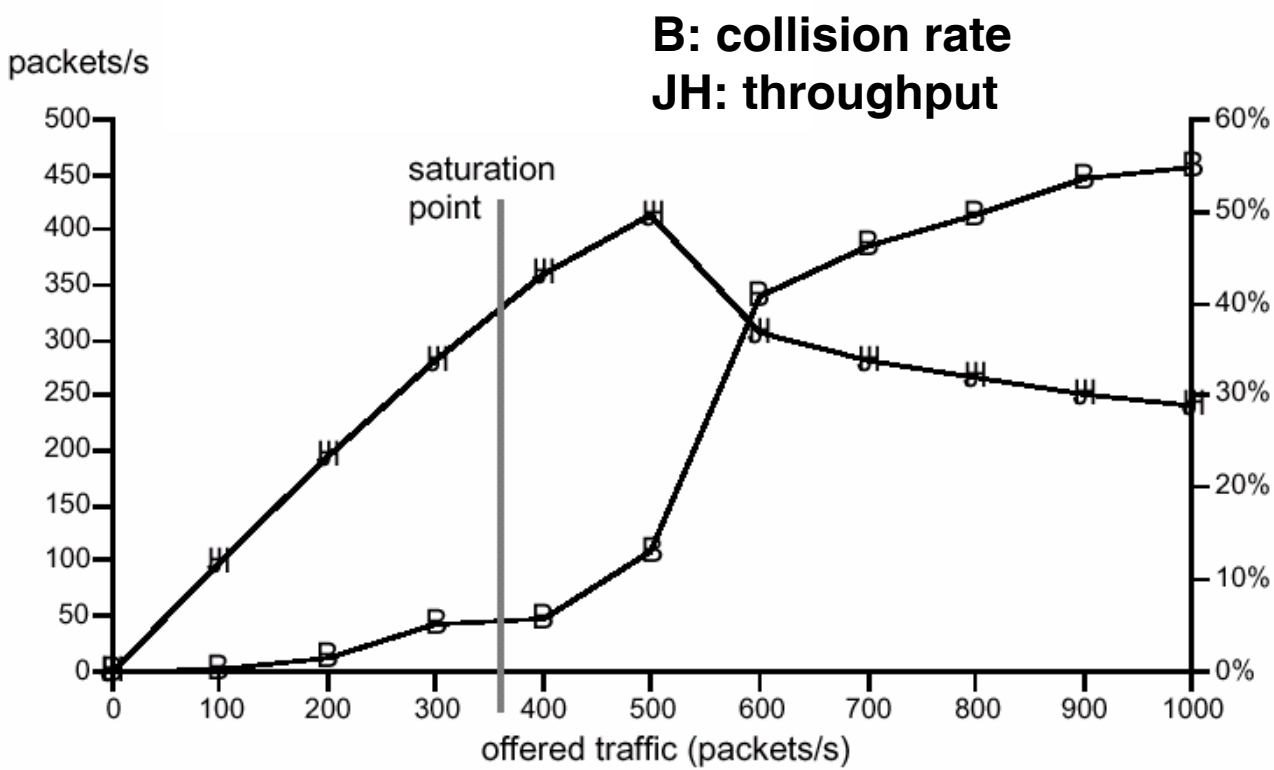


Figure 1: Effective network throughput versus offered traffic on a LONWORKS network using unacknowledged services. Note that in this mode, LonTalk behaves like conventional, p-persistent CSMA.

