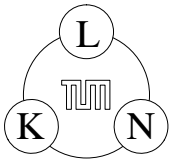
TECHNISCHE UNIVERSITÄT MÜNCHEN
Fakultät für Elektrotechnik und Informationstechnik
Institut für Informationstechnik

A Flexible Set of Methods for the Analysis and Optimization of CRMA–II Cycle Time and Scheduling Mechanisms

Andreas Kirstädter

TUM–LKN–TR–9401



Lehrstuhl für Kommunikationsnetze
Prof. Dr.–Ing. J. Eberspächer
Arcistraße 21
80290 München

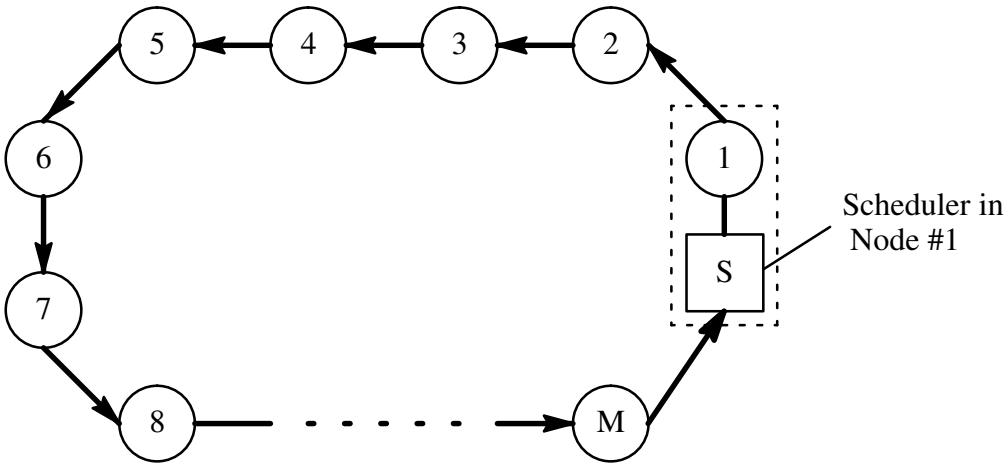
Juni 1994

1 Summary:

In a lot of previous papers the excellent throughput and fairness characteristics of the CRMA–II (Cyclic Reservation Multiple Access) Gbit/s LAN and MAN Medium Access Control protocol have been shown by extensive simulations (e.g. [v.As 92], [Mar 92], [Bac 93], and [Meu 93]). Also new mechanisms and features have been proposed to further enhance the performance of CRMA–II ([Meu 94], [Geh 94]).
The purpose of this paper now is first to give new approximate analytical expressions for the distributions of the internal system variables of CRMA–II. The derived methods are then used to detect and correct two sources of possible deadlocks in the protocol itself and in a proposed enhancement. Since the delay–oriented aspects of how fairness is achieved within CRMA–II networks are mostly determined by the structure of the scheduling cycles and their duration T_{cycle} an approximation for the distribution of T_{cycle} is derived by analyzing its constituting processes and effects. As a result guidelines are given concerning the necessary calculation speed of the CRMA–II scheduler and an adaptive marking method is proposed to further optimize the scheduling process.

2 CRMA–II

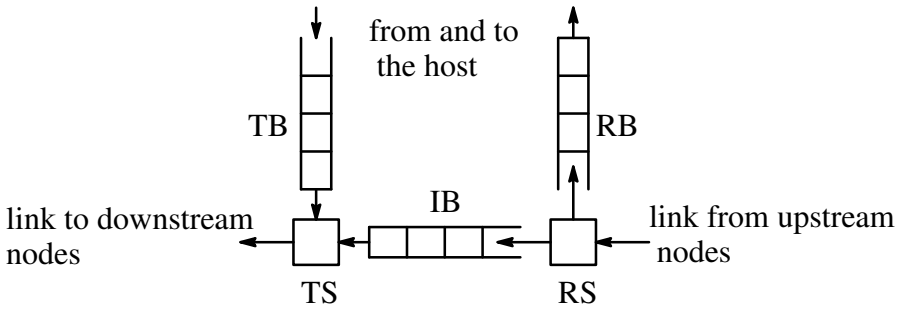
Based on slotted Buffer Insertion (BI) as the main access principle allowing concurrent access CRMA–II uses a superimposed cyclic reservation mechanism to correct the inherent unfairness of the applied destination removal technique. A CRMA–II network consists of a number of M BI nodes connected by unidirectional links to a single or dual ring topology. A scheduler active in one of the BI nodes manages the reservation process:



In this article mostly single ring topologies are considered; the results also apply to dual rings since each ring handles the reservation and scheduling on its own.

2.1 Slotted Buffer Insertion

Each BI node generally consists of an insertion buffer (IB), a transmit buffer (TB), a receive buffer (RB), a receive switch (RS), and a transmit switch (TS):



CRMA–II uses a slotted transmission structure that permits concurrent access by different nodes. The slot size is chosen to support cell based ATM traffic. A LAN packet therefore consists of one or more slots transmitted contiguously (due to the buffer insertion mechanism). Packets with a destination address matching the actual node address are switched by the RS into the RB and thus removed from the ring. This destination removal leads to an increase (up to 100%) of the maximum throughput of the network due to spatial re–use of transmission capacity.
Slots from packets that are not addressed to the actual node are appended to the IB. A node can only start a transmission of a own packet if its IB is empty; otherwise it uses the transmission capacity of the outgoing link to service its IB (ring priority). But a node currently transmitting a packet out of its TB continues to do so even if arriving packets from upstream nodes refill its IB during transmission. Before transmitting the next packet from the TB the node has again first to empty its IB by discarding arriving free slots.

2.2 Access Mechanisms in CRMA–II

In addition to the above mentioned buffer insertion mechanisms CRMA–II uses a cyclic reservation process to guarantee each node a throughput oriented fairness:

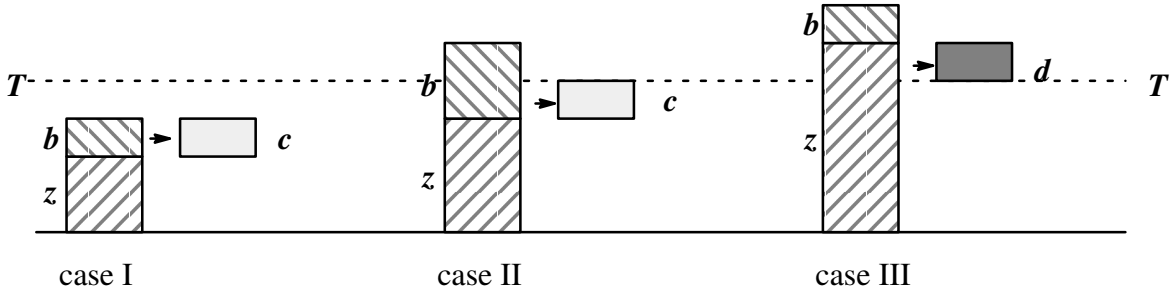
A CRMA–II slot carries two flags:
– a FREE / BUSY flag indicating whether the slot is currently transporting parts of a packet and
– a GRATIS / RESERVED flag set by the scheduler in order to control fairness.

