

# **Entwicklung und Implementierung eines Übungsapplets zur Vertiefung des Token Ring Standards 802.5**

Studienarbeit

vorgelegt am

Lehrstuhl Praktische Informatik IV  
Prof. Dr. W. Effelsberg  
Fakultät für Mathematik und Informatik  
Universität Mannheim

10. Februar 2004

von

**Lidia Konurov**  
aus Furmanowa

Betreuer: Dipl.-Wirtsch.-Inf. Nicolai Scheele

# Inhaltsverzeichnis

Abkürzungen.....	II
1 Einleitung.....	1
2 Token Ring .....	2
2.1 Netzwerk–Technologien und LAN–Topologien.....	2
2.2 Token Ring: Entstehungsgeschichte.....	5
2.3 Grundlagen .....	6
2.3.1 Physikalische Struktur eines Token Ring Netzes.....	6
2.3.2 Token Passing – Steuerungsverfahren .....	8
2.3.3 Token und Rahmenformate.....	10
2.3.4 Adressierung im Token Ring .....	13
2.4 Monitoring in Token Ring.....	14
2.4.1 Funktionen des aktiven und des Stand-by-Monitors.....	15
2.4.2 Fehlerbehandlung .....	15
2.4.3 Ring Purge Process .....	16
2.4.4 Claim Token Process.....	17
2.4.5 Beacon Process.....	18
2.5 Priorisierungsmechanismus .....	19
3 Token Ring Trainer.....	26
3.1 E-Learning-Grundlagen des Übungsapplets TokenRingTrainer.....	26
3.2 Aufgabentypen .....	28
3.2.1 Theorieaufgaben.....	29
3.2.2 Rechenaufgaben .....	30
3.3 GUI.....	31
3.4 Implementation.....	32
4 Zusammenfassung .....	35
Literaturverzeichnis .....	III

## Abkürzungen

ANSI	American National Standards Institute
CATS	Communication and Tutoring System
CT	Claim Token MAC Frame
GAN	Global Area Network
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standardization Organization
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MAN	Metropolitan Area Network
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
WAN	Wide Area Network

# 1 Einleitung

Ziel dieser Studienarbeit ist, zum einen theoretische Grundlagen über die in einem Token Ring Netzwerk eingesetzten Protokolle zu erarbeiten und zum anderen ein in Java implementiertes Übungstool zum Thema Token Ring zu realisieren, mit dessen Hilfe es Studierenden des Fachs Rechnernetze oder auch allgemein Interessierten ermöglicht werden kann, sich einen tieferen Einblick in die Konzepte des Token Rings zu verschaffen. Der Anwender kann sich Schritt für Schritt anhand konkreter Beispiele oder Szenarien in die einzelnen Konzepte und ihre Zusammenarbeit einarbeiten.

Im Kapitel 2 werden zunächst die Architektureigenschaften und Grundlagen eines Token Ring Netzwerks vorgestellt. Zuerst wird darauf eingegangen, in welche Kategorie der Rechnernetze ein Token Ring Netzwerk einzuordnen ist. Danach wird kurz die Entstehungsgeschichte des Token Rings geschildert. Der letzte Punkt betrifft die Grundlagen des Token Rings: physikalische Struktur, das grundlegende Steuerungsverfahren Token Passing, Token- und Rahmenformat und Adressierungsmöglichkeiten im Token Ring.

Das Thema des 3. Kapitels ist das Übungsapplet „Token Ring Trainer“. Es werden das Rahmenkonzept, in dem das Applet realisiert wurde, der grundlegende Aufbau des Applets, Bedienung der Benutzeroberfläche und einige Implementationsaspekte vorgestellt. Anschließend gibt das Kapitel 4 eine kurze Zusammenfassung.

## 2 Token Ring

### 2.1 Netzwerk-Technologien und LAN-Topologien

In diesem Abschnitt sollen Klassifikationen von Rechnernetzen kurz dargestellt werden.

Eine Möglichkeit der Klassifikation von Rechnernetzen betrifft ihre Reichweite:

- Lokale Netze (LAN).
- Stadtnetze (MAN)
- Fernnetze (WAN)
- Globale Netze (GAN)

Man spricht von einem lokalen Netzwerk (Local Area Network), wenn die im Netz verbundenen Stationen auf eine geringe räumliche Ausdehnung (maximal 10 km) begrenzt sind, z.B. das Gebäude einer Firma. Daneben ist auch schnelle Datenübertragung und niedrige Fehlerrate charakteristisch für LANs. Lokale Netze sind private Netze, sie stehen jedoch üblicherweise nicht alleine, sondern sind mit anderen LANs und Fernnetzen verbunden.

Ein Stadtnetz (Metropolitan Area Network) deckt eine Gruppe von nahe beieinander liegenden Firmenbüros oder eine Stadt ab und es kann privat oder öffentlich sein.

Ein Fernnetz (Wide Area Network) erstreckt sich über einen großen geographischen Bereich, meist ein Land oder einen Kontinent. Ein globales Netz (GAN) ist, wie der Name schon sagt, ein erdumspannendes Netz, bei dem Rechner auf verschiedenen Kontinenten z.B. per Satellitenfunk miteinander verbunden werden.

Rechnernetze können weiterhin in zwei Kategorien hinsichtlich der Übertragungstechnik unterteilt werden:

- Netze mit Punkt-zu-Punkt-Verbindungen
- Netze mit Broadcast-Kanälen.

Punkt-zu-Punkt-Netze bestehen aus vielen Verbindungen mit separaten Kommunikationskanälen für jeweils zwei Maschinen. Diese Art der Rechnernetze eignet sich hauptsächlich für Fernnetze.

Die Idee der Broadcast-Netze besteht aus der gemeinsamen Nutzung eines physikalischen Mediums. Alle am Netz angeschlossene Maschinen empfangen alle Datenpakete.

Ideale Einsatzgebiete der Broadcast-Netze liegen in der lokalen Kommunikation.

Der wichtigste Aspekt in jedem Broadcast-Netz ist die Festlegung, wer den Kanal benutzen darf, falls darum konkurriert wird. Dafür sind einige Regeln erforderlich, die in Protokollen festgehalten werden. Protokolle, mit denen bestimmt werden kann, wer wann in einem Mehrfachzugriffskanal an die Reihe kommt, gehören zu einer Teilschicht der Sicherungsschicht namens MAC (Medium Access Control).

Die LAN-Architektur nach der Spezifikation 802 unterteilt die zweite Schicht des ISO/OSI Schichtenmodells - die Sicherungsschicht - in zwei Unterschichten: Logical Link Control und Medium Access Control (Abbildung 1).

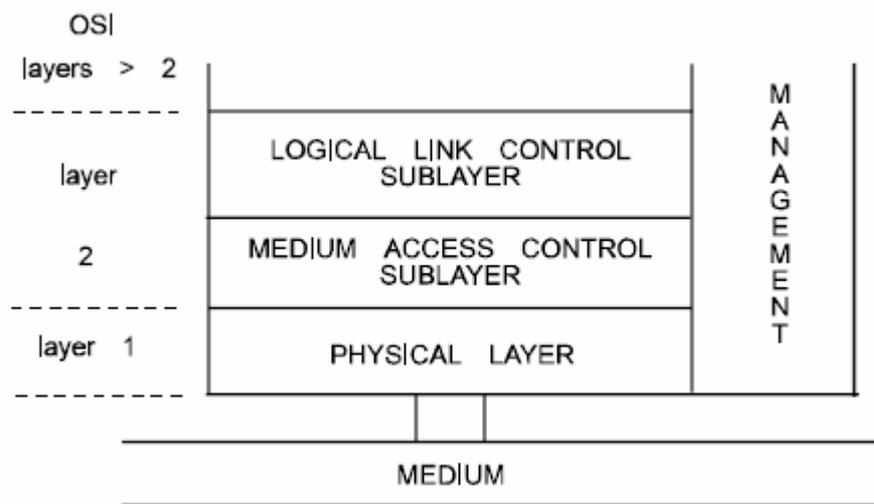


Abb. 1: Die Token Ring - Protokollelemente und ihre Relation zu den OSI-Schichten (Quelle: ANSI/IEEE Std. 802.5: *Token Ring access method and Physical Layer specifications.*)

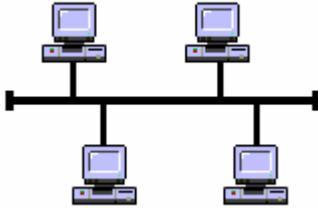
Die Sicherungsschicht (Data Link Layer) im OSI-Referenzmodell sorgt dafür, dass die Bitübertragung zwischen zwei Netzanschlüssen funktioniert. Bei lokalen Netzwerken werden in dieser Schicht folgende Punkte abgehandelt:

- Überprüfung der korrekten Datenfolge und Behebung von Übertragungsfehlern
- Flußkontrolle (Anpassung der Sendegeschwindigkeit an die Empfangsgeschwindigkeit)
- Zugang zu den Kanälen.

Von diesen Funktionen sind der Teilschicht LLC Fehlererkennung und Flußregelung zugeteilt und der MAC Teilschicht die Zugangskontrolle.

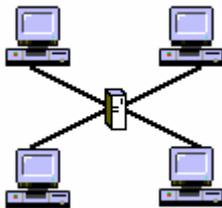
Broadcast-LANs können in verschiedenen Topologien realisiert werden. Die Topologie beschreibt die logische Verbindungsstruktur eines Netzwerks. In der Theorie ist die Klassifizierung meist eindeutig, in der Praxis finden sich allerdings auch Mischformen. Die klassischen Topologien:

- Bus-Topologie



Bei einer Bus-Topologie werden alle beteiligten Rechner an einem durchgehenden passiven Kabel angeschlossen, welches mit Endwiderständen abgeschlossen ist. Die Nachrichten bzw. Datensignale laufen von der sendenden Einheit über das gemeinsame Übertragungsmedium (Bus) nach beiden Richtungen und werden vom Zielsystem aufgenommen. Alle im Netzwerk integrierten Systeme prüfen laufend, ob Daten für sie auf dem Bus vorhanden sind, und nehmen diese ggf. entgegen. Neben der leichten Erweiterbarkeit und einfachen Verkabelung hat ein Bus-Netz den Vorteil, dass bei Ausfall eines beteiligten Rechners das Netzwerk nicht gestört wird. Ein Kabelbruch führt allerdings zum Ausfall des gesamten Netzwerks.

- Stern-Topologie

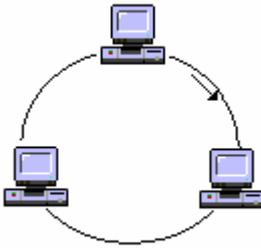


Bei einer Stern-Topologie sind alle beteiligten Rechner direkt an einem zentralen Verteiler angeschlossen. Will eine Station senden, so geht die Nachricht zuerst an den zentralen Knoten, den sogenannten Server-PC, und wird von dort aus weitergeleitet. Eine direkte Kommunikation zwischen den einzelnen Teilnehmern findet hier nicht statt, da alle Nachrichten über den zentralen Knoten laufen und dieser in Abhängigkeit von der gegebenen Zieladresse die Weiterleitung der Informationen vornimmt.

Die Kommunikation ist ähnlich organisiert wie eine Telefonzentrale, die die einzelnen Teilnehmer miteinander verbindet.

Ein Vorteil der Stern-Topologie ist die Tatsache, dass alle im Netz angeschlossenen Computer, mit Ausnahme des Netz-Servers, ihre volle Funktionsfähigkeit behalten, da die Kapazität dieser Geräte nicht durch Zusatzaufgaben belastet wird. Stern-Netzwerke haben allerdings einen großen Nachteil: bei Ausfall des zentralen Vermittlers (zentraler Knoten) wird das Netzwerk vollkommen lahmgelegt.

- Ring-Topologie



In einem Ringnetz werden die beteiligten Rechner mit jeweils einem Vorgänger und einem Nachfolger verbunden, so daß sich ein Ring schließt. Die Daten, die in einem Ring weitergegeben werden, werden von der physikalischen Nachbarstation regeneriert, bis sie schließlich zu ihrem ursprünglichen Absender zurückkommen. Jede Station ist somit Signalverstärker. Durch diese Signalregeneration bestehen keine Ausdehnungsbeschränkungen für das Netzwerk und die Anzahl der Rechner ist somit theoretisch unbegrenzt. Daneben gehören gesicherte Bandbreite und geringe Verkabelungskosten zu den Vorteilen der Ring-Architektur. Zu den Nachteilen zählt hier die Tatsache, dass der Ausfall eines Rechners zur Störung des gesamten Netzwerks führen kann.