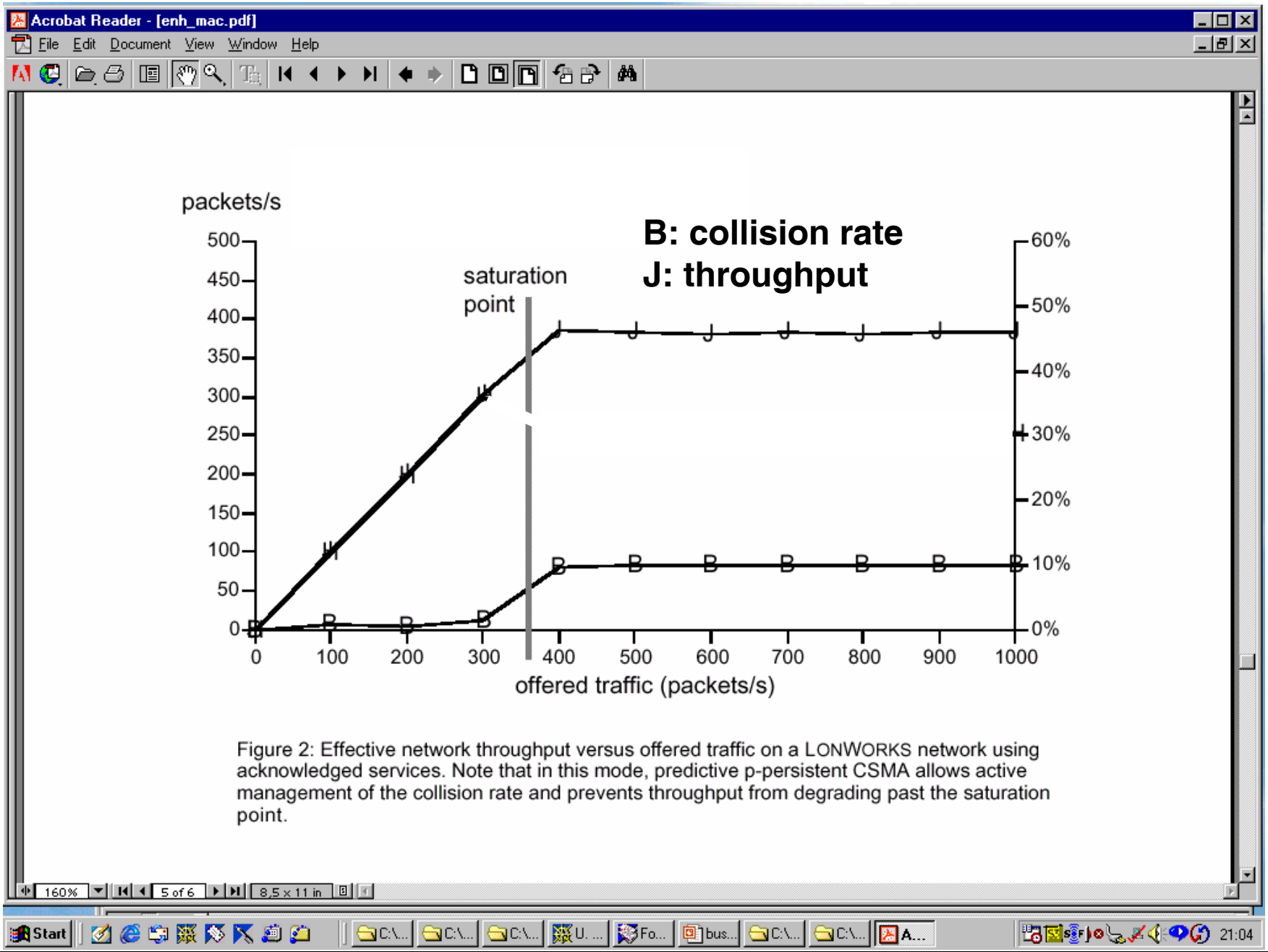


Figure 2: Effective network throughput versus offered traffic on a LONWORKS network using acknowledged services. Note that in this mode, predictive p-persistent CSMA allows active management of the collision rate and prevents throughput from degrading past the saturation point.

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:

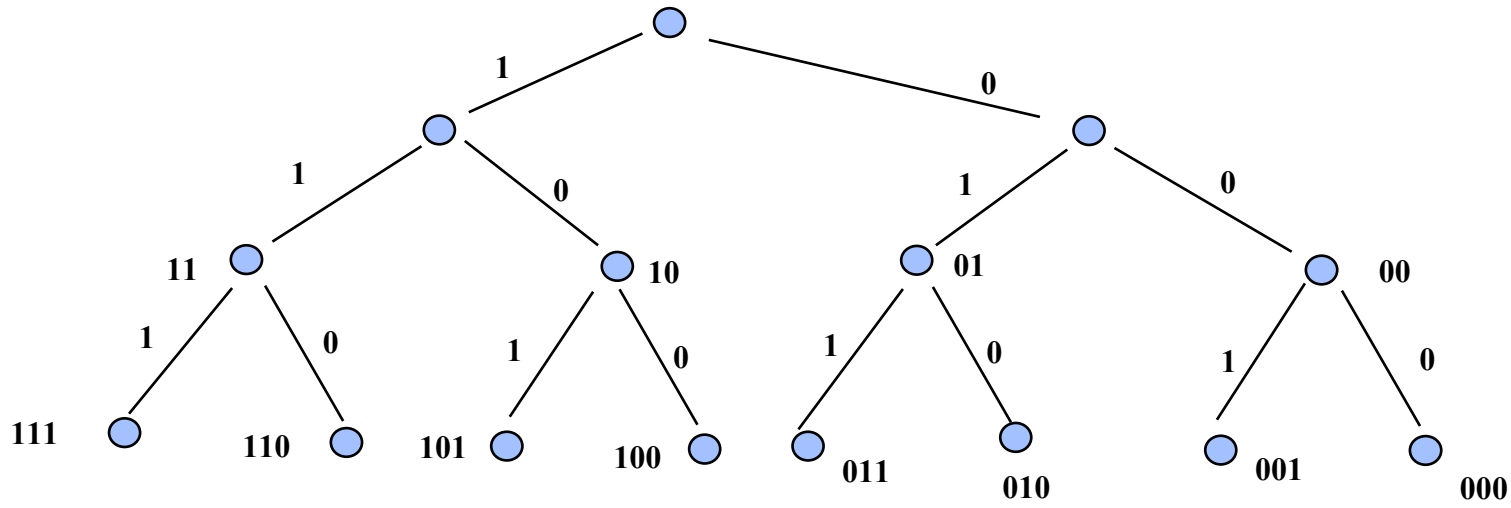
- **Status des Kanals (frei oder besetzt)**
- **globale Zeit durch synchronisierte Uhren**
- **Die LTTT der sendebereiten Nachrichten (LTTT: Latest Time To Transmit)**