Therefore four types of CRMA–II slots can be distinguished:
FG (free, gratis): empty slot for normal BI access;
FR (free, reserved): empty slot for reservation controlled access (marked by the scheduler);
BG (busy, gratis): filled slot after BI access;
BR (busy, reserved): a previously filled slot marked by the scheduler;

The scheduler starts a reservation cycle by issuing a special reservation command frame. Each node informs the scheduler about its current fairness situation by inserting into the reservation frame the value of its transmit counter z corresponding to the number of TB slots transmitted since the passing of the last reservation frame and the number of reserved slots b necessary to empty both IB and TB of the node.
Upon reception of the reservation frame the scheduler compares the values b and z of each node with a threshold T in order to compute the reservations c and deferments d for each node to maintain the throughput fairness. During these fairness calculations full network utilization is maintained. The threshold T is either a fixed value set by the network management or is made dynamically adaptive to the current load situation within the network [Lem 93].

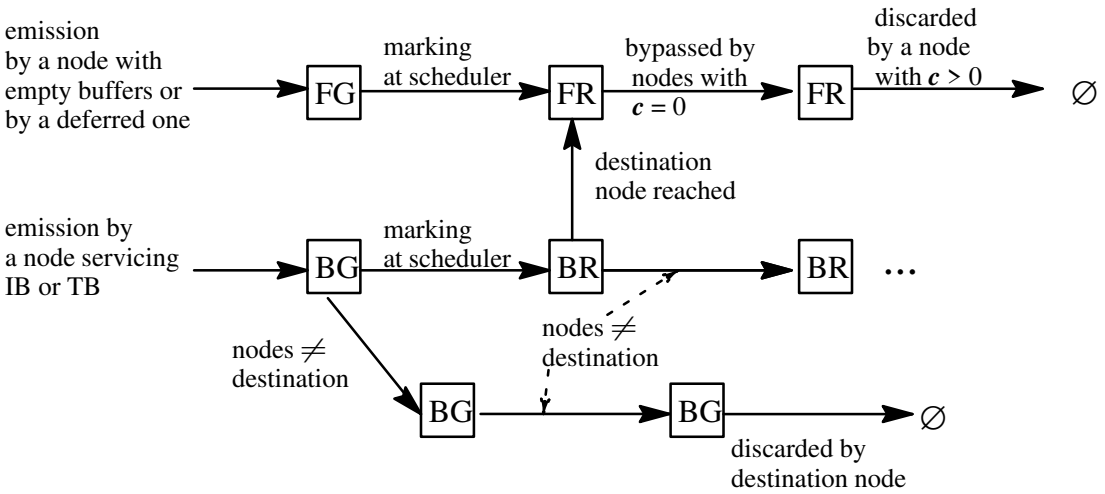
In the case of a fixed T the scheduler may use the following scheduling algorithm [v. As 92]:

– nodes with a sum of b and z below T are granted $c = b$ reserved slots (case I in the figure below),
– nodes with this sum exceeding T, but with z below T are granted $c = T–z$ reserved slots and
– nodes with z exceeding T must defer a possible access $d = z–T$ times for the duration of a slot transmission time T_{slot} (case III).



The scheduler then communicates the value of T around the ring using a confirm command frame. A node receiving this frame calculates its own values of c and d from T, updates its confirm and deferment counters, and decreases its transmission counter z by the value T.
Immediately after emitting the confirm frame the scheduler starts marking a number of gratis slots (FG and BG) as reserved that corresponds to the sum of confirms of all nodes. A FR slot may then only be discarded by a node holding confirms: $c > 0$ (decreasing its confirm counter by one).

On its way around the ring a slot can change its state according to the following rules:



In order to decrease the load of the IBs (and thus the transmission delays) CRMA–II uses the following bypassing techniques:

- Nodes with confirm counters c=0 bypass arriving FR slots on frame boundaries and do not enqueue them into their IBs;
- Reserve and confirm command frames are bypassed on slot boundaries to ensure minimal round trip delays.

For maintaining the integrity of each reservation cycle the scheduler sets a special End–Of–Cycle flag in the header of the first marked slot. A node receiving this EOC signal is only allowed to re-

transmit it if its confirm counter is zero (i.e. this node had used up all its reserved slots of the current cycle). The scheduler then only starts a new cycle (by emitting a reserve command) when it has received the EOC signal.

2.3 Outline of the Rest of This Paper

The next chapter contains the analytical methods needed to describe the main system parameters of a CRMA–II ring network. Since for an exact analysis of the fairness mechanisms mean values of the system parameters are not sufficient the probability distribution functions (pdf) of the following internal state variables are derived:

- waiting time $\boldsymbol{w}$ of busy slots in the IB of a node,
- number $\boldsymbol{b}$ of slots currently in the IB and TB of a node,
- number $\boldsymbol{z}$ of slots transmitted out of the TB of a node per cycle,
- and the number of confirms $\boldsymbol{c}$ and deferments $\boldsymbol{d}$ per node and cycle.

In chapter 4 these methods are then combined in order to:

- detect and eliminate an incorrectness and a possible deadlock within the CRMA–II protocol concerning the manipulation of the transmit counters within the nodes,
- to explain the instability of a proposed adaptive scheduling technique [Lem 93],
- describe analytically the scheduling mechanism as a whole leading to the pdf of the duration $\boldsymbol{T_{cycle}}$ of the scheduling cycles (the random variable $\boldsymbol{T_{cycle}}$ reflects the delay oriented aspects of the quality of the fairness that CRMA–II offers),
- discuss the influence of the duration of the scheduler calculations on the performance of the fairness mechanisms within CRMA–II,
- and to derive and introduce some slight protocol modifications for minimizing the worst case access delay a node experiences.

3 Analytical Methods for Describing CRMA–II Behaviour

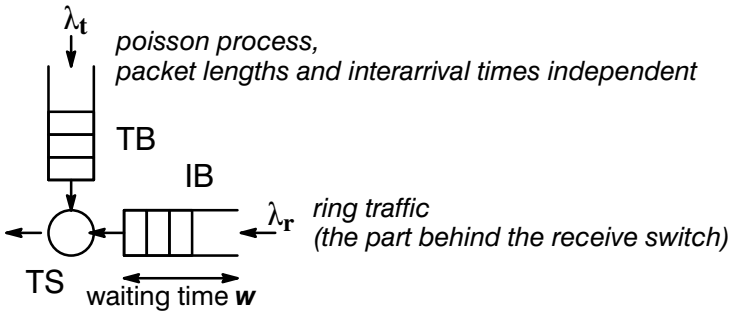
3.1 Assumptions

At this point a summary of the assumptions used later on is given:

The traffic from the hosts is assumed to be Poisson with arrival rate λ_t, a mean packet length of $\overline{x}$ slots, and equally distributed destination adresses. Unless otherwise noted the considered networks contain $M = 20$ nodes equally distributed on a single ring of $L = 2$ to 50 km. The mean packet length is assumed to be $\overline{x} = 19$ ATM–sized slots of 53 bytes each. The usable transmission bit rate on the links is $R = 1.96$ Gbit/s and the calculation time of the scheduler $T_{calc} = 20$ microseconds.
All time variables (unless otherwise denoted) are normalized to the transmission time of a slot T_{slot} and the loads ϱ_t of a single node resp. S of the whole system are normalized to the link bitrate.
For tractability the waiting times in the different insertion buffers are assumed to be independent of each other while the maximum buffer lengths are assumed to be infinite. The reduction of the transmission capacity of the network by reservations and deferments is neglected as are the processing times within the nodes that normally would delay the retransmission of slots.
As CRMA–II networks at bit rates in the Gbit/s range are considered the duration of a slot T_{slot} is smaller than the propagation delay between two adjacent nodes.
During the analysis the scheduling threshold T is considered a fixed value.
As opposed to the protocol in [v.As 92] the manipulation of the transmit counters within the nodes has been changed:
instead of decrementing the transmit counters by the threshold value T at the passing of the confirm frame they are reset to zero each time after their current value has been inserted into the reserve command frame. Decrementing the counters only by T would lead to counter over– or underflows as will be shown in chapter 4.1.
A list of symbols, abbreviations and variables is given in the appendix.