## **2.2 Token Ring: Entstehungsgeschichte**

Das Token Ring-Steuerungsprinzip für LANs wurde bereits 1972 von Willemjin bei IBM entwickelt. Herr Willemjin hat einige Jahre lang erfolgreich gegen IBM Patent- und Lizenzklagen geführt, was ihn zu einem reichen Mann machte [2].

Einer anderen Quelle [3] zufolge besitzt jedoch Olof Söderbloom die Patentrechte für die Technologie der Datenkommunikation in Ring-Netzwerken, der diese Konzepte während seiner Arbeit am Design eines On-line Banking Netzwerks entwickelte.

Sicher ist, dass Anfang der 80er von der Arbeitsgruppe 802 des IEEE ein Standardisierungsvorschlag für u.a. Token Ring-Netzwerke vorgelegt wurde. Dieser Standard beschränkt sich auf die unteren zwei Schichten des OSI-Referenzmodells. Die fertigen Elemente der IEEE 802-Standardisierungsgruppen wurden als ISO-Standards übernommen: so hat der Standard für Token Ring-Systeme, der von IEEE 802.5 entwickelt wurde, die ISO-Nummer 8802.5.

Der von IEEE 1985 ausgearbeitete Standard 802.5 für Token Ring ist kompatibel zum IBM-Ring. In der Tat wurde die IEEE 802.5 Spezifikation erst nach dem IBM-Ring entwickelt.

Im Folgenden wird der Begriff „Token Ring“ im Bezug auf die IEEE 802.5 Spezifikation verwendet [1].

## 2.3 Grundlagen

### 2.3.1 Physikalische Struktur eines Token Ring Netzes

Token Ring gehört, wie der Name schon sagt, zu den Ring-LAN's, bei denen die Stationen sozusagen „in Reihe“ geschaltet werden und die Erste mit der Letzten verbunden wird.

Die Übertragung erfolgt unidirektional. Somit hat jede Station eine eindeutige Up- und Downstream (Vorgänger und Nachfolger) Nachbarstation. Die Daten wandern jedoch nicht direkt von einer Station zu der nächsten: jede Station besitzt eine Ringschnittstelle, welche einen Puffer von wenigen Bits besitzt. Jedes Bit, das an einer Schnittstelle ankommt, wird in den Puffer und dann wieder auf den Ring kopiert. Während sich das Bit im Puffer befindet, kann es untersucht und eventuell verändert werden, bevor es wieder ausgelesen wird. Dieser Kopierschritt führt zu einer Verzögerung von einem Bit pro Schnittstelle. Die Ringschnittstellen werden miteinander durch Punkt-zu-Punkt-Kabel verbunden (Abbildung 2).

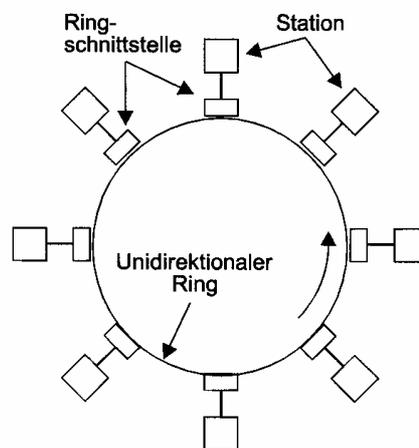


Abb. 2: Token Ring Netz

(Quelle: Tanenbaum, A.. *Computernetzwerke*, 2000)

Das IBM Token Ring - Netzwerk geht von einer physikalischen Sternverkabelung aus: es besteht aus Sicherheits-, Fehlertoleranz- und Redundanzgründen aus einer Reihe ringförmig gekoppelter Sterne, logisch verhält es sich aber wie ein Ring. Ein Kabelbruch würde in einem physikalischen Ring zur Stilllegung des gesamten Netzwerks führen, was jedoch durch ein Wire-Center (Verkabelungszentrum) gelöst werden kann (Abbildung 3).

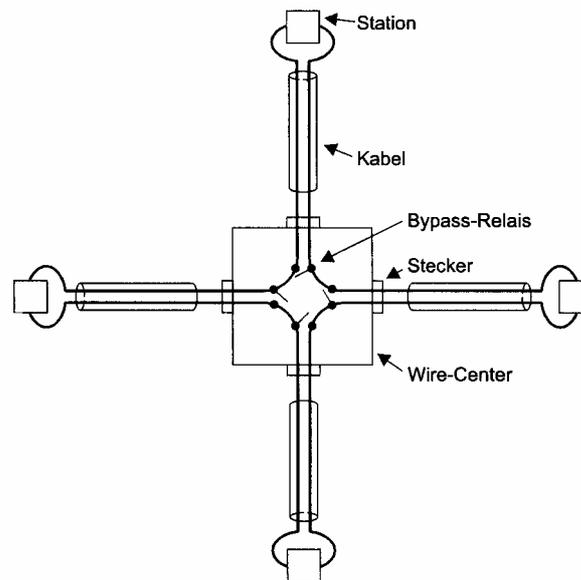


Abb. 3: Token Ring mit Wire-Center

(Quelle: Tanenbaum, A.. *Computernetzwerke*, 2000)

Jede Station wird mit einem Wire-Center verbunden, wobei das Kabel mindestens zwei verdrehte Kabelpaare enthält, von denen je eines für die Daten von und zur Station dient (jeweils ein Pfad für den Hin- und Rückweg). Im Wire-Center befinden sich Umgehungsrelais, die von den Stationen mit Strom versorgt werden. Falls der Ring unterbrochen oder eine Station ausgeschaltet wird, schließt sich durch die fehlende Versorgungsspannung das Relais und umgeht die Station. Dieses Verfahren beinhaltet eine Reihe von Vorteilen:

- Durch Kurzschluss im Wire-Center können sehr schnell fehlerhafte Anschlussleitungen oder Endgeräte abgehängt werden
- Neuanschlüsse können leicht und problemlos durchgeführt werden
- Unproblematische Fehlererkennung und Meßverfahren sind möglich.

Weitere wichtige Eigenschaften eines Token Ring-Netzes sind in der folgenden Tabelle aufgelistet:

<i>Eigenschaft</i>	<i>Beschreibung</i>
Übertragungsmedium	Lichtwellenleiter, verdrehtes Kabel, Koaxialkabel
Zugriffverfahren	Token Passing
Topologie	Sternförmig verkabelter Ring
Übertragungsrates	4 bzw. 16 Mbits/s

<i>Eigenschaft</i>	<i>Beschreibung</i>
Max. Distanz zwischen Stationen	800 m ohne Verstärker
Max. Anschlüsse pro Ring	Max. 260
Signalkodierung	Manchester Encoding- Verfahren

Die maximale Ausbaufähigkeit eines einzelnen Token Rings ist auf 250 Komponenten begrenzt. Diese Einschränkung resultiert aus den für MAC und PHY Layer spezifizierten Anforderungen, auf die vor allem die empfohlenen Timing Parameter abgestimmt sind.

Ringschnittstellen können sich in zwei Zuständen befinden: Sende- und Lesemodus.

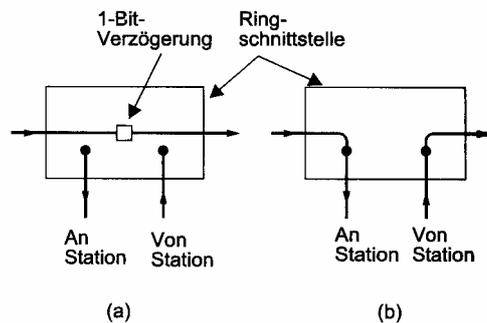


Abb. 4: Lesemodus und Sendemodus

(Quelle: Tanenbaum, A.. *Computernetzwerke*, 2000)

Im Lesemodus (Abbildung 4 a) werden die eingelesenen Bits einfach zum Ausgang kopiert. Im Sendemodus (Abbildung 4 b) unterbricht die Schnittstelle die Verbindung zwischen Ein- und Ausgabe und übergibt eigene Daten an den Ring. Die Daten werden Bit für Bit von Station zur Station weitergereicht, bis sie ihre Zielstation erreichen. Diese kopiert die Daten, markiert sie als empfangen und leitet sie auch ihrerseits weiter. Wenn die gesendeten Bits durch den ganzen Ring gewandert sind, werden sie von ihrem Absender wieder vom Ring entfernt. Der Absender kann nun die empfangenen Daten mit Originaldaten vergleichen, um die möglichen bei der Übertragung aufgetretenen Fehler zu identifizieren und eventuell die Übertragung zu wiederholen [4].

### 2.3.2 Token Passing – Steuerungsverfahren

Den im Token Ring verwendeten Steuerungsverfahren nennt man Token Passing. Will eine Station senden, so muß sie auf das sogenannte Token, genauer gesagt freies Token, warten. Ein Token stellt die Sendeberechtigung dar und ist ein 24 Bit-langes spezielles Bitmuster, das in einem Ring kreist, wenn alle Stationen untätig sind. In dem Moment, in dem das Token bei

einer sendewilligen Station ankommt, wird es von ihr vom Ring entfernt. Die Station kann nun einen Rahmen generieren und senden, in dem unter anderem ihre eigene Adresse, die Adresse des Empfängers und die zu übertragenden Informationen, die in ihrer Länge begrenzt sind, enthalten sind. Da es nur ein Token gibt, kann immer nur eine Station übertragen, wodurch das Problem des Kanalzugriffs gelöst wird. Das Token Passing Prinzip gehört somit zu den kollisionsvermeidenden Verfahren.

Die gesendeten Daten werden von Station zur Station weitergereicht, auch wenn sie selbst Daten zu übertragen hätten, bis sie ihren Absender erreichen, der sie vom Netz nimmt und ein freies Token generiert. Die nachfolgende Station kann nun senden, sofern sie Daten zu versenden hat. Ist dies nicht der Fall, so gibt sie das freie Token ohne Veränderung weiter.

Das Token Passing ist ein deterministisches Verfahren: jede Station darf nur eine begrenzte Zeit das Token für sich behalten bzw. senden. Die in IEEE 802.5 spezifizierte Zeit liegt zwischen 10 ms und 11 ms. Einerseits lässt sich dadurch die maximale Wartezeit bestimmen, bevor eine Station mit der Übertragung von Daten beginnen kann (vorausgesetzt die Stationen machen keinen Gebrauch vom Priorisierungsmechanismus). Andererseits ist damit auch die maximal mögliche Rahmengröße vorgegeben: üblicherweise beträgt sie bei 4 Mbit/s ca. 4000 Bytes und bei 16 Mbit/s ca. 18000 Bytes. Die wirkliche maximale Rahmengröße wird durch die Anwendung oder die Schnittstelle vorgegeben.

Braucht die Station für die Übertragung eines Rahmen weniger als 10 ms, so kann sie auch weitere Rahmen übertragen.

Es gibt zwei grundsätzliche Verfahren, zu welchem Zeitpunkt ein neues Token generiert werden kann:

a) *Normal token release*: Die Station gibt das Token erst dann frei, wenn sie überprüft hat, dass auch der von ihr zuletzt übertragene Rahmen bei ihr wieder ankam. Die Station gibt das Token nicht frei, wenn die Startadresse des Rahmens, den sie vom Ring entfernt, nicht mit der Startadresse des von ihr zuletzt übertragenen Rahmens übereinstimmt. Wird die Station mit der Überprüfung der Adresse fertig, während sie noch den Rahmen sendet, so wird das Token direkt an den Rahmen angehängt. Ansonsten wird das Token erst nach der Überprüfung generiert.

Dieses Verfahren gilt sowohl für 4 als auch für 16 Mbit/s Token Ringe, wobei bei 4 Mbit/s ausschließlich dieses Verfahren gilt [1].

b) *Early token release*: Das freie Token wird von der Station generiert sobald sie mit der Übertragung ihres letzten Rahmens fertig wird, d.h. sobald sie die letzten Bits des letzten Rah-

men gesendet hat. Dieses Verfahren wird aus Effizienzgründen in einem 16 Mbit/s Token Ring angewandt [1].

### 2.3.3 Token und Rahmenformate

Im Token Ring können vier unterschiedliche Symbolsequenzen umkreisen:

- Token
- Abort Sequenz
- Rahmen
- Fill.

Sowohl ein Token (Abbildung 5) als auch ein Rahmen (Abbildung 7) müssen als solche in beliebigen Datenströmen von den Stationen erkannt werden. Sie enthalten daher zwei besondere Felder, die jeweils 1 Byte lang sind:

- Starting Delimiter
- Ending Delimiter

Die Besonderheit des Starting und des Ending Delimiters liegt in ihrer Kodierung auf der physikalischen Ebene: einer bestimmten Signalfolge, die innerhalb eines Rahmens oder Tokens nicht auftreten darf.