Grundidee des Algorithmus:

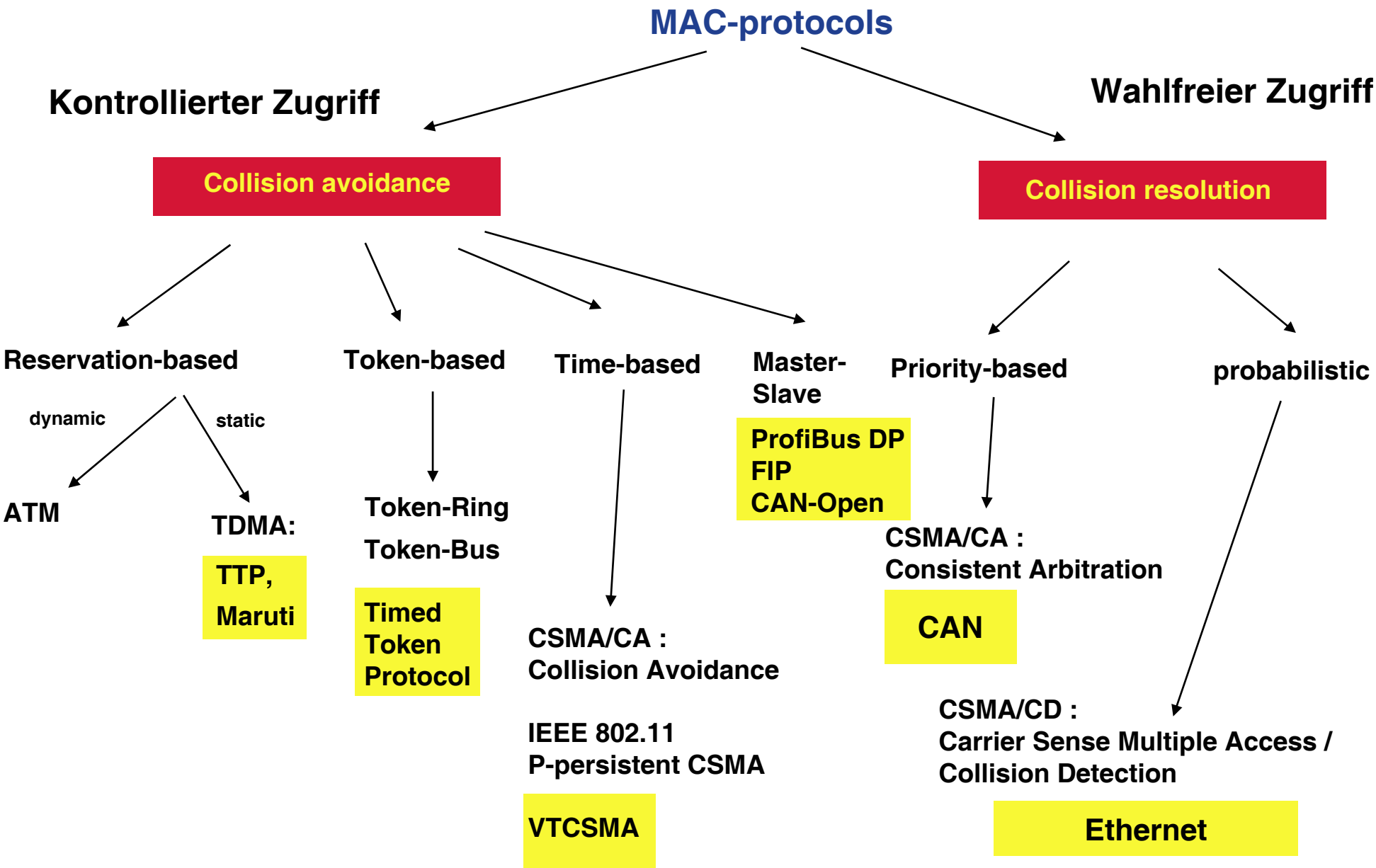
- **Teile die Zeit in Fenster fester Länge**
- **Falls der Bus frei ist sende die Nachricht mit der geringsten LTTT zu Beginn eines Fensters**
- **Falls eine Kollision auftritt, halbiere das Fenster. Falls die LTTT in das neue Fenster fällt, sende die Nachricht.**
- **Falls eine Nachricht nicht zu ihrer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


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

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)