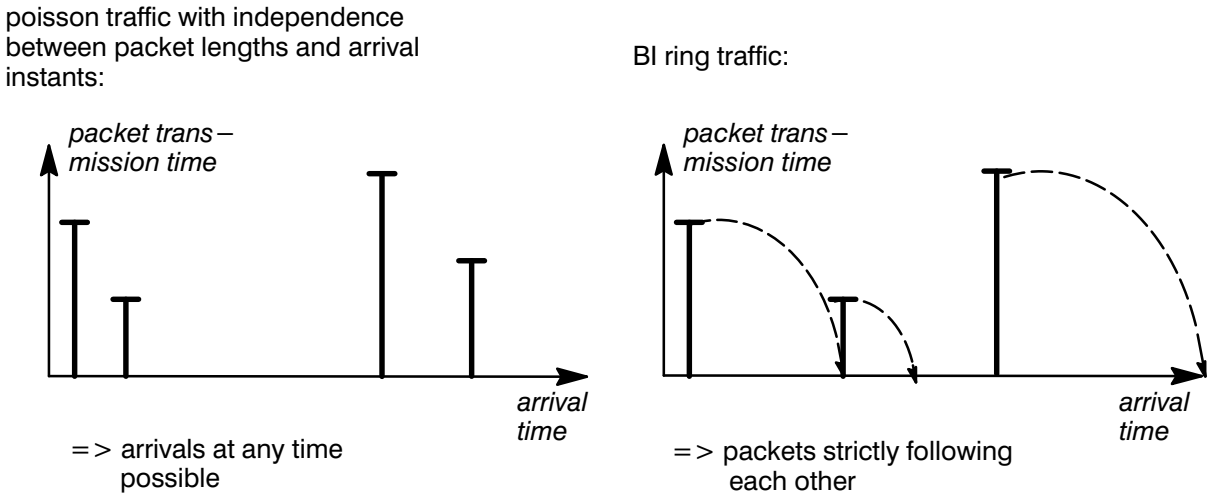
3.2 Distribution of the Waiting Time *w* of a Busy Slot in the IB of a Node

Considering a single CRMA–II station we get the following simplified structure after removing the receive switch and the receive buffer:



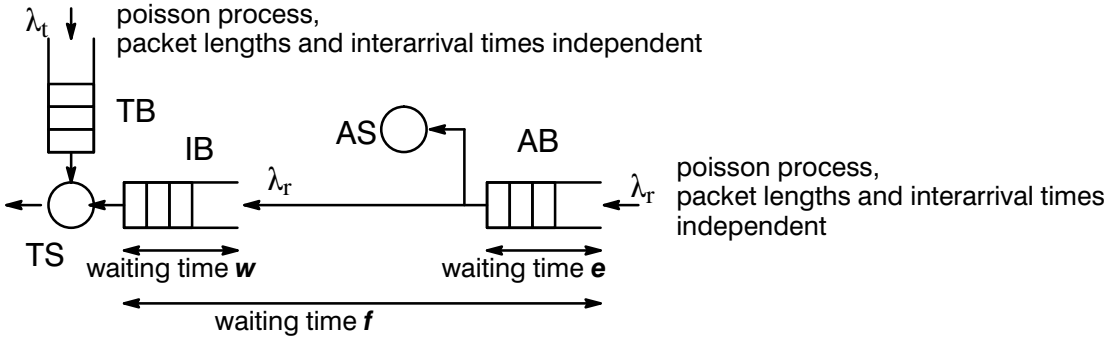
Neglecting the slotted transmission structure of CRMA–II (the mean packet length being much greater than the length of a slot) an approximate expression for the pdf of the delay ***w*** experienced by packets in the IB of an isolated BI node can be given by applying the techniques of "delay cycle analysis" [Kl II 76, p. 110] to a head–of–the–line nonpreemptive priority queueing system [Ham 86, p.262] with the following arrival rates:
λ_t : lower priority, TB packets
λ_r : higher priority, IB packets (BI with ring priority)

But due to the mechanisms of BI and the serial transmission of data on the ring the arrival times and packet lengths at the input of the IB cannot be treated as independent poisson processes (especially at high loads and a large number of nodes):



In order to handle such a Buffer Insertion system analytically we here use a method extending that of Bux and Schlatter [Bux 83] (leading only to the mean value of ***w***) to the determination of the pdf of ***w***:

An "imaginary" additional buffer (AB) with an assumed second server (AS) is added to the original two buffer / one server system of a BI node in order to generate the typical BI traffic at the input of the IB:



The kind of connection between IB and AB / AS symbolizes that a packet arrives at IB at the same moment when AS starts servicing it out of AB.
At the input of AB a poisson process is assumed with an arrival rate of λ_r so that at the input of IB we get a traffic that resembles the typical BI traffic.
With the necessary independence in the arrival processes at both inputs the whole nonpreemptive head–of–the–line queueing system can now be described by the delay cycle analysis:

For the Laplace–transform of the pdf of ***f*** (the sum of the waiting times in IB and AB) we get:

$$F^*_f(s) = \frac{(1-\varrho)\ s + \lambda_t\ (1 - B^*(s))}{s - \lambda_r + \lambda_r B^*(s)}$$

B*(s) being the Laplace–transform of the service time pdf of a packet.

For neg. exp. distributed packet lengths (mean $\bar{x}$) we have:

$$B^*(s) = \frac{1}{s\bar{x} + 1}$$

and:

$$F^*_f(s) = \frac{s\bar{x}(1-\varrho) + 1 - \varrho_r}{s\bar{x} + 1 - \varrho_r} \qquad \begin{aligned} \varrho &= \varrho_r + \varrho_t \\ \varrho_t &= \lambda_t\ \bar{x} \\ \varrho_r &= \lambda_r\ \bar{x} \end{aligned}$$

For the Laplace–transform of the pdf of ***e*** we get (same procedure, only $\varrho_t = 0$ –> a single buffer):

$$F^*_e(s) = \frac{s\bar{x}(1-\varrho_r) + 1 - \varrho_r}{s\bar{x} + 1 - \varrho_r}$$

In order to get the pdf of ***w*** we assume statistical independence between the waiting times of ***e*** and ***w*** (Note: this strange looking assumption is justified by the fact that the waiting times ***e*** and ***w*** are generated by completely different processes: ***e*** corresponds to the M/M/1 waiting time that arises when independently arriving packets have to be serially transmitted on the ring medium; while ***w*** is produced by the transmission of TB packets, i.e. by a completely different source.). The distribution of $\boldsymbol{f = e + w}$ is therefore the convolution of the distributions of ***e*** and ***w***:

$f_f(f) = f_w(f) * f_e(f)$ [Pap 84, p. 135].