Neben Starting und Ending Delimiter enthält ein Token ein weiteres Feld - das Acces Control - Feld (Zugriffskontrollfeld), dessen genauere Bedeutung im Zusammenhang mit Rahmenformat näher erläutert wird.

Anfangskennung Starting Delimiter 1 Byte	Zugriffskontrolle 1 Byte	Ending Delimiter 1 Byte
--	-----------------------------	----------------------------

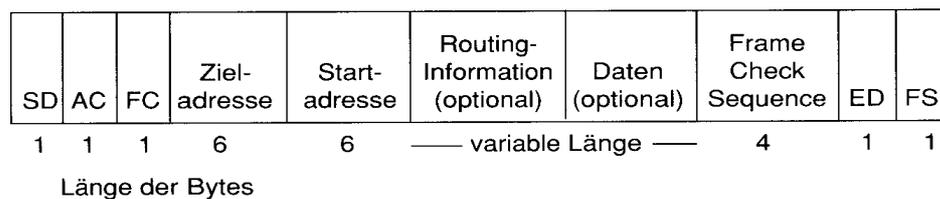
Abb. 5: Token Format (Quelle: Göhring, H.-G., Kauffels, F.-J. *Token Ring: Grundlagen, Strategien, Perspektiven*, 1990.)

Es gibt noch eine Symbolsequenz, die einen Starting und einen Ending Delimiter enthält: Abort Sequenz (Abbildung 6). Sie enthält keine weitere Felder und wird von einer Station im Ring dann erzeugt und gesendet, wenn sie einen Fehler erkennt, wie z.B. ein fehlerhaftes Token oder einen Fehler in der eigenen Arbeitsweise. Wenn die Station eine Abort Sequenz erzeugt hat, unterbricht sie die Rahmenübertragung. Wurde der Fehler in der eigenen Arbeitsweise festgestellt und ist nicht behebbar, deaktiviert sich die Station selbst.



Abb. 6: Abort Sequenz Format (Quelle: Göhring, H.-G., Kauffels, F.-J. *Token Ring: Grundlagen, Strategien, Perspektiven*, 1990.)

Das Rahmen-Format (Abbildung 7) dient zur Informationsübermittlung im Token Ring. Die beiden ersten Felder: Starting Delimiter und Access Control bilden die Startsequenz des Rahmens.

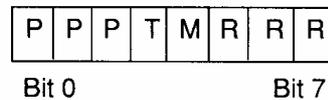


- SD = Starting Delimiter
- AC = Access Control (Zugriffskontrollfeld)
- SFS = SD + AC; Start of Frame Sequence
- FC = Frame Control
- DA = Destination Address 2 oder 6 Bytes lang)
- FCS = Frame Check Sequence (4 Bytes)
- ED = Ending Delimiter
- FS = Frame Status
- EFS = ED + FS; End of Frame Sequence

Abb. 7: Rahmen Format (Quelle: Göhring, H.-G., Kauffels, F.-J. *Token Ring: Grundlagen, Strategien, Perspektiven*, 1990.)

Das Access Control Feld (Abbildung 8) enthält zum einen das Token Bit. Handelt es sich bei der Sequenz um ein Token, so ist das Token Bit auf 0 gesetzt. In einem Rahmen ist das Token Bit auf 1 gesetzt. Das Monitor Bit hat eine Kontrollfunktion und wird von dem aktiven Monitor gesetzt, sobald er einen Rahmen oder ein Prioritätstoken empfängt. Bei der Erzeugung eines neuen Rahmens oder Prioritätstokens muss das Monitor Bit auf 0 gesetzt werden. Empfängt der aktive Monitor einen Rahmen oder Prioritätstoken mit schon gesetztem Monitor Bit, dann handelt es sich dabei um einen zirkulierenden Rahmen oder Prioritätstoken, die von der Sendestation nicht vom Netz genommen wurden. Wie der aktive Monitor auf diese Situation reagiert, wird im Abschnitt „Fehlerbehandlung“ unter Ring Purge Prozess näher erläutert. Die P Bits – Prioritätsbits – stellen die aktuelle Ring Service Priorität dar. Generell hat eine Station nur dann Zugriff auf den Ring, d.h. sie kann aus dem Token einen Informationsrahmen machen, wenn die Priorität der Daten, die die Station zu versenden hat, größer oder gleich der des Tokens ist. Wenn die Station einen Rahmen erzeugt, übernimmt sie die Prioritätsbits aus dem Token, mit anderen Worten: ein Rahmen trägt immer die gleiche Priorität,

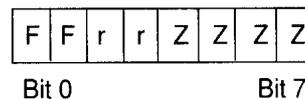
wie das Token, das für das Versenden des Rahmens empfangen wurde. Mit Hilfe der Reservierungsbits können Stationen, die Daten mit höherer Priorität zu versenden haben, festlegen, dass das nächste Freitoken mit der entsprechend hohen Priorität versehen werden soll. Auf die genaue Arbeitsweise des Priorisierungsmechanismus wird im Kapitel 2.5 näher eingegangen.



P = Bits für die Zugriffspriorität  
 T = Token Bit  
 M = Monitor Bit  
 R = Reservierungsbits

Abb. 8: Access Control Field (Quelle: Göhring, H.-G., Kauffels, F.-J. *Token Ring: Grundlagen, Strategien, Perspektiven*, 1990.)

Mit Hilfe der F-Bits im Rahmen-Kontrollfeld (Frame Control) kann unterschieden werden, ob es sich bei dem Rahmen um einen LLC Frame (bzw. Daten-Rahmen) oder einen MAC Frame (Steuerungsinformationen der Schicht 2) handelt. Die MAC Frames müssen von Stationen vorrangig verarbeitet werden.



F = Frame-Typ Bit  
 B'00' = MAC-Frame  
 B'01' = LLC-Frame  
 B'10' oder B'11' vorerst nicht definiert  
 r = reserviert  
 Z = Kontroll Bit

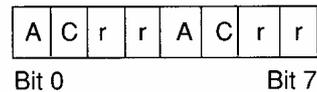
Abb. 9: Frame Control Field (Quelle: Göhring, H.-G., Kauffels, F.-J. *Token Ring: Grundlagen, Strategien, Perspektiven*, 1990.)

Als nächstes kommen die Felder Destination Address (Zieladresse) und Source Address (Startadresse), die jeweils 6 Byte lang sind (s. Kapitel 2.3.4: Adressierung im Token Ring). Die beiden nächsten Felder Routing Information und Information (oder Daten) sind optional. Die Einschränkung der Länge des Datenfeldes kommt nur dadurch, dass die Station in einer begrenzten Zeit (Token Holding Time) die Übertragung absolvieren soll.

Für die Prüfsumme (Frame Check Sequence) sind im Rahmenformat 4 Bytes vorgesehen. Die Prüfsumme basiert auf einem Prüfpolynom des 32. Grades (CRC Verfahren) und deckt alle Felder mit Ausnahme des Frame Status - Feldes und des Ending Delimiters ab.

Das 1 Byte lange Rahmenstatus - Feld (Abbildung 10) enthält die Bits A und C. Wenn ein Rahmen bei der Schnittstelle der Zielstation ankommt, setzt diese Schnittstelle das A Bit,

konnte der Rahmen von der Schnittstelle auf die Station auch kopiert werden, dann setzt sie auch das C Bit. Die Sendestation untersucht die A und C Bits, wenn sie ihren Rahmen wieder von Ring entfernt. Nur wenn beide Bits auf 1 gesetzt sind, war die Übertragung erfolgreich. Sind beide Bits nicht gesetzt, konnte die Zielstation nicht erreicht werden. Ist nur das A Bit gesetzt, so ist das Kopieren des Rahmen fehlgeschlagen. Andere Kombinationsmöglichkeiten sind nicht zulässig. Die beiden Bits kommen doppelt vor, da das Rahmenstatus - Feld ausserhalb der Prüfsumme liegt und nicht durch sie gegen Fehler abgesichert sind. Zusammen mit dem Ending Delimiter bildet das Rahmenstatus - Feld die Endsequenz des Rahmens.



- A = Address-Recognized Bit  
 C = Frame-Copied Bit  
 r = reserviert, momentan ohne Funktion

Abb. 10: Frame Status Field (Quelle: Göhring, H.-G., Kauffels, F.-J. *Token Ring: Grundlagen, Strategien, Perspektiven*, 1990.)

Eine Fill-Sequenz ist eine beliebige Kombination aus zulässigen Datensymbolen. Eine Station im Sendemodus muss anschließend an ihre Daten Fill übertragen, bis sie ihre Daten wieder empfängt und ein Freitoken generiert.

### 2.3.4 Adressierung im Token Ring

Es gibt drei Möglichkeiten der Adressierung im Token Ring:

- Individuelle Adressierung
- Gruppenadressierung
- Broadcast - Adressierung

Individuelle Adressierung betrifft eine einzige Station im Ring. Die Adresse muß daher innerhalb des Rings eindeutig sein. Nur diese eine Station wird den Rahmen kopieren und ihn als empfangen und kopiert markieren.

Muß ein Rahmen an mehrere Stationen gleichzeitig adressiert werden, so verwendet man eine Gruppenadresse. In diesem Fall ist die Station in der Gruppe, die den Rahmen als erste empfängt, auch die einzige, die den Empfang des Rahmen bestätigen kann, d.h. sie setzt die A und C Bits.

Broadcast-Adressierung wird vor allem für Managementfunktionen verwendet. Ein Rahmen mit einer Broadcast-Adresse ist für alle aktiven Stationen im Ring bestimmt.

## 2.4 Monitoring in Token Ring

Im Normalfall läuft das Token- Verfahren ohne weiteren Eingriff. Gehen aber Token verloren, fehlen Token oder „vergessen“ Stationen ihre Rahmen vom Netz zu nehmen etc., so wird ein Ringwartungsmechanismus notwendig. Die Ringwartung wird im Token Ring dezentralistisch gelöst. Sie basiert auf dem Vorhandensein eines Monitors. Jeder Ring besitzt genau eine Station, die die Funktion eines aktiven Monitors übernimmt. Alle anderen aktiven Stationen im Ring übernehmen die Funktion des Standby-Monitors.

Der Ringwartungsmechanismus besteht aus drei Prozessen, wie die Abbildung 11 Ring-Protokoll-Hierarchie zeigt, und ist hierarchisch aufgebaut:

Stellt das Protokoll mithilfe des aktiven Monitors und/oder der Standby-Monitore eine Unterbrechung des regulären Token Protokolls fest (einen Soft- oder Harderror), so geht der entsprechende Monitor ohne Ausnahmen die Hierarchie herunter und hoch wie folgt:

- war ein Prozess erfolgreich, wird der in der Hierarchie nächst höhere Prozess gestartet
- misslingt ein Prozess, wird der Prozess auf der nächst tieferen Ebene gestartet.

Die Abbildung 11 stellt den Aufbau der Hierarchie der Ringwartungsprozesse und die Übersicht der Funktionen eines aktiven und eines Stand-by-Monitors dar.

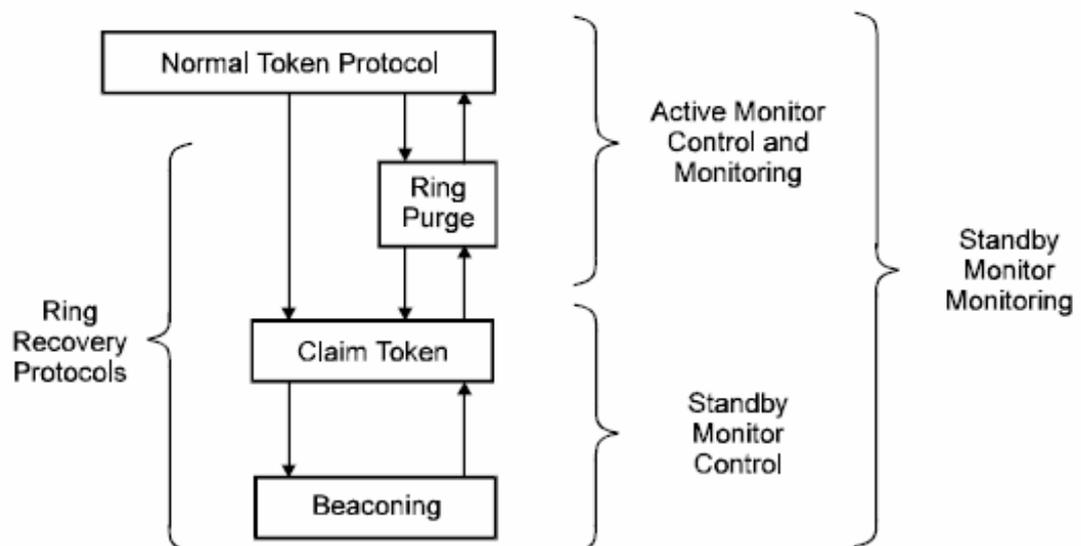


Abb. 11: Ring-Protokoll –Hierarchie (Quelle: ANSI/IEEE Std. 802.5: *Token Ring access method and Physical Layer specifications.*)

Der aktive Monitor kontrolliert und überwacht den Ablauf des regulären Token Protokolls. Stellt er aber einen Fehler fest, so stößt er den Ring Purge-Prozess und damit die Neuinitiali-

sierung des Rings an. War der Ring Purge-Prozess erfolgreich, generiert der aktive Monitor ein Freitoken und der Ring ist wieder im normalen Token- Betrieb. Konnte der Ring-Purge-Prozess nicht erfolgreich abgeschlossen werden, dann startet der aktive Monitor den Claim-Token-Prozess und gibt seine Rolle des aktiven Monitors auf. Der Claim-Token-Prozess kann auch von einem Stand-by-Monitor gestartet werden, wenn er die Abwesenheit oder die Fehlfunktion des aktiven Monitors feststellt.