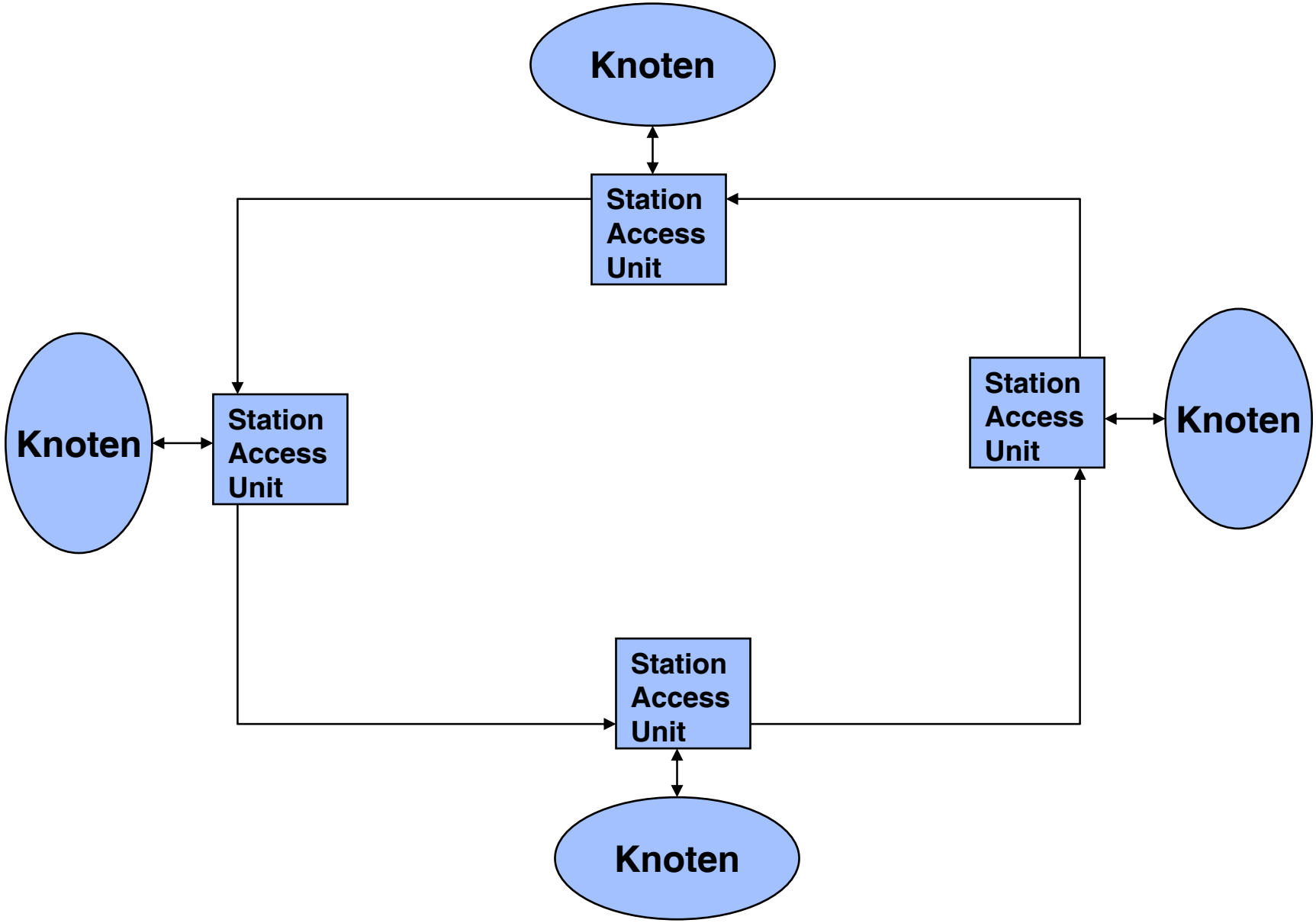
Delegated Token:

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

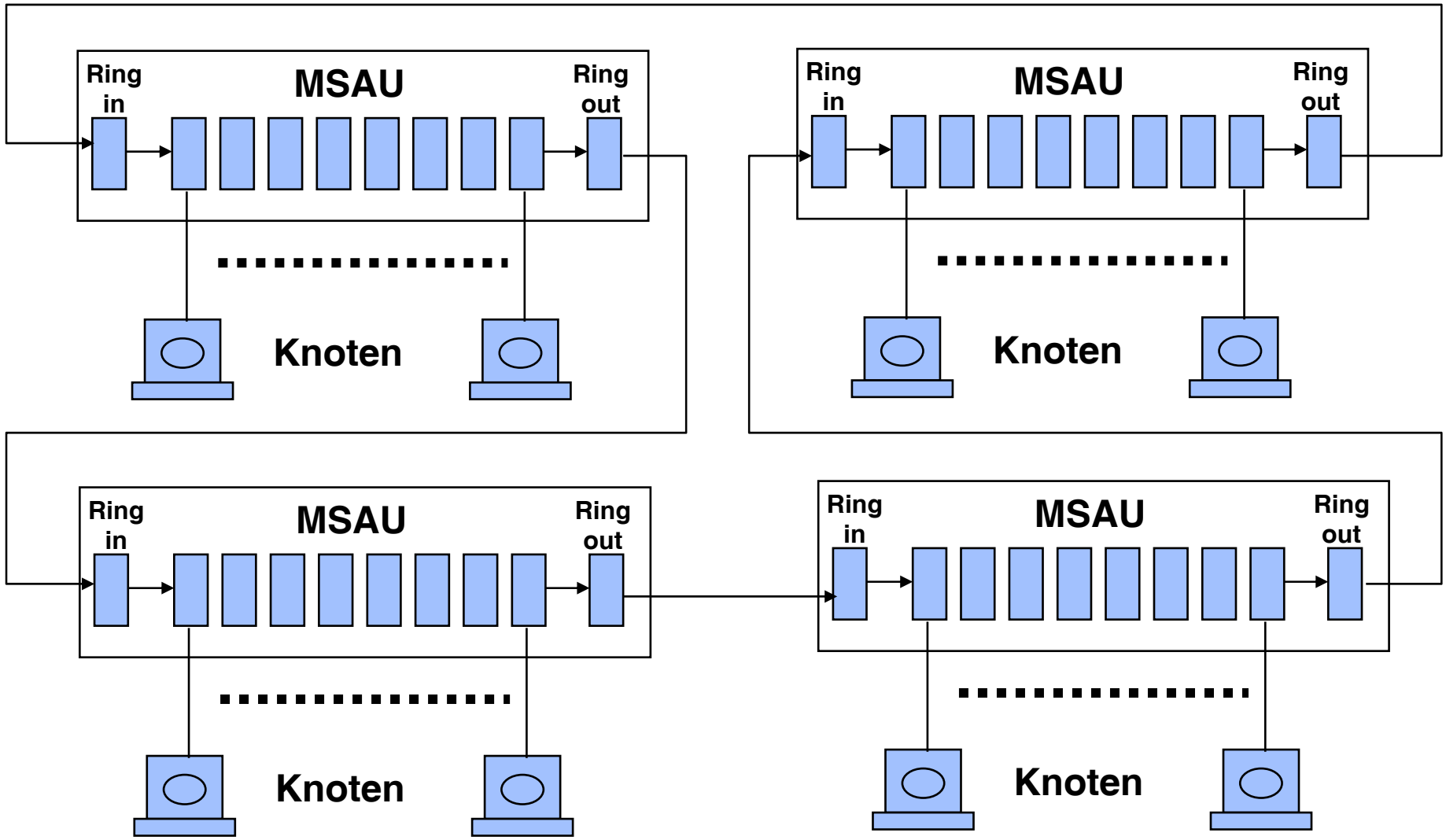
Nachrichtenorientiert: Busarbiter fordert über eine spezielle Nachricht, dem Token, das die ID der gewünschten Nachricht enthält, den zuständigen Teilnehmer zum Senden der Nachricht auf. Zentral gesteuerte Nachrichtenverteilung.

Beisp.: Verbindungsorientiert: Profibus (Token Passing (logischer Ring), Master und Slaves)
Nachrichtenorientiert: FIP (Factory Instrumentation Protocol)