And in the spectral range we have:

$$F^*_f(s) = F^*_w(s) \cdot F^*_e(s)$$

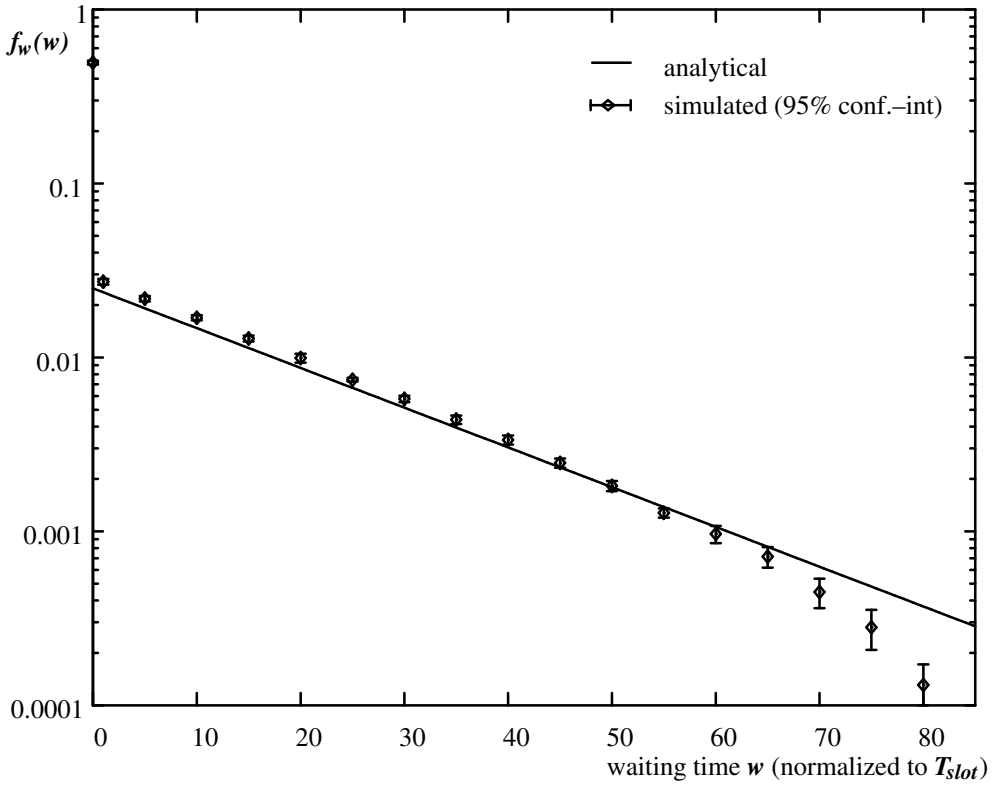
And:

$$F^*_w(s) = \frac{F^*_f(s)}{F^*_e(s)} = \frac{s\bar{x}(1-\varrho) + 1 - \varrho_r}{s\bar{x}(1-\varrho_r) + 1 - \varrho_r}$$

So that for the pdf of the waiting time ***w*** in the IB follows:

$$f_w(w) = \frac{1-\varrho}{1-\varrho_r}\delta(w) + \left(1 - \frac{1-\varrho}{1-\varrho_r}\right)\frac{1}{\bar{x}}e^{-w/\bar{x}}$$

The following figure shows this result compared with that of a simulation yielding a high congruence even at a high overall system load of S = 1.8 and a high number of nodes where the independence between packet lengths and interarrival times has completely vanished:



*pdf of waiting time **w** in the IB of a BI node for a ring with a length of **L** = 2km, at a normalized system load of **S** = 1.8.*

3.3 Distribution of the Number of Slots in IB and TB: f_b (b)

At the beginning of a scheduling cycle each node reports the number of slots ***b*** (waiting for transmission) in its transmission and insertion buffers to the scheduler via the reserve command.
The distribution of this important internal variable ***b*** can be evaluated using a Markov chain with

b as the state variable and considering only time instants at the completion of a transmission of a single slot. During the corresponding time interval T_{slot} the following events can occur:

- **arrival of a slot from the ring into the insertion buffer (IB)**; the corresponding probability is denoted with σ equal to the load ϱ_r into the IB of a node.
- **arrival of a packet from the host into the transmit buffer (TB)**; the corresponding probability is denoted with π.
  Since we assume a negative exponential distribution of the interarrival times at the TB the number ***k*** of arrivals during T_{slot} has a Poisson distribution (all times are normalized to T_{slot}):

  $$f_k(k) = \frac{\lambda_t^k}{k!} e^{-\lambda_t}$$

  The probability for zero arrivals during T_{slot} is therefore:

  $$f_k(0) = e^{-\lambda_t}$$

  And neglecting the probability of more than one packet arriving during T_{slot} we get:

  $$\pi = 1 - e^{-\lambda_t}$$

  (this approximation is justified by the fact that the mean interarrival time at the input of TB is much greater than T_{slot}, especially for rings with a large number of nodes and therefore a small load per node).
  We get the distribution $f_s(s)$ of the packet length ***s*** measured in slots from the continuous neg. exp. distribution $f_x(x)$ of the packet length (***x*** and the mean packet length $\overline{x}$ being normalized to the length of a slot) by the following considerations:

  $$s = 1 \quad if \quad 0 \le x < 1 \qquad so\ that \quad f_s(1) = \int_{x=0}^{1} f_x(x)dx = 1 - e^{-\frac{1}{\overline{x}}}$$

  $$s = i \quad if \quad i-1 \le x < i \qquad so\ that\ f_s(i) = \int_{x=i-1}^{i} f_x(x)dx = e^{-\frac{i-1}{\overline{x}}}\left(1 - e^{-\frac{1}{\overline{x}}}\right)$$

  and it follows the geometric distribution: $f_s(s) = (e^{-\frac{1}{\overline{x}}})^{s-1} \cdot (1 - e^{-\frac{1}{\overline{x}}})$

- **a slot is transmitted onto the ring** if at least one slot is waiting / or has arrived in IB or TB during T_{slot}.