#### 2.4.1 Funktionen des aktiven und des Stand-by-Monitors

Der aktive Monitor ist eine Station im Ring, die für einen korrekten Ablauf des Token Protokolls zuständig ist. Zu seinen Aufgaben gehören:

- Kontrolle des Timing und Synchronisation aller Timer im Ring
- Sicherstellung der minimalen Ring-Speicherfähigkeit von 24 Bits
- Überwachung der Übertragung von Token und Rahmen
- Erkennung verlorengangener Token oder Rahmen
- Erkennung zirkulierender Pioritätstoken oder Rahmen
- Wiederherstellung des normalen Token Betriebs im Fehlerfall
- Initiierung und Überwachung des Neighbour Notification – Verfahren

Jeder Stand-by-Monitor (alle anderen aktiven Stationen im Ring) überwacht den aktiven Monitor auf Fehlfunktionen und kontrolliert den Ablauf der Claim Token –und Beaconing Prozesse.

#### 2.4.2 Fehlerbehandlung

Im Token Ring wird zwischen Soft - und Hard - Errors unterschieden. Unter dem Begriff "Softerror" werden Defekte, die Verfälschungen der Daten verursachen, aber das Umkreisen der Daten nicht verhindern, verstanden (z.B. zirkulierendes Pioritätstoken, zirkulierender Rahmen etc.). Um solche Softerrors zu beheben, wird der Ring Purge Prozess oder das normale Token Protokoll verwendet. Als "Harderrors" werden Defekte, die das Umkreisen des Token und/oder des Frames in dem Ring hindern, bezeichnet. Ein Harderror kann z.B. ein Kabelbruch sein. Wenn eine Station einen Harderror entdeckt, startet sie den Beacon Prozess. In regelmäßigen Abständen wird vom aktiven Monitor ein Neighbor Notification Prozess initiiert. Dieser wird benötigt, um den Stationen die Adresse des aktiven Monitors mitzuteilen

und sie mit ihren jeweiligen Vorgängern bekannt zu machen. Diese Information ist zur Reparatur des Rings im Fehlerfall notwendig.

### 2.4.3 Ring Purge Process

Das Ziel des Ring Purge Prozesses ist eine Neuinitialisierung des Rings: der Ring wird geleert und ein neues Token wird generiert und freigegeben. Es gibt zwei Gründe, warum der aktive Monitor den Ring Purge Prozess startet:

- 1) der Claim Token Prozess wurde erfolgreich abgeschlossen, d.h. eine Station wurde zu neuem aktiven Monitor, und der Ring Purge Prozess ist nach dem hierarchischen Prinzip der nächste, der gestartet werden muß.
- 2) der aktive Monitor entdeckt einen Softerror der folgenden Art:
  - a. das Fehlen eines Tokens auf dem Ring: ein Token muss immer innerhalb eines festen Zeitrahmens an dem Monitor vorbeigehen. Läuft diese Zeit ab, dann geht der Monitor von einem Softerror aus.
  - b. ein zirkulierender Rahmen wird vom aktiven Monitor identifiziert: er empfängt einen Rahmen mit gesetztem Monitor Bit, was darauf zurückzuführen ist, dass die Sendestation ihren Rahmen nicht vom Ring genommen hat. Denn bei der Neuerzeugung eines Rahmens wird das Monitor Bit immer mit dem Wert 0 übertragen.
  - c. ein zirkulierendes Prioritätstoken wird vom aktiven Monitor identifiziert: analog zum zirkulierenden Rahmen ist auch hier das Monitor Bit gesetzt. Jedes Prioritätstoken wird vom aktiven Monitor "markiert" (das Monitor Bit wird gesetzt), wenn er es zum ersten Mal empfängt. Kommt das Prioritätstoken mit schon gesetztem Monitor Bit, so wurde es von seiner Sendestation nicht wieder vom Ring genommen.

In allen diesen Fällen wird vom aktiven Monitor Ring Purge MAC Frame gesendet und veranlasst alle Stationen, ihre Sendevorgänge abubrechen und die Timer neu zu initialisieren. Der aktive Monitor kann auch mehrere Ring Purge MAC Frames versenden, bevor er das normale Token Protokoll startet. Empfängt der aktive Monitor seinen eigenen Ring Purge MAC Frame innerhalb einer bestimmten Zeitspanne wieder, dann erzeugt er ein Freitoken, und der Ring ist wieder im Normalbetrieb. Konnte der Prozess innerhalb dieser Zeitspanne nicht beendet werden, dann startet der aktive Monitor den Claim Token Prozess und gibt seine Funktion als aktiver Monitor auf. Es kann auch vorkommen, dass der aktive Monitor einen Ring Purge Frame empfängt, der nicht von ihm erzeugt wurde (die Startadresse des MAC-

Frames entspricht nicht der Adresse des aktiven Monitors). Auch in diesem Fall gibt er seine Funktion als aktiver Monitor auf.

#### 2.4.4 Claim Token Process

Das Ziel des Claim Token Prozesses ist es, einen neuen aktiven Monitor festzulegen. Das Starten dieses Prozesses kann in drei Fällen vorkommen:

- 1) der Beacon Prozess wurde abgeschlossen,
- 2) der Ring Purge Prozess scheiterte oder
- 3) eine Station stellte fest, dass der aktive Monitor seine Funktionen nicht korrekt ausführt. Alle Stand-by-Monitore überwachen den normalen Ablauf des Token Protokolls und somit auch den aktiven Monitor. Stellen sie aber fest, dass nach Ablauf von bestimmten Zeitspannen der aktive Monitor einen vorgekommenen Fehler nicht beseitigt, gehen sie von seiner Fehlfunktion aus.

Stationen, die die Berechtigung haben, ein aktiver Monitor zu werden (der letzte aktive Monitor hat sie z.B. nicht, da er seine Funktion als aktiver Monitor nicht korrekt erfüllte), konkurrieren miteinander darum, wer der nächste aktive Monitor wird, indem sie Claim Token MAC Rahmen (CT) versenden. Im Endeffekt gewinnt diejenige Station von den am Prozess aktiv beteiligten Stationen, die die höchste MAC - Adresse besitzt. Sie startet dann den Ring Purge Prozess. Im Detail läuft der Claim Token Prozess auf folgende Weise ab:

- 1) Die aktiv beteiligte Stationen können sich im Laufe des Prozesses in zwei Zuständen befinden: dem Sendezustand (Station sendet selbst CT) oder Lesezustand (Station „gibt sich geschlagen“ und nimmt nur passiv an dem Prozess teil). Alle Stationen, die an dem Claim Token Prozess nicht beteiligt sind, und der letzte aktive Monitor gehen in den Lesezustand über. Ob die Station an dem Prozess teilnehmen kann, hängt davon ab, ob die entsprechende Option bei ihr gesetzt wurde. Allerdings muss diejenige Station, die den Fehlerzustand erkennt, an dem Prozess teilnehmen, unabhängig von der gesetzten Option.
- 2) Alle aktiv beteiligte Stationen, die sich noch nicht im Sendezustand befinden, vergleichen die Startadresse des empfangenen CT mit der eigenen MAC Adresse. Ist die Startadresse höher als die eigene, dann geht die Station in den Lesezustand über und wartet, bis der Prozess beendet wird. Ist die Startadresse kleiner als die eigene, dann geht die Station in den Claim Token Sendezustand über, generiert und sendet einen CT und wartet auf seinen eigenen Claim Token MAC Rahmen oder auf einen CT mit

höherer Adresse. Ist die Startadresse gleich der empfangenen und die Station befindet sich noch nicht im dem Sendezustand, dann nimmt sie den Lesezustand an: diese Situation ist ein Indikator dafür, dass eine weitere Station die gleiche MAC Adresse im Ring hat.

- 3) Eine Station im Sendezustand wird normalerweise alle 20 ms einen CT versenden. Sie sendet Fill, wenn sie nicht gerade ein CT sendet, und leitet keine empfangenen Rahmen weiter. Deswegen wird keine Station im Sendemodus ihren eigenen CT wieder empfangen, solange sich noch weitere Stationen im Ring im Sendemodus befinden. Stationen im Sendezustand vergleichen ebenfalls die Startadresse des empfangenen CT mit der eigenen MAC Adresse. Ist die Adresse kleiner als die eigene, sendet die Station erneut ihren CT. Ist die Adresse größer als die eigene, dann geht die Station in den Lesemodus über. Ist die Adresse gleich der eigenen, dann kann anhand der im CT übertragenen Adresse der Nachbarstation vor der Sendestation (UNA) festgestellt werden, ob der Rahmen auch wirklich von der Station stammt. Jede Station kennt die Adresse der vor ihr liegenden Station, daher kann sie die UNA mit der bei ihr gespeicherten Adresse der Nachbarstation vergleichen. Stimmen sie überein, dann hat die Station ihren eigenen CT empfangen und damit auch das Verfahren gewonnen. Sie wird zum neuen aktiven Monitor und startet den Ring Purge Prozess. Alle anderen Station befinden sich im Lesemodus und wechseln bei Empfang des Ring Purge Rahmen in den normalen Lesemodus. Stimmt die UNA Adresse nicht mit der Nachbaradresse überein, ist das ein Zeichen dafür, dass eine weitere Station die gleiche MAC Adresse hat, und die Station geht in den Lesemodus über.
- 4) Alle Station im Claim Token Prozess überwachen den Ablauf des Prozesses anhand des Timer Claim Token (TCT). Der Timer wird zurückgesetzt, wenn die Station entweder in den Claim Lese- oder Sendemodus übergeht. Läuft der Timer ab und die Station befindet sich immer noch in einem dieser Zustände, dann startet sie den Beacon Prozess. Der Wert des Timers liegt bei 1 sec.

#### 2.4.5 Beacon Process

In diesem Prozess geht es vor allem darum, aufgetretene Netzwerkfehler zu erkennen und zu lokalisieren. Eine Station X, die nach Ablauf einer bestimmten Zeitspanne (Timer Signal Loss zwischen 200 und 250 ms) überhaupt keine Rahmen oder Token mehr empfangen hat, nimmt ein Versagen der vor ihr liegenden Station Y (Upstream Neighbour) an und beginnt Beacon

MAC Rahmen zu senden. Dieser Rahmen enthält die Adresse der Sendestation und ihrer Vorgängerin, die der Station beim letzten Neighbor Notification Prozess mitgeteilt wurde. Alle Stationen, die Beacon MAC Rahmen erhalten, senden diese unverändert weiter und stoppen das Token Passing Protokoll: der Ring befindet sich im Beaconing Zustand. Weiterhin wird dieser Prozess nach dem Scheitern des Claim Token Verfahrens gestartet.

## 2.5 Priorisierungsmechanismus

Der Priorisierungsmechanismus bietet den Stationen die Möglichkeit ihre Rahmen vorrangig zu transportieren.