Token Protokolle

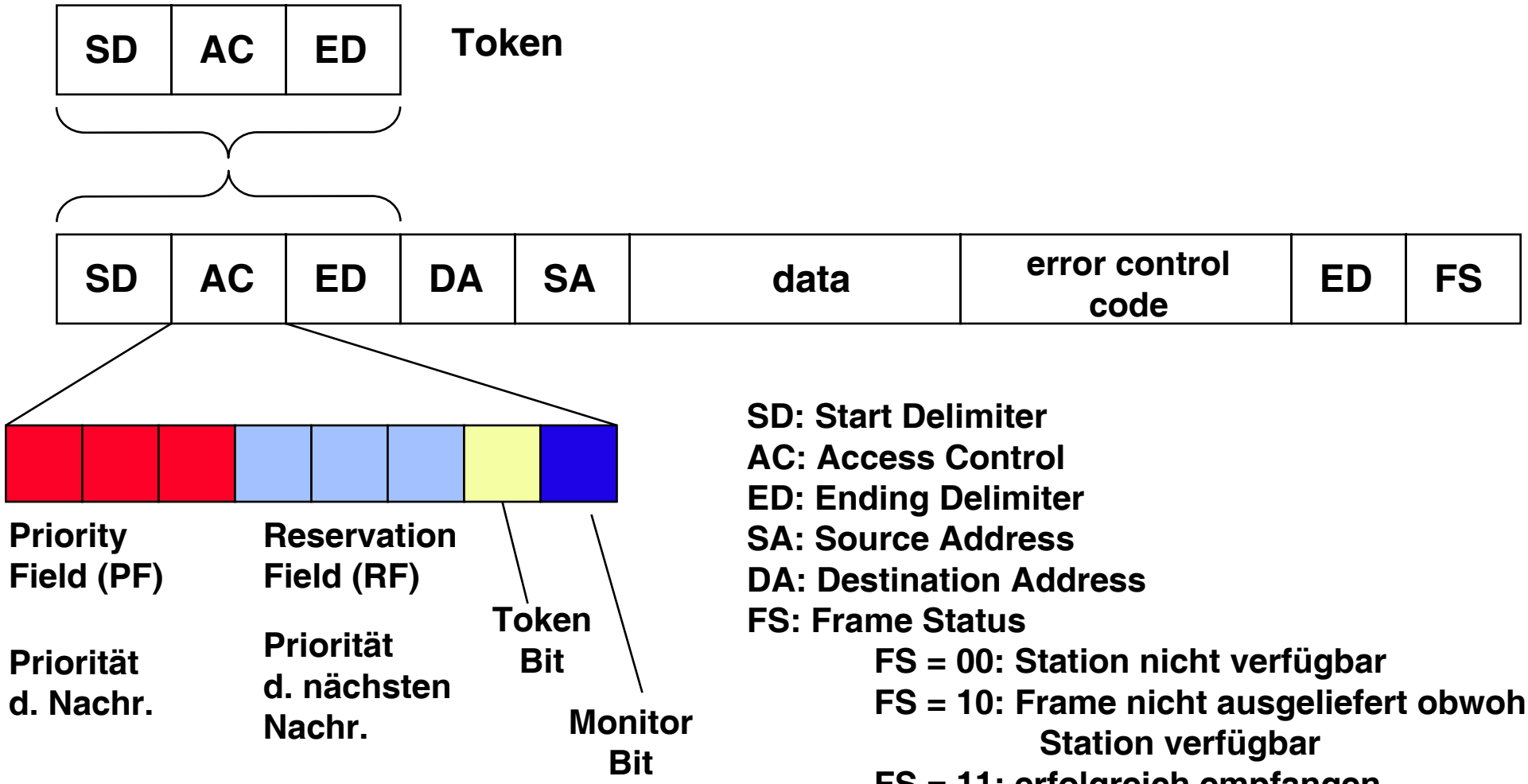


Token Netzwerk Topologien



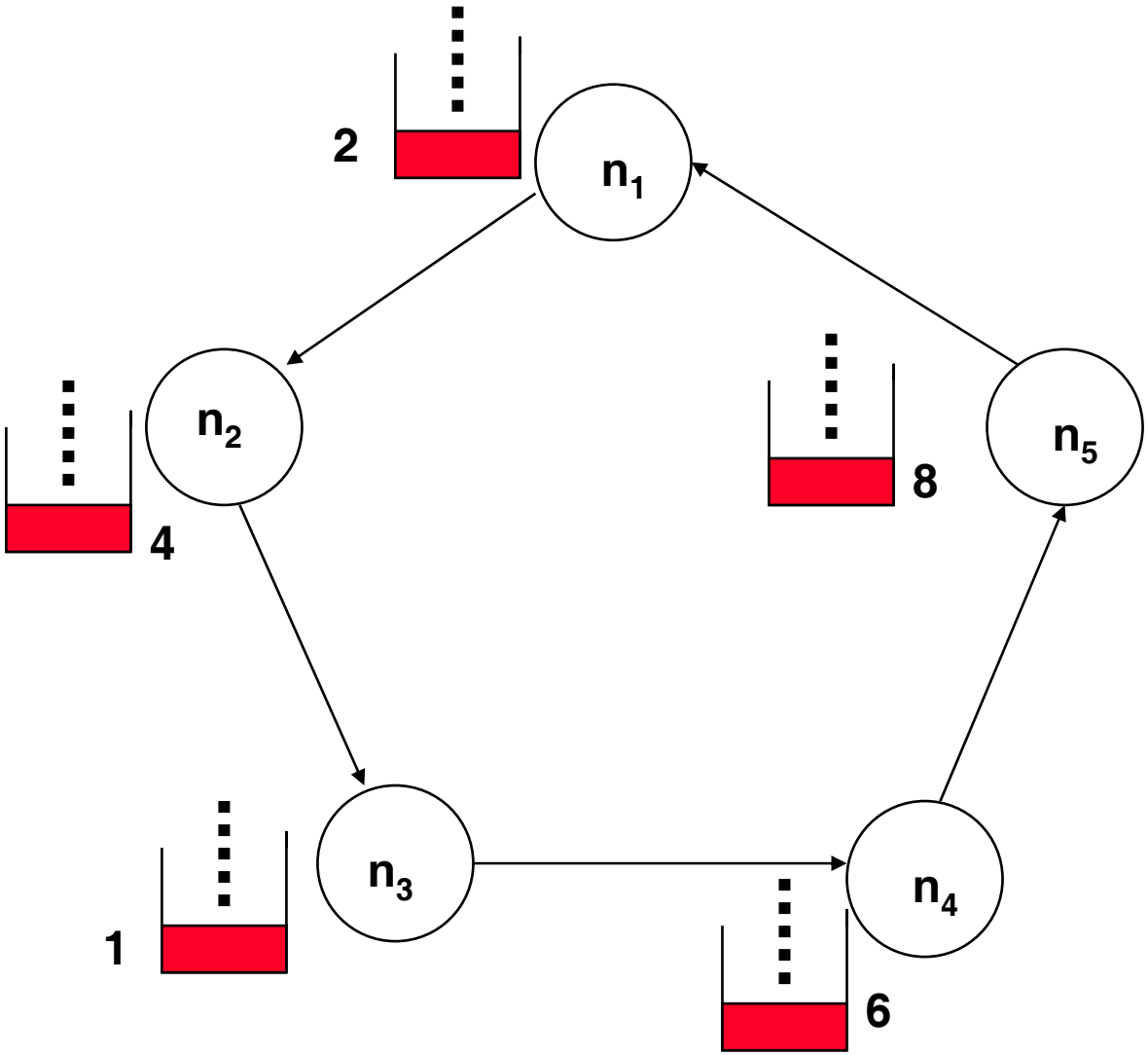
MSAU: Multi Station Access Unit

802.5 Token Ring Frame Format



SD: Start Delimiter
AC: Access Control
ED: Ending Delimiter
SA: Source Address
DA: Destination Address
FS: Frame Status
FS = 00: Station nicht verfügbar
FS = 10: Frame nicht ausgeliefert obwohl Station verfügbar
FS = 11: erfolgreich empfangen

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.

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

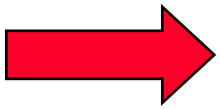
- **Nachrichten können nicht unterbrochen werden**
- **Wegen der Laufzeit im Ring kann die Priorität im RF-Feld veraltet sein**
- **Zusätzlich zur Nutzlast der Nachricht muss die Kontrollinformation übertragen werden**