Since for the sum ***b*** of slots in both IB and TB the priority mechanism of buffer insertion is unimportant the effects at the input and output side of the node lead to the following transitions of ***b*** after T_{slot}:

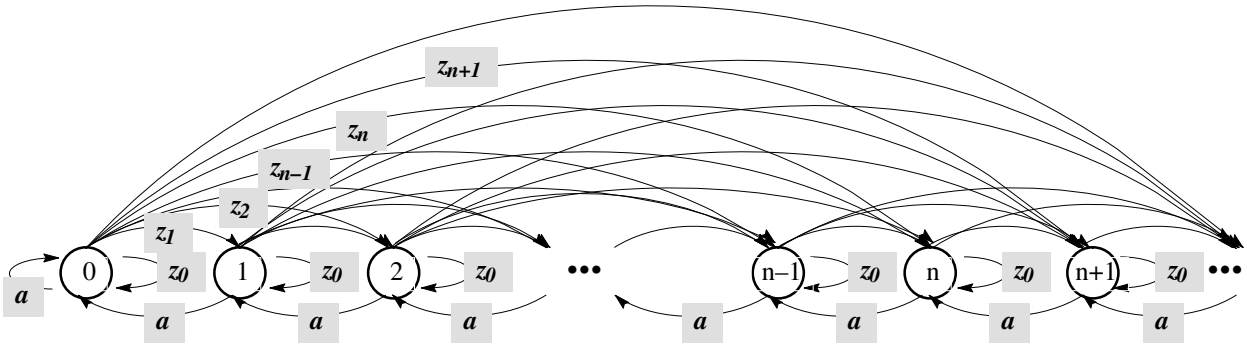
input of IB and TB: node output:

arrival of a TB−packet (length ***s*** slots)

arrival of a IB−slot	no	yes
no	$b \to b$	$b \to b+s$
yes	$b \to b+1$	$b \to b+s+1$

$b \to \max(b-1;0)$

So we can draw the following markov chain for the state variable ***b***:



a: probability of no arrival
z_i: probability that ***b*** is increased by ***i***

The probabilities of the transitions are:

- no TB–packet and no IB–slot arriving; $b \to \max(b-1;0)$:
  $a = (1-\pi)\cdot(1-\sigma)$
- ***b*** is increased by ***i*** slots; $b \to b+i$:
  $z_i = \sigma\cdot\pi\cdot f_s(i) +$ (TB–packet and IB–slot arriving; one slot is transmitted)
  $+ (1-\sigma)\cdot\pi\cdot f_s(i+1)$ (TB–packet and no IB–slot arriving; one slot is transmitted)
  special case:
  $z_0 = \sigma\cdot(1-\pi) + (1-\sigma)\cdot\pi\cdot f_s(1)$ (a single slot arriving; a slot transmitted)

With these probabilities we get the following flow equations for calculating the state probabilities $p_i = P\{b = i\}$:

	into state $\boldsymbol{b}$		out of state $\boldsymbol{b}$
state $\boldsymbol{b}$ = 0:	$a\ p_1$	=	$p_0 \sum_{i=1}^{\infty} z_i$
state $\boldsymbol{b}$ = 1:	$a\ p_2 + z_1\ p_0$	=	$p_1 \sum_{i=1}^{\infty} z_i$
state $\boldsymbol{b}$ = 2:	$a\ p_3 + z_2\ p_0 + z_1\ p_1$	=	$p_2 \sum_{i=1}^{\infty} z_i$
.	.	.	
.	.	.	
.	.	.	
state $\boldsymbol{b}$ = n:	$a\ p_{n+1} + \sum_{i=0}^{n-1} z_{n-i}\ p_i$	=	$p_n \sum_{i=1}^{\infty} z_i$

It can be shown (either by simple but extensive analytical considerations or by numerical and simulative evaluation) that the state probabilities of the above markov chain are also very good described by the following equation:

$$f_b(b) = f_m(0) \cdot \delta(b) + \sum_{i=1}^{\infty} f_m(i) \cdot (f_s(b))^{*i}$$

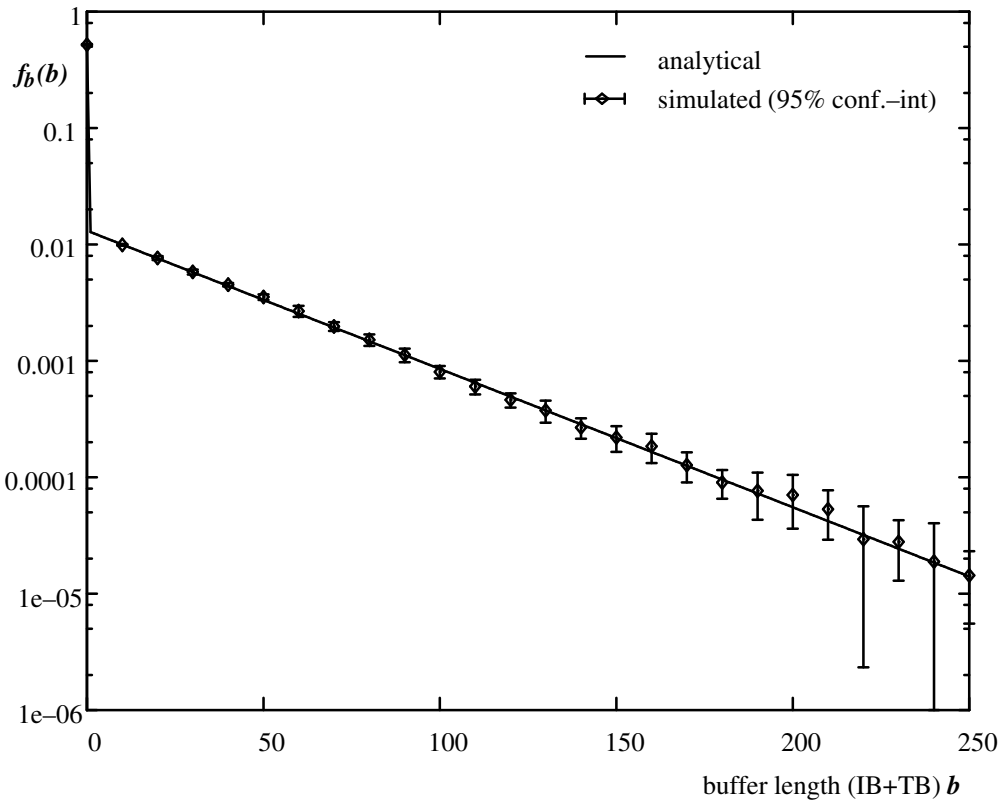
with:

$$f_s(s) = (e^{-\frac{1}{\bar{x}}})^{s-1} \cdot (1 - e^{-\frac{1}{\bar{x}}}) \qquad for\ s = 1\ ..\infty$$

and:

$$f_m(m) = (1 - \frac{\varrho_t}{1 - \varrho_r}) \cdot (\frac{\varrho_t}{1 - \varrho_r})^m \qquad for\ m = 0\ ..\infty$$

Where $(f_s(b))^{*i}$ denotes the (i–1)–fold convolution of $f_s(b)$, e.g. $(f_s(b))^{*3} = f_s(b)*f_s(b)*f_s(b)$.