Grundsätzlich gilt: will eine Station Daten mit höherer Priorität als 0 übertragen, muß sie ein Prioritätstoken anfordern. Dies geschieht, indem sie die Reservierungsbits des von ihr weitergeleiteten Access Control Feldes eines Rahmen (oder eines Prioritätstoken) gleich der gewünschten Priorität setzt. Für die Reservierung als auch für die aktuell genutzte Priorität sind jeweils 3 Bits im Access Control Feld vorgesehen, somit können 8 unterschiedliche Prioritätsstufen angefordert werden. Jedoch können nur 4 Stufen von den Anwendungen genutzt werden, der Rest dient Netzwerkmanagement-Funktionen:

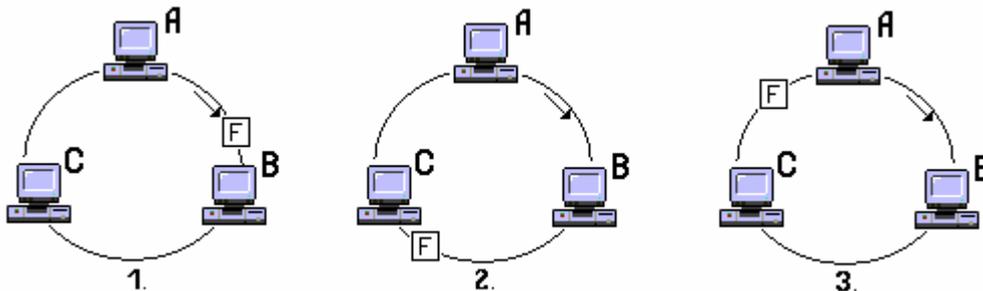
<i><b>Priorität</b></i>	<i><b>Anwendung</b></i>
0	Freie Nutzbarkeit; von den meisten Anwendungen benutzt
1-3	Freie Nutzbarkeit
4	Von Bridges benutzt
5-6	Reserviert, aber ungenutzt
7	Vom Ringmanagement genutzt

Tab. 2: In Token Ring benutzte Prioritäten (Quelle: Steinmetz, R. Multimedia-Technologie: Grundlagen, Komponenten und Systeme, 2000)

Wie genau die Nutzung des Priorisierungsmechanismus im Ring erfolgen kann, soll an folgenden Beispielen gezeigt werden.

Generell gilt: Eine Station setzt die Übertragung solange fort, bis entweder die PDU vollständig übertragen worden ist oder bis die Übertragung wegen Überschreitung der THT abgebrochen wird.

- **Szenario 1:** Station A ist gerade am Senden mit Priorität 0, ihre Daten sind an die Station B adressiert, Station B hat Daten mit Priorität 0 an C zu versenden, Station C hat Daten mit Priorität 1 an B zu versenden. Der aktive Monitor im Ring ist die Station B.

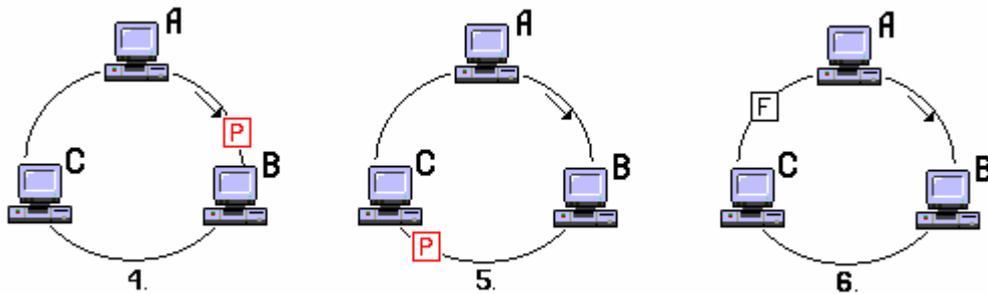


Jede Station speichert den Wert der aktuellen Priorität (Prioritätsbits) und den Wert der Reservierung (Reservierungsbits) des bei ihr gerade angekommenen Rahmen bzw. Tokens oder Prioritätstokens in den Registern Pr resp. Rr.

(1.) Station B ist der aktive Monitor im Ring, daher setzt sie das Monitor Bit im gerade empfangenen Rahmen. Als nächstes kopiert sie den Rahmen (setzt die A und C Bits im Frame Status Feld) und leitet ihn weiter. Die Werte der Pr und Rr Registern sind beide 0.

(2.) Der Rahmen der Station A kommt bei der Station C an. Hat eine Station Daten mit höherer Priorität als 0 zu versenden und sie empfängt einen Rahmen, dann vergleicht sie den Wert des Rr – Registers (Reservierungsbits) mit der Priorität der eigenen zu versendenden Daten (Pm). Ist dieser Wert kleiner als die Priorität ihrer Daten ( $R < P_m$ ), setzt sie die Reservierungsbits des Rahmen gleich der gewünschten Priorität ( $R := P_m$ ). Ist der Wert des Rr Registers größer oder gleich der eigenen Priorität ( $R \geq P_m$ ), leitet die Station den Rahmen unverändert weiter. Die Reservierung kann auch in einem Ring Purge Frame vorgenommen werden. C setzt nun die Reservierungsbits auf 1.

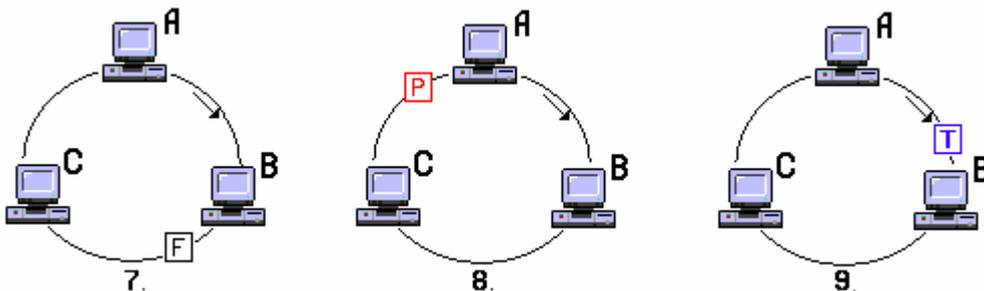
(3.) A empfängt nun ihren eigenen Rahmen wieder. Bevor sie ein Token wieder freigibt, muss sie prüfen, ob von irgendeiner Station die Reservierung gesetzt wurde. Ist die Reservierung größer als die aktuelle Ring Service Priorität (Priorität, mit der der letzte Rahmen gesendet wurde), so erzeugt die Station ein neues Prioritätstoken, dessen Priorität gleich der reservierten Priorität ist und dessen Reservierungsbits wieder auf 0 gesetzt werden. Die Station wird zu einer Stacking Station (Station, die für das zurücksetzen der Priorität zuständig ist) und merkt sich den letzten Wert der Ring Service Priorität und den neuen aktuellen Wert. Station A erzeugt ein Prioritätstoken mit Priorität gleich 1 und Reservierung gleich 0.



(4.) Das Prioritätstoken kommt bei der Station B an. Obwohl sie Daten zu versenden hat, muss sie das Prioritätstoken weiterleiten, da seine Priorität größer ist, als die Priorität der Daten der Station B. Eine Station darf immer nur das Token zum versenden von Daten greifen, dessen Priorität kleiner oder gleich der Priorität der eigenen zu versendenden Daten ist.

(5.) Station C empfängt das Prioritätstoken und fängt ihre Datenübertragung an. Die Priorität des gesendeten Rahmens entspricht immer der Priorität des Tokens, das für die Übertragung empfangen wurde. Der Rahmen hat Priorität 1 und Reservierung 0.

(6.) Station A leitet den Rahmen unverändert weiter. Station B setzt das Monitor Bit und markiert den Rahmen als empfangen und kopiert bzw. setzt die A und C Bits und leitet ihn weiter.



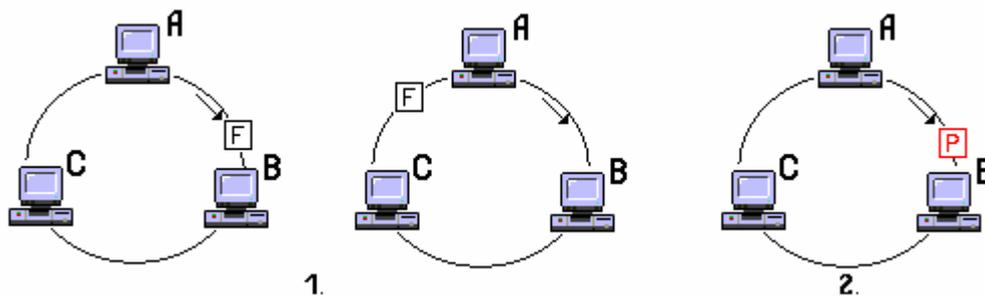
(7.) Station C empfängt ihren eigenen Rahmen wieder. Die Station gibt immer das Token wieder frei, das sie auch zum versenden ihrer Daten verwendet hat. Es sei denn, eine andere Station würde eine Reservierung im Rahmen vornehmen. Wie in so einem Fall verfahren wird, wird im Szenario 2 gezeigt. Station C erzeugt ein Prioritätstoken mit Priorität 1 und Reservierung 0.

(8.) Station A empfängt das Prioritätstoken. Wenn die Station eine Stacking Station ist, muss sie bei jedem Token, das sie empfängt, prüfen, ob das ein Prioritätstoken ist, und ob es von ihr generiert wurde. Dies geschieht indem sie die Tokenpriorität mit dem bei ihr gespeicherten Wert der „neuen“ Ring Service Priorität vergleicht, und wenn die Werte gleich sind, kann die Station die Priorität wieder zurücksetzen und ihre Rolle als Stacking Station beenden, vorausgesetzt es wurden keine weiteren Reservierungen im Prioritätstoken vorgenommen (s.

Szenario 2). Station A identifiziert das empfangene Prioritätstoken als das von ihr erzeugte, nimmt das Token vom Ring und erzeugt ein neues „normales“ Freitoken. A beendet ihre Rolle als Stacking Station.

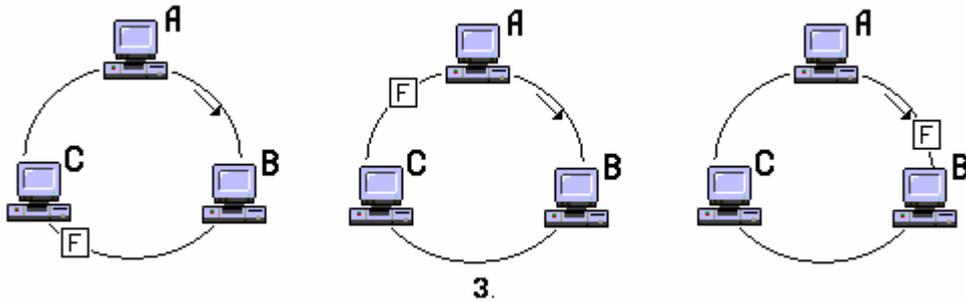
(9.) Station B kann nun ihre Daten versenden.

**Szenario 2:** Station A ist gerade am Senden mit Priorität 0, sie hat noch ein weiteres Datenpaket zu versenden mit Prioritäten 1. Station B hat auch Daten mit Priorität 2 zu versenden. Als die Station B zu senden beginnt, entscheidet sich die Station C auch ein Datenpaket zu senden mit Priorität 3. Station B ist der aktive Monitor im Ring.

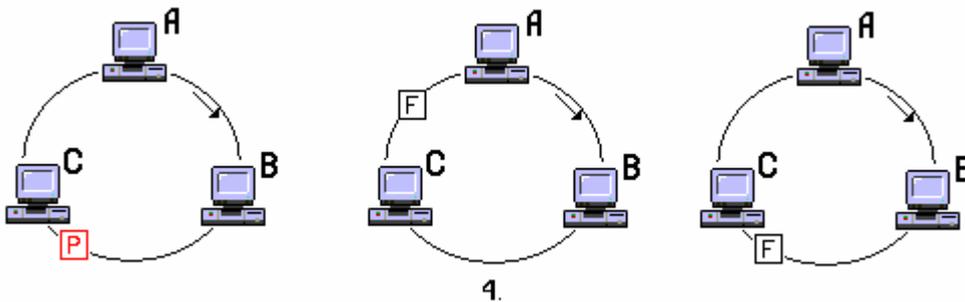


(1) Der Rahmen der Station A kommt bei der Station B an. B setzt das Monitorbit auf 1 und die Reservierungsbits im empfangenen Rahmen, die beim Erzeugen des Rahmen auf Null gesetzt wurden, auf 2 und leitet den Rahmen weiter. Station C hat zu diesem Zeitpunkt noch keine Daten zu versenden, daher leitet sie den Rahmen unverändert weiter.