Das “Timed Token Protokoll”

Timed Token Protokoll

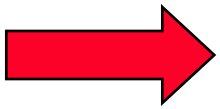
Eigenschaften:

- 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?**

Taget 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**

- **Zeit, um die Nachrichten zu übertragen**

} **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=TTRT nicht garantiert werden. Es kann aber gezeigt werden , dass durch das Timed Token Protokoll die obere Schranke: T= 2 TTRT garantiert wird.

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

$TTRT = T/2$ 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

$$\sum f_i \, t_p <= TTRT \quad \sum f_i <= TTRT$$

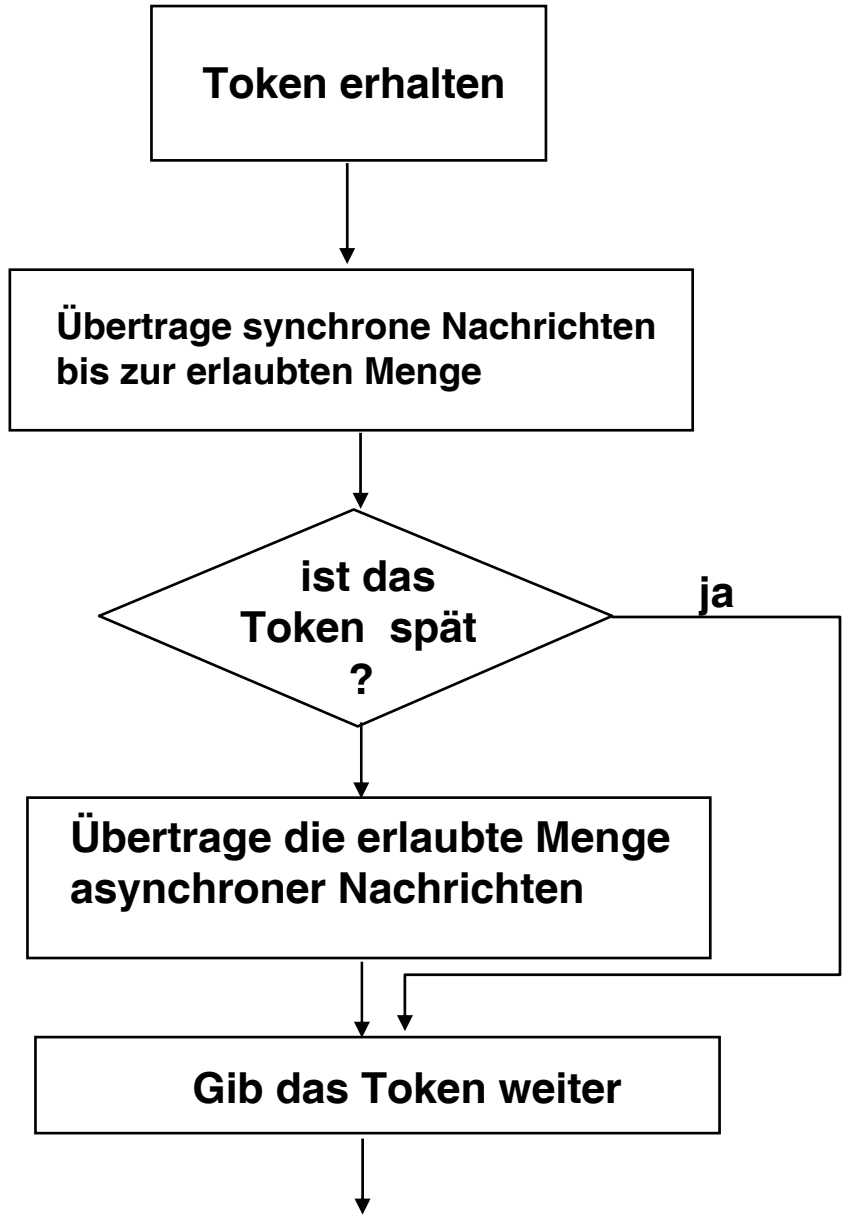
beschränkt ($\sum f_i <= 1$!).