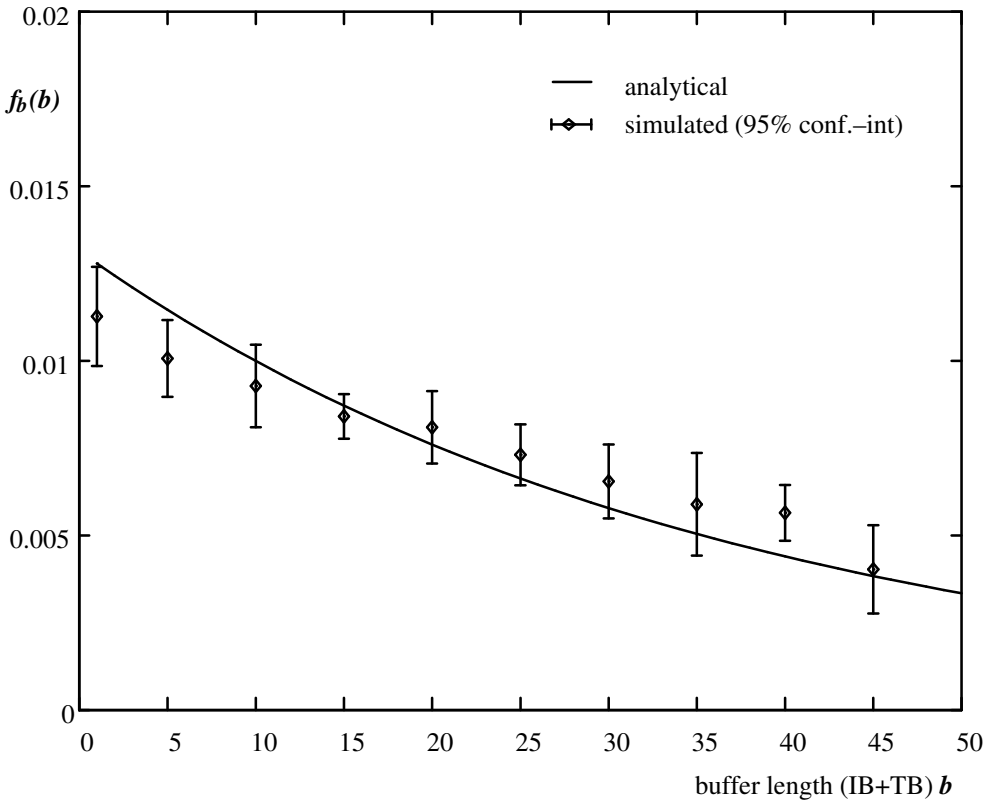
Comparison of the above equation for $f_b(b)$ and a simulation of the corresponding markov chain for a load of $S = 1.8$ and a mean packet length of $\bar{x} = 19$ slots.

The above equation for m is the also the geometric distribution of the number of packets in a virtual M/M/1 one–buffer queueing system with the load ϱ_t and a reduced server capacity of $1-\varrho_r$ (resp. a virtual increased load). The reason for this rather simple result is the following:

- The buffer length b mainly consists of the TB slots. So the TB packets constitute the number of packets in the queue of the virtual M/M/1 system. Slots are only found in the IB during and after the transmission of a TB packet. At a randomly chosen instant the slots already transmitted out of TB are compensated by the buffered slots in the IB. The latter ones together with the rest of the currently transmitted TB packet constitute the packet in the server of the virtual M/M/1 system
- The TB packets must be transmitted in the (collected) gaps of the IB traffic and thus can only use a reduced transmission capacity.

The number of slots in the system can then be derived by weighting the convolutions of the packet length distributions (measured in slots) with the corresponding probability of the number of packets in the system.

The results of this approach are compared with a simulated Buffer Insertion system at the following system parameters: $S = 1.8$, $L = 2$km



*pdf of the number of slots **b** in IB and TB of a node for a BI ring with a ring length of **L** = 2km at a normalized system load of **S** = 1.8.*

3.4 Distribution of the Number of Slots Transmitted Per Node and Cycle f_z (z)

Since the average packet length is small compared to the length of a scheduling cycle (see for example the plots in chapter 4.3) we make the assumption that all packets received from the host during a scheduling cycle are transmitted during a cycle.
Therefore we get $f_z(z)$ as the distribution of the number of slots received from the host of a node during a scheduling cycle (duration T_{cycle}).
The number g of packets received during a cycle has a poisson distribution:

$$f_g(g) = \frac{G^g}{g!} e^{-G} \qquad \text{with } G \text{ being the expectation of } g$$

With S (the total network throughput normalized to the link bit rate), M (the number of nodes in the network), and the average packet length $\bar{x}$ we have:

$$G = \frac{E(T_{cycle}) \cdot S}{\bar{x} \cdot M}$$

Note: in later chapters the distribution of z (being an input of the scheduling algorithm) is used for calculating the distribution of T_{cycle} itself. But the unwanted side effects of this logical feedback are limited since the expectation of T_{cycle} is strongly determined by the fixed parameters T_L (for large rings) and T_{calc} (for small rings). So here in advance we use the sum $T_L + T_{calc}$ as an expression for $E(T_{cycle})$.

Now we can calculate $f_z(z)$ by a weighted sum over the convolutions of the packet length pdf $f_s(s)$ measured in slots:

$$f_z(z) = f_g(0)\delta(z) + \sum_{i=1}^{\infty} f_g(i)\left(f_s(z)\right)^{*i}$$

no packet transm. — i packets transm.

A good approximation for low to medium throughput ($S = 0 .. 1.5$) and short to medium sized rings (T_{cycle} / M in the order of $\overline{x}$) is to neglect the transmission of more than one packet during a cycle:

$$f_z(z) \approx f_g(0)\delta(z) + \left(1 - f_g(0)\right)f_s(z)$$

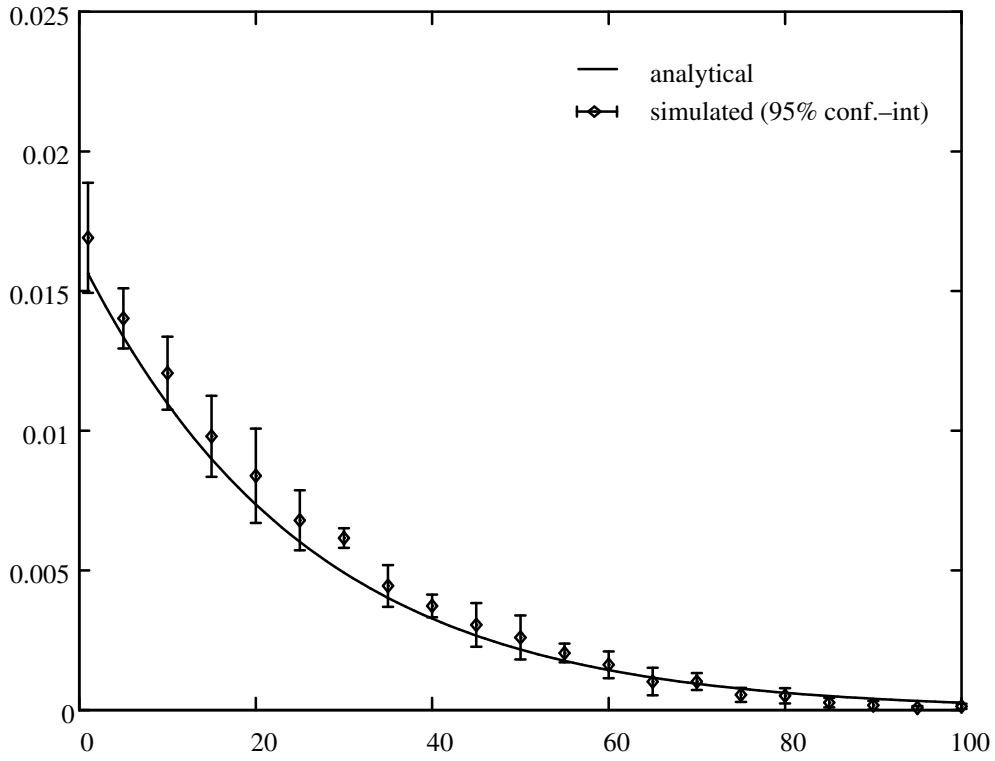
and with:

$$f_g(0) = e^{-G}$$

it follows that:

$$f_z(z) \approx e^{-G}\delta(z) + \left(1 - e^{-G}\right)f_s(z) \ .$$

In the CRMA–II protocol [v.As 92] the scheduler uses as one of its inputs the transmission counters of the nodes that are decremented each cycle by the passing confirm command so that the transmit counter cannot be described by the above equation of z. But as will be shown in chapter 4.1 this kind of manipulation of the transmit counters narrows the choices of possible thresholds and leads to the danger of overflowing transmit counters. So that in this analysis the transmit counters of the nodes are reset after each passing reserve command making z the correct input variable of the scheduler.



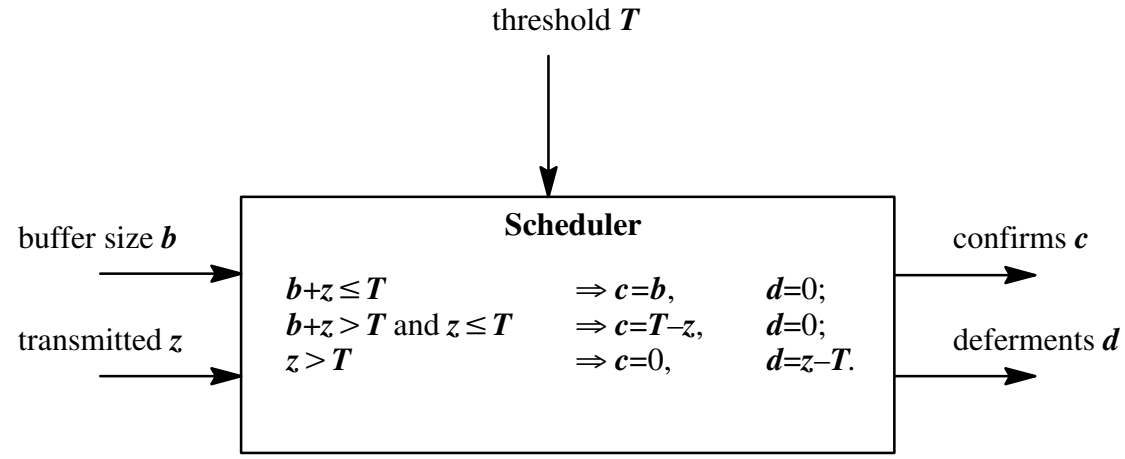
*pdf of the number of slots **z** transmitted per node and cycle for a BI ring having **L** = 2km and **S** = 1.4.*

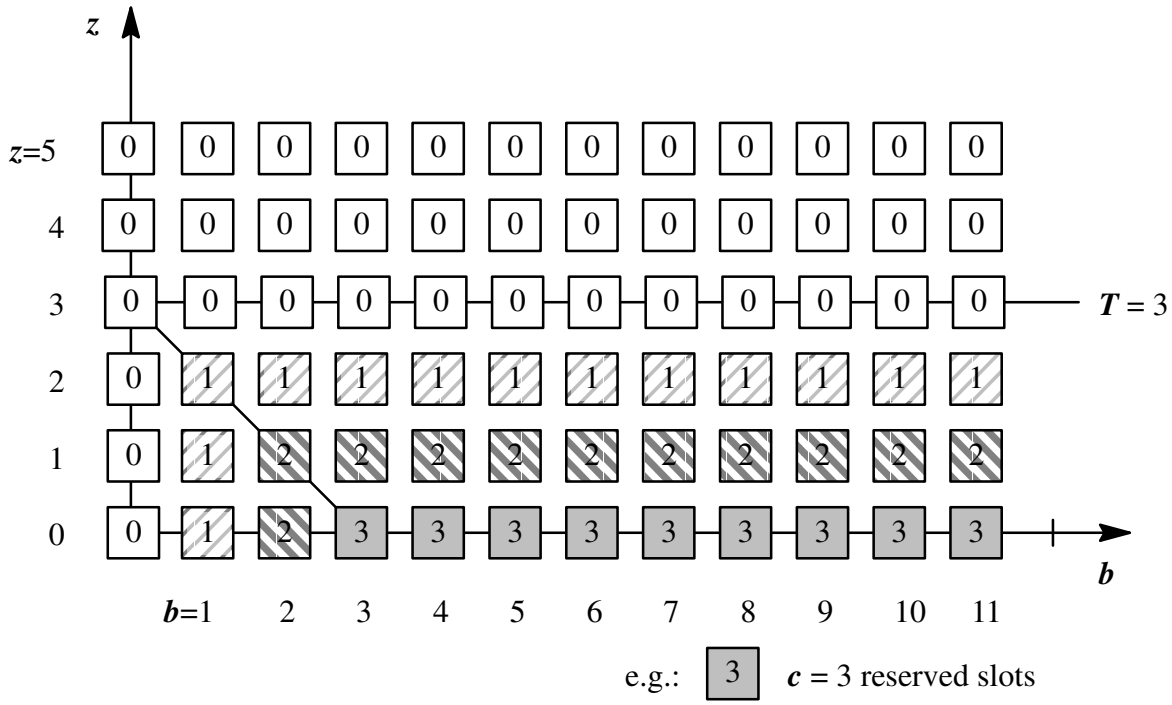
3.5 Distribution of the Number of Confirms Per Node and Cycle

From each node the scheduler gets as input the random variables

- z (number of slots accessed by this node during the last cycle) and
- $\boldsymbol{b}$ (number of discardible slots necessary to empty both the IB and TB of this node)

to calculate the number of confirms $\boldsymbol{c}$ and deferments $\boldsymbol{d}$ of this node:
To evaluate the nonlinear combination of $\boldsymbol{b}$ and z it is convenient to use the following two–dimensional diagram proposed in [Zin 92] (drawn here for a value of $\boldsymbol{T} = 3$) where $\boldsymbol{b}$, z and the resulting values of $\boldsymbol{c}$ are displayed:


threshold T
Scheduler
buffer size b
transmitted z
confirms c
deferments d
b+z ≤ T ⇒ c=b, d=0;
b+z > T and z ≤ T ⇒ c=T–z, d=0;
z > T ⇒ c=0, d=z–T.


z
z=5
4
3
2
1
0
T = 3
b
b=1 2 3 4 5 6 7 8 9 10 11
e.g.: 3 c = 3 reserved slots

From the above diagram we then can calculate the probabilities for the individual values of $\boldsymbol{c}$:

We get $\boldsymbol{c}$ = 0 confirms if: $\boldsymbol{b}$=0 and $z < \boldsymbol{T}$ or
z=$\boldsymbol{T}$ or
$z > \boldsymbol{T}$.