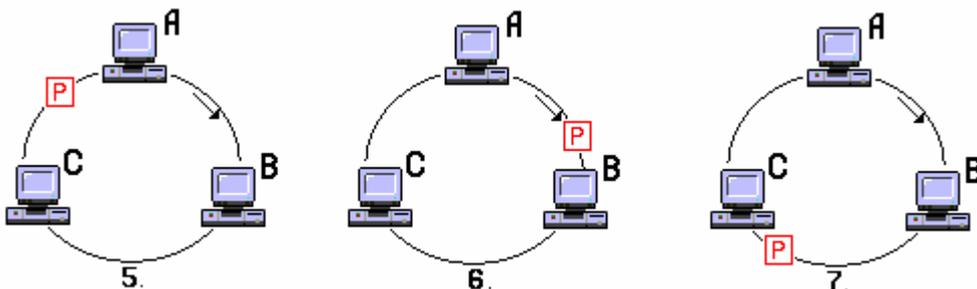
(2) Die Station A empfängt ihren Rahmen wieder und stellt fest, dass eine Reservierung vorgenommen wurde, die höher ist als die Priorität der Daten, die A selbst zu versenden hat. Wenn bei der Station eine PDU mit Priorität  $P_m$  zur Übertragung bereitsteht oder eine Reservierung ( $R_r$ ) vorliegt, wobei beide höher als die aktuelle (Ring-)Servicepriorität sind, wird ein Token generiert mit der höheren der beiden  $P_m$  und  $R_r$  Priorität und den Reservierungsbits gleich 0. Ab dem Zeitpunkt, an dem die Station den (Ring-) Serviceprioritätslevel erhöht hat, wird sie zur Stacking Station und speichert als solche den Wert der alten (Ring-)Servicepriorität als  $S_r$  und den neuen als  $S_x$  (Stack Operation). Diese Werte werden später dafür benutzt, die (Ring-) Servicepriorität wieder herabzusetzen, wenn keine PDUs zur Übertragung mehr vorliegen, dessen  $P_m \Rightarrow S_x$  ist. A generiert ein Prioritätstoken mit  $P = 2$  und  $R = 0$  und wird dadurch zur Stacking Station.



(3) Station B empfängt das Prioritätstoken und beginnt mit der Übertragung ihrer Daten. C hat zu diesem Zeitpunkt auch Daten mit Priorität 3 zur Übertragung bereit. Sie setzt nun die Reservierung im empfangenen Rahmen auf 3. Der Rahmen kommt bei der Station A an, sie hat zwar auch Daten mit Priorität  $> 0$  zu übertragen, aber die gesetzte Reservierung ist größer, als die Priorität ihrer Daten, so muß sie den Rahmen unverändert weiterleiten.



(4) Station B empfängt ihren Rahmen wieder und stellt fest, dass eine Reservierung vorgenommen wurde, die höher ist, als die aktuelle (Ring-) Servicepriorität. B wird zur Stacking Station und speichert den Wert der alten Servicepriorität  $S_r = 2$  und den neuen Wert als  $S_x = 3$ . Station C kann das Prioritätstoken zur Übertragung ihrer Daten nutzen. A nimmt diesmal eine Reservierung vor und setzt die Reservierungsbits im empfangenen Rahmen auf 1. B setzt das Monitorbit auf 1.

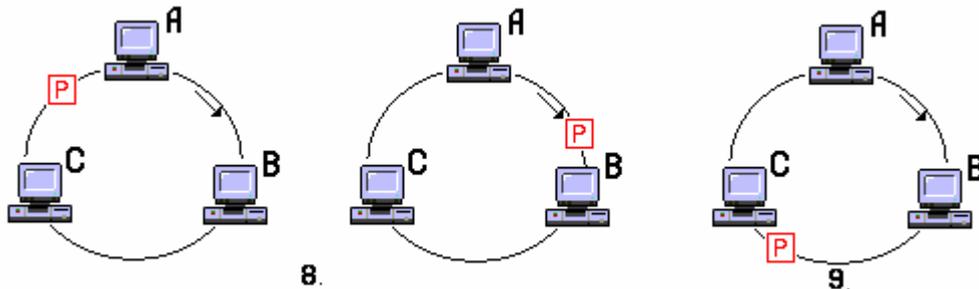


(5) Station C generiert nach dem Empfang des eigenen Rahmens ein Prioritätstoken mit  $P = 3$  und  $R = 1$ . Wenn die Station, die ihren Rahmen wiederempfängt, keine zusätzliche PDUs zu übertragen hat, deren Priorität ( $P_m$ ) höher als die aktuelle (Ring-) Servicepriorität ist, und es liegen keine Reservierungen vor (im Register  $R_r$ ), die höher als die aktuelle (Ring-) Servi-

cerpriorität sind (enthalten im Register Pr), so wird das Token mit der aktuellen (Ring-) Servicepriorität und dem Reservierungsbit R, der den höheren der beiden Rr und Pr Werte annimmt, gesendet und es werden keine weiteren Aktionen durchgeführt.

(6) A leitet das Prioritätstoken unverändert weiter.

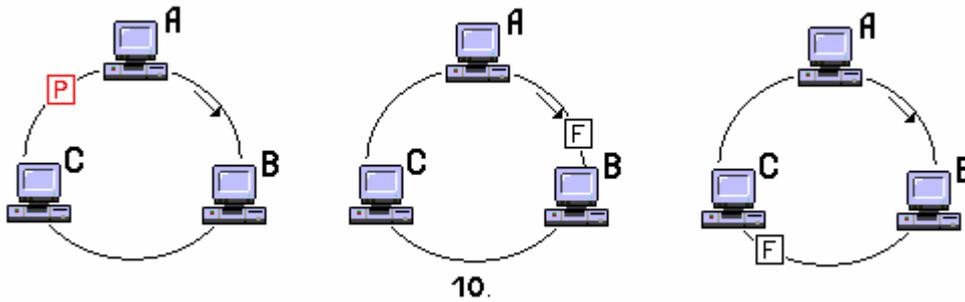
(7) Station B ist Stacking Station und identifiziert das empfangene Prioritätstoken als das von ihr erzeugte. Wenn die Reservierung Rr und die Priorität der evtl. zu versenden Daten (Pm) kleiner oder gleich dem Wert Sr sind, wird ein neues Token generiert mit der Priorität Sr und beide Sx und Sr werden aus den Stacks entfernt (POP operation). Wenn keine Sx und Sr Werte gespeichert wurden, beendet die Station ihre Rolle der Stacking Station. Die R bits des neuen Token werden entweder dem Rr oder Pm gleichgesetzt, je nachdem welcher der größere ist. Station B generiert ein neues Prioritätstoken mit Priorität 2 und Reservierung 1, entfernt die Werte Sr = 2 und Sx = 3 aus den Stacks und beendet ihre Rolle als Stacking Station.



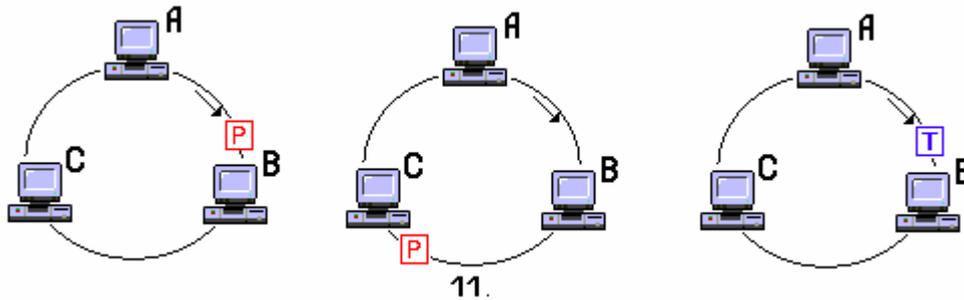
(8) C leitet das Prioritätstoken weiter. Station A identifiziert das empfangene Prioritätstoken als das von ihr erzeugte. Wenn die Station keine zu übertragende PDU's mit aktueller (Ring-) Servicepriorität hat, wird ein neues Token generiert, dessen P bits entweder dem Wert der Reservierungsbits R, dem Wert Sr oder dem Wert Pm entsprechen, je nachdem welcher der drei Werte der größere ist. A generiert ein neues Prioritätstoken mit  $P = 1$ .

Wenn der Wert der neuen (Ring-)Servicepriorität größer ist als Sr ( $P = Rr$  oder  $Pm$ ), dann werden die R bits mit 0 gesendet, der alte Sx Wert wird durch den neuen P Wert ersetzt und die Station setzt ihre Rolle als Stacking Station fort (RESTACK operation). A ersetzt den Wert 2 im Stack, in dem die neue Servicepriorität gespeichert wird, durch den Wert 1 und setzt ihre Rolle als Stacking Station fort.

(9) B setzt das Monitorbit auf 1. C leitet das Prioritätstoken weiter.



(10) A empfängt das für die Übertragung ihrer Daten benötigte Token und startet die Datenübertragung mit  $Priorität = 1$ . B setzt das Monitorbit aus 1.



(11) A empfängt ihren Rahmen wieder und generiert erneut ein Prioritätstoken mit  $P = 1$ . Wenn das Prioritätstoken wieder bei A ankommt (die einzige Veränderung am Token wurde durch B vorgenommen: das Monitorbit = 1), entfernt A die Werte  $Sr = 0$  und  $Sx = 1$  aus den Stacks. Da keine weitere Werte in den beiden Stacks enthalten sind, beendet A ihre Rolle als Stacking Station und generiert ein "normales" Token.

Weitere Anmerkungen zum Prioritätenmechanismus:

- Eine Station kann die (Ring-)Servicepriorität mehrmals erhöhen, bevor sie sie wieder zurücksetzt (z.B. von 1 zu 3 und dann 5 zu 6). Die Station kann mehrfache Einträge von  $Sx$  und  $Sr$  haben.
- Die Rahmen, die gesendet werden um den Ring zu initialisieren (Ring Purge Rahmen), haben im P Feld den Wert 0 stehen. Ein Empfang des P Feldes, dessen Wert kleiner ist als der gespeicherte  $Sx$ , führt zur Reinigung des Stack.

## 3 Token Ring Trainer

### 3.1 E-Learning-Grundlagen des Übungsapplets TokenRingTrainer

Das Applet TokenRingTrainer stellt als ein Übungsapplet einen Teil des Communication and Tutoring Systems, das an der Fakultät für Praktische Informatik IV der Universität Mannheim entwickelt wurde, dar. CATS ist ein Online Tutoring System, welches in erster Linie für Fernstudenten gedacht ist und auf den Konzepten aus dem Bereich des E-Learning basiert.

Unter E-Learning wird hier, vereinfacht ausgedrückt, das Lernen in/mit einer computerunterstützten Lernumgebung verstanden. Die Effektivität der Wissensvermittlung durch E-Learning-Programme hängt unter anderem in hohem Maße vom Instruktionsdesign der Lernumgebung ab. Der Begriff der Instruktion umfasst allgemeine Anleitungen jeglicher Art, die dem Nutzer die Orientierung in einem Lernsystem ermöglichen. Konkret wird mit dem Begriff Instruktionsdesign vor allem das Design der Benutzeroberfläche von Software verbunden [6].

Die Aktivität des Lernenden muss durch Instruktionen in Form von Fehlermeldungen, Rückmeldungen, Kommunikation und Hilfestellungen kontrolliert werden, um den Lernprozess zu gewährleisten. Hilfestellungen passiver Art können in das System integriert werden, um den Lernenden mit dem Thema nicht „alleine zu lassen“, sie müssen benutzerfreundlich gestaltet werden und sich auf den aktuellen Lerngegenstand beziehen. Bei der Gestaltung von Fehler- oder auch Rückmeldungen muss in der ersten Linie ihre de-/motivierende Wirkung beachtet werden. Die Motivation des Lernenden ist Voraussetzung, damit ein Lernprozess überhaupt stattfinden kann, daher sollten die Fehler- bzw. Rückmeldungen im Falle eines Fehlers so weit wie möglich wertungsfrei formuliert sein.

Der Lernende muss in der Lage sein, seinen Lernfortschritt für sich festzuhalten. Dies kann vom System unterstützt werden, indem z.B. die für die Bearbeitung von einer Problemstellung benötigte Zeit und der Schwierigkeitsgrad der Aufgabe vom System protokolliert und angezeigt werden. Dem Lernenden muss auch eine Möglichkeit angeboten werden, seinen derzeitigen Ort im Lernprozess zu finden.

Weiterhin fördert eine sinnvolle Kombination multimedialer Gestaltungselementen, Texten und interaktiver Elemente bei der Darstellung des Lerngegenstandes die Aufmerksamkeit und das Interesse des Lernenden, die sich positiv auf die Konzentration auswirken. Animierte Graphiken können beispielsweise zur visuellen Darstellung eines Lerninhalts eingesetzt werden, jedoch wäre es nicht sinnvoll sie als reine Dekoration zu verwenden, die den Lernenden vom Lernstoff ablenken.

Eine gut strukturierte und übersichtliche Oberfläche ist eine weitere Voraussetzung zur Steigerung der Motivation und des Interesses.