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

Def.:
Das Token ist:

***früh*, wenn: Zyklus <= 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:

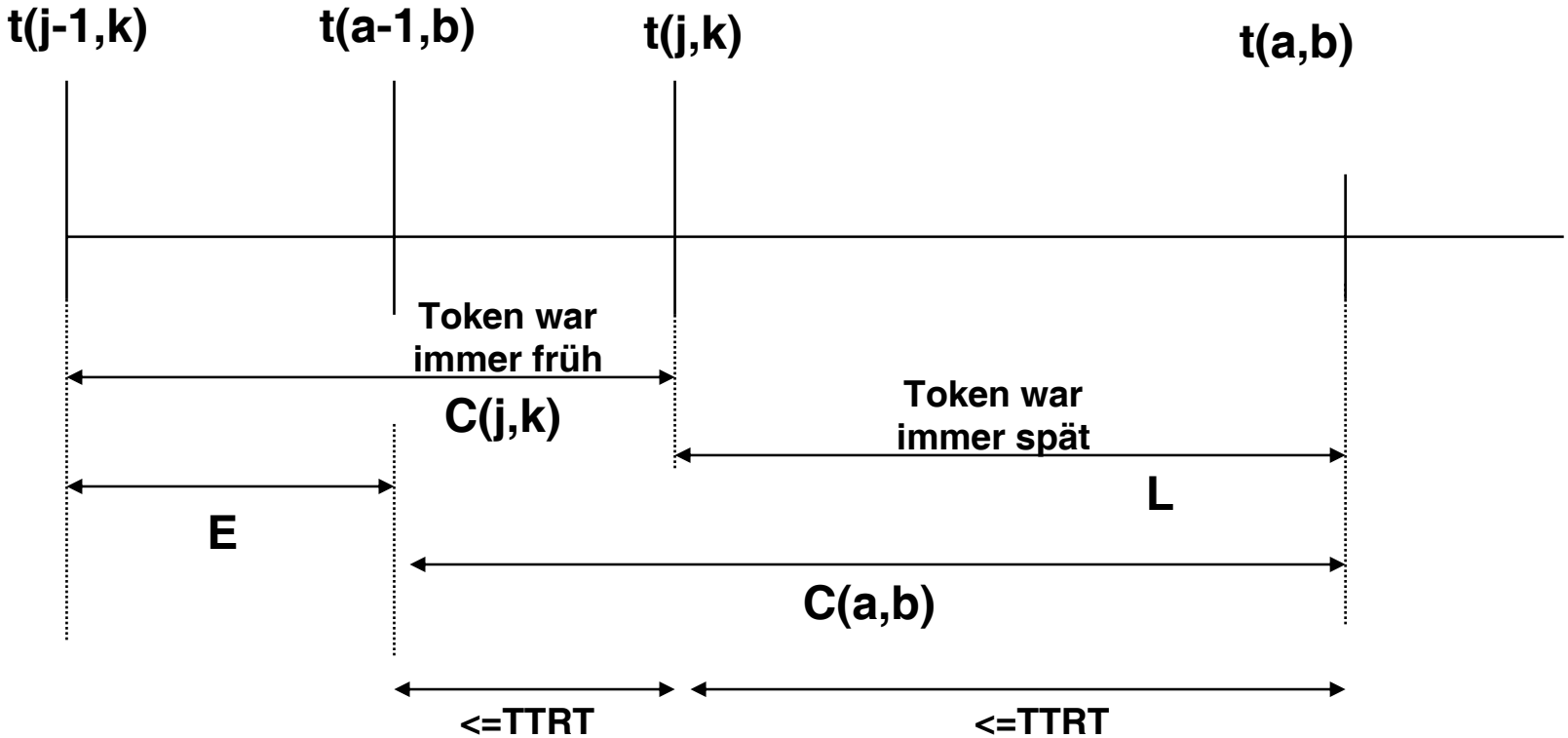
1. Fall: Das Token ist immer früh. Dann ist nach Def. von „früh“ nicht mehr als TTRT < 2TTRT Zeit vergangen.

2. Fall: Das Token ist immer spät. Dann dürfen nur synchrone Nachrichten gesendet werden. Da die Summe der Quota synchroner Nachrichten mit

$\sum f_i \, t_p <= TTRT \;\; \sum f_i <= TTRT$ **beschränkt ist** $(\sum f_i <= 1)$**, gilt die Annahme.**

Analyse des Timed Token Protokoll

3. Fall:



$C(j,k) <= TTRT$

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

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

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

$<= TTRT + TTRT = 2\ TTRT$

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

C(j,k) : j-te Token-Zykluszeit für Knoten k; Σ x,y= j,k....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•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):

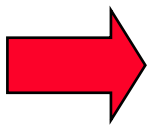
in einem bekannten Zeitintervall treten Fehler in höchstens k Telegrammen auf.

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 Sendeanforderung an gerechnet .



Eigenschaften eines Abstrakten Netzwerks