CATS ist ein Web-basiertes System, das Fernstudenten ermöglicht, sich zu jeder Zeit und an jedem Ort unterschiedliche Lerninhalte anzueignen. Auf der Basis des Proficiency-Levels (einer Größe, die zeigt, Aufgaben welchen Schwierigkeitsniveaus der Anwender imstande ist zu lösen) wird unter den Lernenden ein Ranking ermittelt und dem jeweiligen Lernenden seine Position im Ranking mitgeteilt. Die Studenten werden automatisch in eine Videokonferenz mit anderen Studenten, die sich alle auf dem gleichen Level eines Thema befinden, eingebunden. Der Proficiency-Level gibt auf der anderen Seite dem Lehrer die Möglichkeit, sich über den Lernfortschritt seiner Studenten zu informieren.

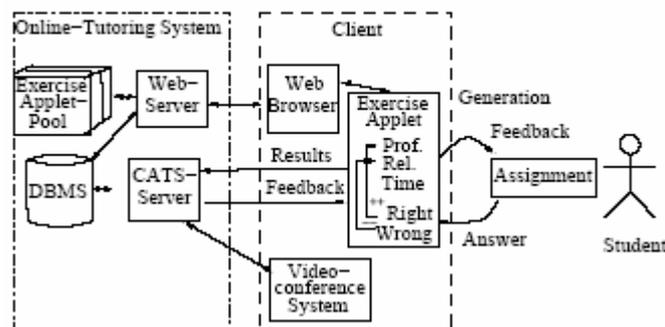


Abb. 12: CATS-Architektur (Quelle: Christian Liebig, Wolfgang Effelsberg. *Seamless Integration of Exercise Problems to the Proficiency Level of the Learner*, 2003)

Wie in der Abbildung 12 dargestellt, ist ein Übungsapplet ein Teil der Client-seitigen Komponente in der Client-Server-Architektur von CATS. Drei Parameter:

- proficiency (Schwierigkeitsgrad der Aufgabe)
- reliability (Zuverlässigkeit der Lösung des Lernenden)
- time (benötigte Zeit für die Aufgabe [msec])

werden vom Applet über ein Interface an den Server übermittelt, auf dessen Basis das Ranking erstellt wird. Der Schwierigkeitsgrad der Aufgaben wird an den Lernerfolg des Anwenders angepasst.

Mit Hilfe des Applets TokenRingTrainer kann das in der Vorlesung Rechnernetze erworbene Wissen zum Thema Token Ring vertieft bzw. ergänzt werden.

### **3.2 Aufgabentypen**

Allen Aufgaben liegt das Normal-Token-Release-Verfahren zugrunde, das Early-Token-Release-Verfahren wird nicht behandelt. Weiterhin wird davon ausgegangen, dass immer nur ein Paket von einer Station gesendet werden kann, einzelne Pakete entsprechen dabei der 10ms-Paketgröße. Der Spezifikation entsprechend können Stationen für die Übertragung nur die 4 ersten Prioritätsstufen nutzen, d.h. die maximal mögliche Priorität liegt bei 3. Es wird auch angenommen, dass alle Stationen unterschiedliche MAC-Adressen besitzen. Eine weitere Annahme bzgl. der Datenübertragung ist, dass das Empfangen und Kopieren der Daten durch die Zielstation nicht fehlschlägt.

Es wird generell zwischen zwei Aufgabentypen unterschieden: Theorieaufgaben und Rechenaufgaben. Die Theorieaufgaben sind darauf ausgerichtet, das Verständnis für die genaue Funktionsweise der im Token Ring – Netz eingesetzten Protokolle zu fördern. Nach jeder Theorieaufgabe wird eine Rechenaufgabe gestellt, die auf das theoretische Wissen der vorherigen Aufgabe aufbaut.

Es werden insgesamt 3 Themengebiete behandelt: Grundprinzip des Token Passing Steuerungsverfahrens, Monitoring Mechanismus und Priorisierungsmechanismus.

Das Verständnis des Grundprinzips ist eine Voraussetzung für das Verständnis der anderen beiden Mechanismen. Deswegen findet der Einstieg über dieses Themengebiet statt, wobei insgesamt 5 Aufgaben (2 Theorie- und 3 Rechenaufgaben) dazu gestellt werden.

Aufgaben zum Monitoring Mechanismus im Token Ring werden als weniger schwierig eingestuft, als Aufgaben zum Priorisierungsmechanismus. Diese Einstufung resultiert vor allem daraus, dass von den 3 Ringwartungsprozessen: Ring Purge, Claim Token und Beaconsing, nur der erste Prozess (Ring Purge) und somit auch nur die Funktionen des aktiven Monitors behandelt werden. Es würde den Rahmen dieser Studienarbeit sprengen, alle Details des Ringwartungsmechanismus einzubeziehen. Auch zu diesem Themengebiet werden insgesamt 5 Aufgaben (2 Theorie- und 3 Rechenaufgaben) gestellt.

Die nächsten vier Aufgaben beziehen sich auf den Priorisierungsmechanismus, der hier als schwierigstes Thema eingestuft wird. Alle darauffolgenden Aufgaben stellen eine Kombination aus beiden letzten Mechanismen dar.

Wird eine Aufgabe durch den Anwender nicht richtig gelöst, erzeugt das Programm eine ähnliche Aufgabe mit dem gleichen Schwierigkeitsgrad. Nur bei einer richtigen Lösung wird der Schwierigkeitsgrad erhöht.

### 3.2.1 Theorieaufgaben

Alle Theorieaufgaben entsprechen dem gleichen Muster: der Anwender wird aufgefordert, die einzelnen Schritte eines vorgeschlagenes Szenarios zu beschreiben. Bei den Theorieaufgaben des ersten Themengebiets: Token Passing Steuerungsverfahren, ist eine sehr detaillierte Beschreibung der Vorgänge erforderlich. Denn dieses Verfahren den Kern der Token Ring Technologie darstellt. Detailliert bedeutet an dieser Stelle, dass die Aktionen aller Stationen im Ring erwähnt werden müssen, auch z.B. das unveränderte Weiterleiten des Rahmens bzw. des Tokens durch eine Station.

Für die anderen Aufgaben (Monitoring und Priorisierung) sind nur die durch eine Station am Rahmen bzw. Token vorgenommenen Veränderungen von Interesse: z.B. das Setzen der Reservierungsbits oder des Monitorbit, aber auch das Setzen der AC-Bits. Nimmt eine Station mehrere Veränderungen gleichzeitig am Rahmen bzw. Token vor z.B. die gerade genannten Veränderungen, ist die richtige Reihenfolge der Aktionen von Bedeutung, denn das setzt unter anderem die Kenntnisse über Rahmen- bzw. Tokenformate voraus. Das Nichteinhalten der richtigen Reihenfolge wird in für den weiteren Ablauf des Szenarios "unkritischen" Fällen (z.B. der aktive Monitor setzt das Monitorbit und nimmt gleichzeitig eine Reservierung vor) toleriert bzw. als richtig bewertet.

Das Lösen der Aufgaben wird dem Anwender dadurch erleichtert, dass ihm schon vordefinierte mögliche Aktionen, die einzelne Stationen durchführen können, zur Auswahl gestellt werden. Er muss nur den Namen der entsprechenden Station mit einer Aktion kombinieren und in die Lösungsliste eintragen. Hat der Anwender seine Lösung der Aufgabe vollständig in die Liste eingetragen, dann kann er sie auf Richtigkeit überprüfen. Dabei werden die richtigen Einträge grün und die falschen rot markiert und die fehlenden Einträge werden hinzugefügt und schwarz angezeigt. Falls gewünscht kann die Lösung in Form einer Animation angezeigt werden. Parallel zur Animation, in der vor allem die Veränderungen des Rahmentyps dargestellt werden, werden die Aktionen, die einzelne Stationen durchführen, aufgelistet: die Animation wird sozusagen kommentiert. Dadurch hat der Anwender die Möglichkeit, den Ablauf des Szenarios besser nachzuvollziehen.

Es wird unter folgenden vier Rahmen- oder auch Sequenztypen unterschieden:

- T normales Token
- P Prioritätstoken
- F Datenrahmen
- R Ring Purge Frame.

### 3.2.2 Rechenaufgaben

Außer den ersten beiden liegt allen Rechenaufgaben auch ein gleiches Muster zugrunde: zu einem vorgeschlagenen Szenario soll die Zeit, in der dieses abgeschlossen wird, berechnet werden. Für die Lösung dieser Art von Aufgaben wird einerseits das Verständnis des theoretischen Hintergrunds erforderlich und andererseits die Fähigkeit, mathematische Aufgaben zu berechnen, gefördert.

Der Anwender kann sich die Lösung der Aufgabe auf zwei Arten anzeigen lassen, nachdem er seinen Wert eingegeben hat: als einen Wert in msec oder als mit der Animation kombinierte Zeitberechnung. Dabei werden jeweils die Zeitpunkte, zu denen Stationen Rahmen empfangen bzw. senden (jedoch nicht weiterleiten, da die 1-Bit-Verzögerung vernachlässigt werden kann), angezeigt.

Die ersten beiden Rechenaufgaben sind weniger auf bestimmte Szenarien ausgerichtet, sondern haben einen eher allgemeinen Charakter: bei der ersten Aufgabe soll anhand von zwei Parametern (Token Holding Time und Übertragungsgeschwindigkeit) die maximal mögliche Rahmengröße berechnet werden. Die Fragestellung der zweiten Aufgabe bezieht sich auf die zufällig generierte Zusammensetzung des Netzwerks, nämlich welche Speicherkapazität dieses Netzwerk hat. Mit anderen Worten: wie viele Bits passen gleichzeitig auf den Ring, wenn das Netzwerk eine bestimmte Anzahl von Stationen hat und wenn Verbindungen zwischen Stationen eine bestimmte Länge haben?

Das für die Eingabe des Ergebnisses vorgesehene Feld wird mit dem erwarteten Parameter (z.B. Frame size) in der erwarteten Einheit (z.B. Bits) gekennzeichnet. Die Eingabe kann auf die Richtigkeit geprüft werden und im Falle einer nicht korrekten Lösung kann auf Wunsch des Anwenders die richtige Lösung angezeigt werden.

Kommt der Anwender mit der Lösung der Aufgaben nicht zurecht, wird ihm in der Online-Hilfe die genaue Formel für die Berechnung der jeweiligen Größen präsentiert.

### 3.3 GUI

Wenn das TokenRingTrainer-Applet gestartet wird, erfolgt als erstes die Anmeldung mit dem Benutzernamen und Kennwort in einem Dialogfenster.

Sowohl für die Theorie als auch für Rechenaufgaben wird die gleiche Benutzeroberfläche verwendet: Abbildung 13. Mit dem Unterschied, dass je nach Aufgabe unterschiedliche Eingabefelder aktiviert werden, um dem Anwender das Bedienen der Benutzeroberfläche zu erleichtern.

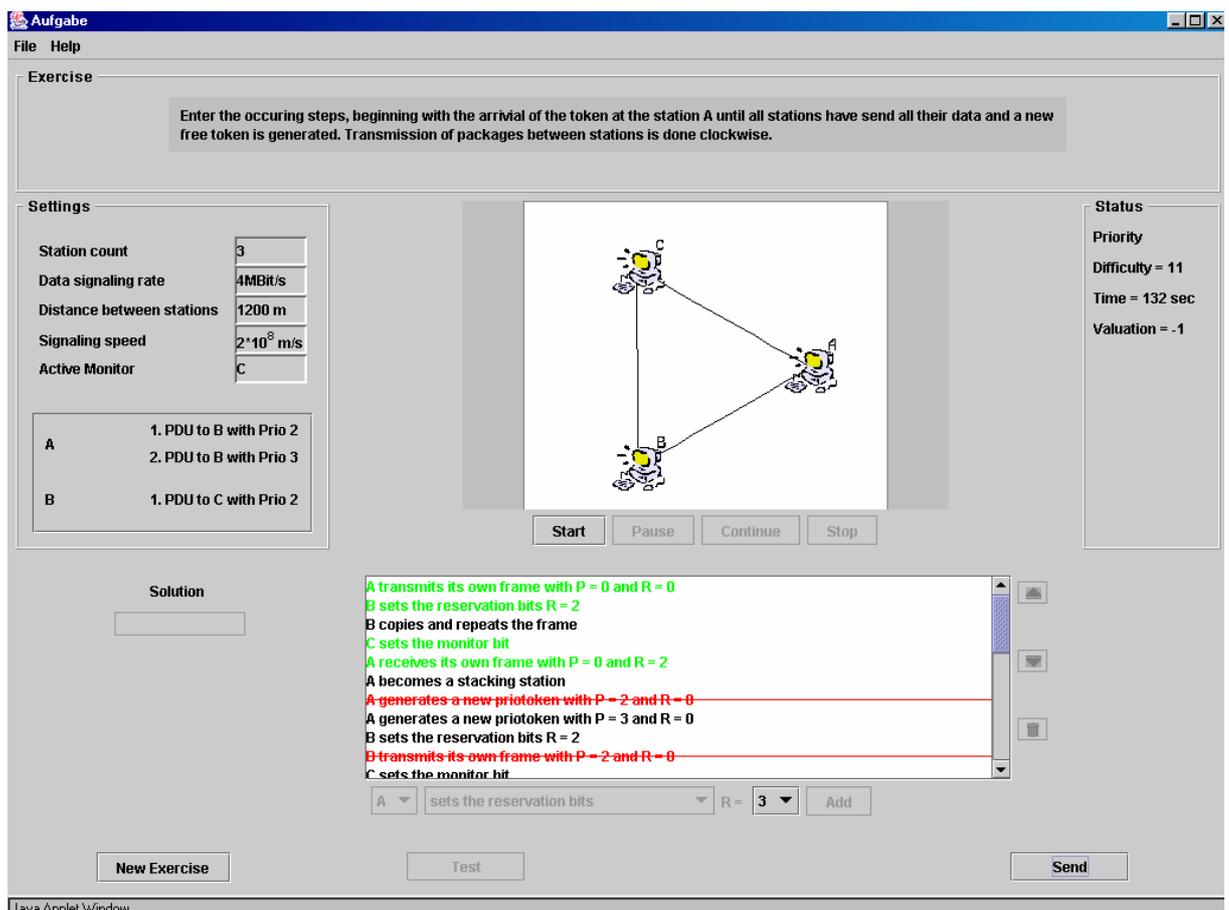


Abb. 13: TokenRingTrainer Benutzeroberfläche

Beim Betätigen des Knopfs „New Exercise“ wird eine neue Aufgabenstellung generiert. Dieser Knopf wird aktiviert, wenn das Applet neu gestartet wurde oder wenn die Lösung der vorhergehenden Aufgabe bereits auf Richtigkeit getestet wurde.

Im Settings-Bereich werden allgemeine Daten über das Netzwerk angezeigt, die in Kombination mit der Aufgabenstellung aus dem „Exercise“- Bereich und der graphischen Darstellung alle notwendigen Informationen für das Lösen von Aufgaben liefern.

In dem unteren Bereich der Oberfläche kann die jeweilige Lösung eingegeben werden.

Korrektur der Eingabe bei Theorieaufgaben kann mit Hilfe der Knöpfe (moveUp, moveDown, delete) vorgenommen werden. Der per Mausklick markierte Eintrag kann gelöscht werden oder mit dem nächst Höheren bzw. nächst Tieferen vertauscht werden.

Zu besseren Orientierung werden in dem als Statusbereich gekennzeichneten Feld (1) das der aktuellen Aufgabe zugrundeliegende Thema und (2) der aktuelle Schwierigkeitsgrad der Aufgabe angezeigt. Nach dem Betätigen des „Test“-Knopfs wird noch (3) die Anzeige des Zeitverbrauchs für das Lösen der Aufgabe und (4) die Zuverlässigkeit der eingegebenen Lösung hinzugefügt.

### **3.4 Implementation**

Beim Erzeugen jeder neuen Aufgabe wird ein virtuelles Netzwerk aufgebaut. Dieses Netzwerk besteht aus einer Anzahl von Stationen, Verbindungen zwischen Stationen und einem Rahmen und hat zwei Attribute Übertragungsgeschwindigkeit und Ausbreitungsgeschwindigkeit. Die Übertragungsgeschwindigkeit variiert je nach Aufgabenschwierigkeit zwischen drei Werten: 4, 10 und 16 Mbit/s. Ausbreitungsgeschwindigkeit bleibt immer konstant:  $2 * 10^8$  m/s.

Jede Station besitzt einen Namen, eine Vorgänger- und eine Nachfolgerstation, zwei Verbindungen, jeweils eine zum Vorgänger und zum Nachfolger, eine Queue mit den zu versendenden PDUs, boolean Attribute: aktiver Monitor und Stacking Station und die für die Verwaltung von Prioritäten benötigten Stacks und Register [1]. Verbindungen besitzen eine Referenz auf ihre Anfangs- und Endstation und haben Länge in [m] als Attribut.

Beim Generieren des neuen Netzwerks wird per Zufall entschieden, aus wie vielen Stationen das Netz besteht, wie viele von diesen Station Daten zu versenden haben, an wen diese Daten adressiert sind und mit welcher Priorität diese versendet werden müssen, welche Station die Rolle des aktiven Monitors übernimmt, wie lang die Verbindungen zwischen Stationen sind und die Position des Tokens zu Beginn des Szenarios. Die zufällige Generierung wird durch den Schwierigkeitsgrad der Aufgabe eingeschränkt. Mit dem steigenden Schwierigkeitsgrad werden die entsprechenden Parameter erhöht, allerdings werden bei den ersten Aufgaben eines neuen Themengebiets die Anzahl der Stationen im Ring allgemein, die Anzahl der sendewilligen Stationen und die Anzahl der zu versenden PDUs so klein wie möglich gehalten, da die Steigerung der Schwierigkeit der Aufgabe durch das neue Thema schon stattfindet. Die Anzahl der Stationen im Ring variiert zwischen 3 und maximal 12 Stationen. Ein Netzwerk

aus mehr als 12 Stationen ist bei der gewählten Größe der graphischen Darstellung wegen der Unübersichtlichkeit nicht sinnvoll.

Das erstellte Szenario wird direkt nach dem Erstellen der Aufgabe gesteuert durch eine zentrale Instanz (TokenRingManager) durchgespielt. Dabei werden zum einen die Aktionen, die Stationen beim Empfang eines Datenpakets oder Tokens vornehmen, festgehalten und zum anderen die jeweiligen Zeitpunkte, wann die Aktionen stattfinden, sowie die Gesamtdauer des Ablaufs berechnet. Das Netzwerk besitzt einen Rahmen, der von Station zur Station weitergereicht wird. Der Rahmen enthält folgende Informationen: Prioritätsbits, Reservierungsbits, Monitor- und Tokenbit und eine PDU, falls es sich beim Rahmen um einen Datenrahmen handelt. Anhand dieser Informationen wissen die Stationen, wie sie auf den Empfang des Rahmen reagieren müssen.

Für die Darstellung der Animation wird die im Rahmen der Diplomarbeit von Ralf Gitzel an der Universität Mannheim entwickelte Animationsbibliothek benutzt. Diese Bibliothek wurde in der ersten Linie für Netzwerksimulationen im Kontext des multimedia-basiertes Lernen entwickelt. Für ausführliche Information zu der Bibliothek sei an dieser Stelle auf die Diplomarbeit von Ralf Gitzel [8] verwiesen.

Die graphische Darstellung setzt auf das generierte Netzwerk auf. Die paketverarbeitende Logik wurde komplett von der graphischen Schicht getrennt. Der Vorteil dieser Delegation besteht darin, dass das zufällig generierte Szenario im Hintergrund durchgespielt werden kann/muss, um die Eingaben des Anwenders auf Richtigkeit zu prüfen, bevor die Animation gestartet wird.

Eine Animation besteht aus statischen und dynamischen Elementen: Netzwerkknoten (Router), Verbindungen respektive Paketen. Wie ein Netzwerkknoten auf den Empfang eines Pakets zu reagieren hat, wird über einen PackageHandler reguliert. Jeder Knoten besitzt seinen eigenen PackageHandler. Die für dieses Applet spezifische Implementierung des PackageHandlers (RahmenHandler) leitet den Impuls, dass ein Rahmen empfangen wurde, an die entsprechende Station weiter:

Router → RahmenHandler → Station

Diese verarbeitet den empfangenen Rahmen entsprechend dem Token Ring Protokoll und liefert das Ergebnis zurück an den RahmenHandler. Der RahmenHandler entscheidet nun wie der verarbeitete Rahmen bzw. Token darzustellen ist und sendet das neue Paket los, indem er die entsprechenden Methoden des Routers aufruft.

Router ← RahmenHandler ← Station

Die Animation wird somit dezentral durch die RahmenHandler reguliert. Auf diese Weise kann die Animation unendlich weiter fortgesetzt werden. Da aber durch die Aufgabenstellung

sie an einer bestimmten Stelle: nämlich wenn alle Stationen ihre Daten erfolgreich versendet haben und ein freies Token wieder generiert wird, stoppen muss, bedarf es einer zentralen Instanz, die den Fortschritt kontrolliert und zum richtigen Zeitpunkt die Animation anhält. Dies wird dadurch gelöst, dass alle Stationen nach jedem Empfang eines Tokens bzw. eines Rahmens ihren eigenen Fortschritt in eine zentrale Liste eintragen. Jeder RahmenHandler geht diese Liste durch, bevor er ein Paket weiter versendet, und überprüft, ob alle Stationen sich als „fertig mit der Übertragung“ gemeldet haben. Ist dies der Fall, stoppt er die Animation.

Für die Eingabe der Lösung einer Theorieaufgabe wird JList verwendet, zum dynamischen Hinzufügen und Anzeigen der einzelnen Einträge werden dabei ListModel und ListCellRenderer eingesetzt [9], wobei die Eingabe anfangs in schwarzer Schrift dargestellt wird. Nach der Korrektur wird die Liste aktualisiert: die fehlenden Einträge werden hinzugefügt, die richtigen grün und die falschen rot markiert. Dabei wird dem ListCellRenderer ein Vector übergeben, der Informationen darüber enthält, wie die einzelnen Einträge darzustellen sind.

## 4 Zusammenfassung

Mit dem im Rahmen dieser Studienarbeit entwickelten Übungsapplet TokenRingTrainer wird Studenten die Möglichkeit gegeben, das in der Vorlesung Rechnernetze erworbene Wissen zu den Konzepten der Zugriffskontrolle in Token Ring Netzen anhand von Übungsaufgaben zu vertiefen. Das Applet stellt kein autonomes Tool dar, sondern soll in das Framework CATS integriert werden.

Da mit dem Applet nicht alle Aspekte der Zugriffskontrolle abgedeckt wurden, sind weitere Anwendungen denkbar, die z.B. das Early-Token-Release-Verfahren implementieren oder auch Monitoring-Verfahren wie Claim-Token- und Beaconing-Prozesse einbeziehen. Es ist weiterhin ein breiteres Spektrum an Aufgabenstellungen möglich, wie z.B. Fragen nach der Effizienz des Netzwerks in bestimmten Situationen etc. Auch in der Animationsdarstellung sind Erweiterungen bis auf die bit-genaue Darstellung der Datenübertragung denkbar.

Das Applet soll die Studierenden dabei unterstützen, sich die Lerninhalte selbst anzueignen.

Das Lernen an sich ist ein individueller und aktiver Prozess. Deswegen darf nicht vergessen werden, dass diese Anwendung nur eine „Lernhilfe“ darstellt und dem Lernenden nicht die Mühe abnehmen kann, selbst zu lernen.

## Literaturverzeichnis

- [1] ANSI/IEEE Std. 802.5, 1998 Edition: *Token Ring access method and Physical Layer specifications*.
- [2] Hans-Georg Göhring, Franz-Joachim Kauffels. *Token Ring: Grundlagen, Strategien, Perspektiven*, Bergheim 1990.
- [3] Abrahams, John R. *Token ring networks design implementation and management*. NCC Blackwell Limited, 1991.
- [4] Andrew S. Tanenbaum. *Computernetzwerke*. Autorisierte Übersetzung der amerikanischen Originalausgabe: *Computer Networks*, 3<sup>rd</sup> edition by Andrew S. Tanenbaum. Pearson Studium- Verlag, 2000.
- [5] Ralf Steinmetz. *Multimedia-Technologie: Grundlagen, Komponenten und Systeme*. 3., überarb Aufl. Berlin; Heidelberg; New York Springer- Verlag, 2000.
- [6] Horst Dichanz und Annette Ernst. *E-Learning: Begriffliche, psychologische und didaktische Überlegungen zum „electronic learning“*. Juni 2001. [http://www.medienpaed.com/00-2/dichanz\\_ernst1.pdf](http://www.medienpaed.com/00-2/dichanz_ernst1.pdf). Letzter Besuch der Seite: Januar 2004.
- [7] Christian Liebig, Wolfgang Effelsberg. *Seamless Integration of Exercise Problems to the Proficiency Level of the Learner*. Proceedings of the IEEE International Workshop on Multimedia Technologies in E-Learning and Collaboration (WOMTEC 2003), Nice, France, October 2003.
- [8] Ralf Gitzel. *Diplomarbeit: Animationsbibliothek mit RMI-Teachware als Beispielanimation*. Mannheim, Dezember 2000.
- [9] Martin Schader, Lars Schmidt-Thieme. *Java<sup>TM</sup>: Eine Einführung*. – Berlin; Heidelberg; New York Springer- Verlag. 2., aktualisierte und verbesserte Auflage, 1999.