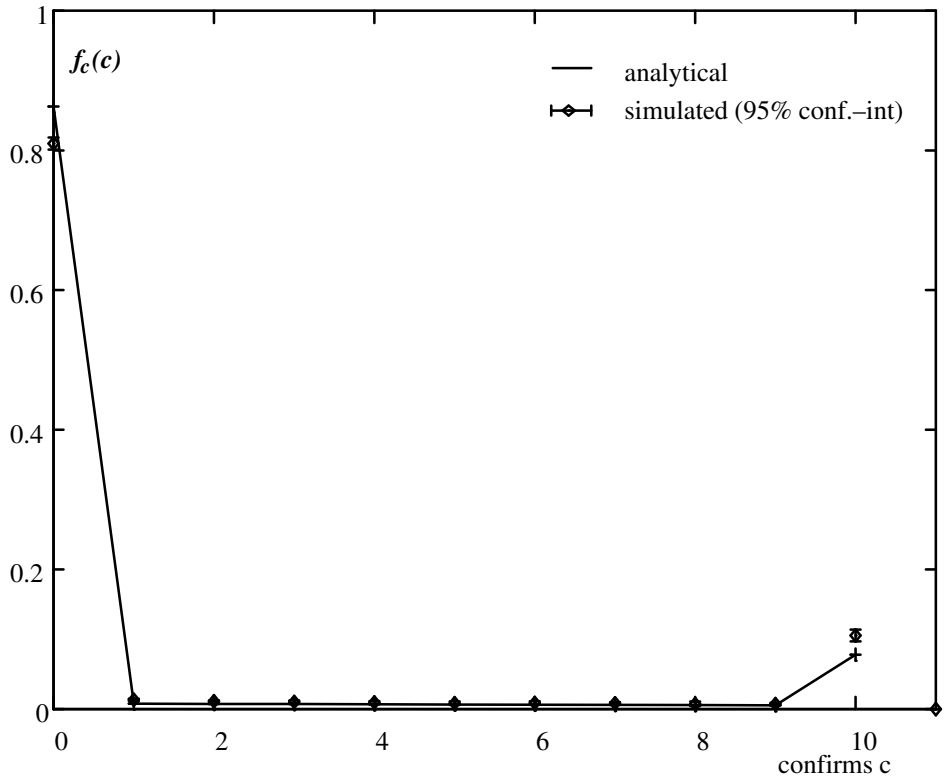
We get $\boldsymbol{c}$ = n confirms if: $\boldsymbol{b}$=n and $z < \boldsymbol{T}$–n or
$\boldsymbol{b} \geq$ n and z=$\boldsymbol{T}$–n.

We get $\boldsymbol{c} = \boldsymbol{T}$ confirms if: $\boldsymbol{b} \geq \boldsymbol{T}$ and z=0.

So that follows for the pdf $\boldsymbol{f_c(c)}$:

$$f_c(c) = \begin{cases} 1 - \left(1 - f_b(0)\right) \sum_{z=0}^{T-1} f_z(z) & for\ c = 0 \\ f_z(T - c)\left[1 - \sum_{b=0}^{c-1} f_b(b)\right] + f_b(c) \sum_{z=0}^{T-c-1} f_z(z) & for\ 0 < c < T \\ f_z(0)\left[1 - \sum_{b=0}^{T-1} f_b(b)\right] & for\ c = T \end{cases}$$

In the following figure these results are compared with those of a simulated BI ring:



pdf of the number of confirms **c** *per node and cycle for a BI ring with* $\boldsymbol{L}$ = *2km,* $\boldsymbol{S}$ = *1.4, and a threshold* $\boldsymbol{T}$ = *10.*

3.6 Distribution of the Number of Confirms Per Cycle (All Nodes) $f_{cc}\ (cc)$

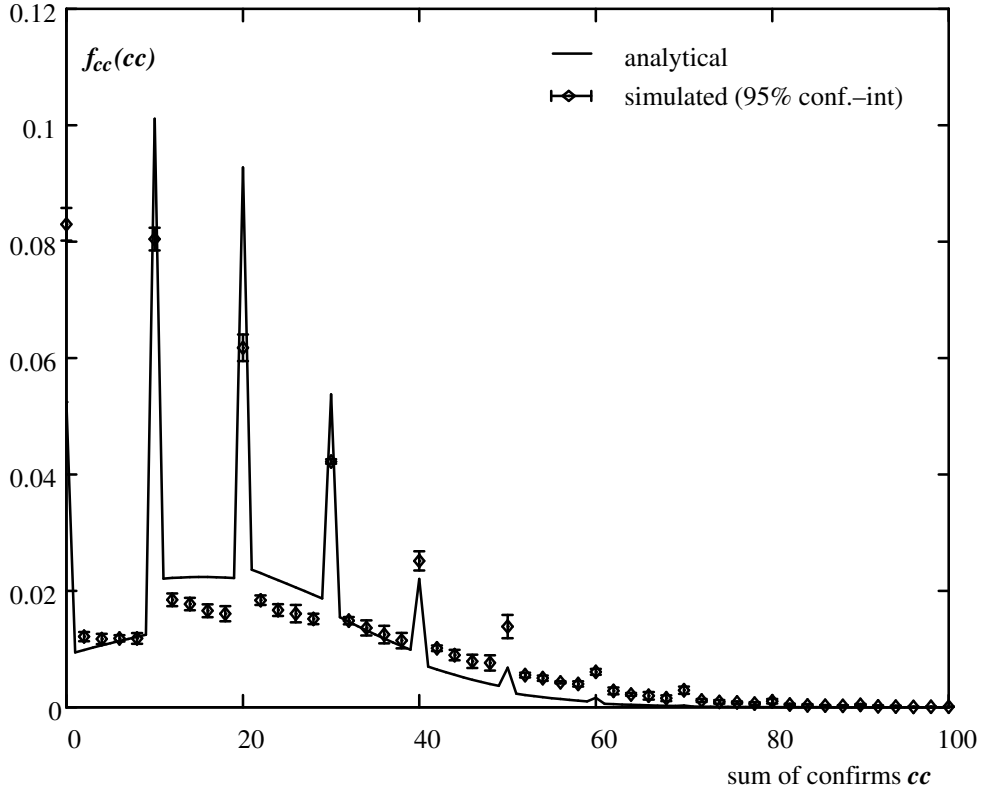
We get the pdf $f_{cc}\ (cc)$ of the sum of confirms per cycle over all nodes M by (M–1) times convoluting the pdf $f_c\ (c)$ with itself:

$$cc = \sum_{i=1}^{M} c_i$$

$$f_{cc}(cc) = \left(f_c(c)\right)^{*M}$$

The results are shown in the following figure:

(note: the spikes at multiples of the threshold T are produced by the rise of $f_c(c)$ at $c = T$.)



pdf of the sum of confirms **cc** *per cycle for a BI ring with* $L = 2km$, $S = 1.4$, *and* $T = 10$.

3.7 Distribution of the Number of Deferred Slots Per Node and Cycle $f_d(d)$

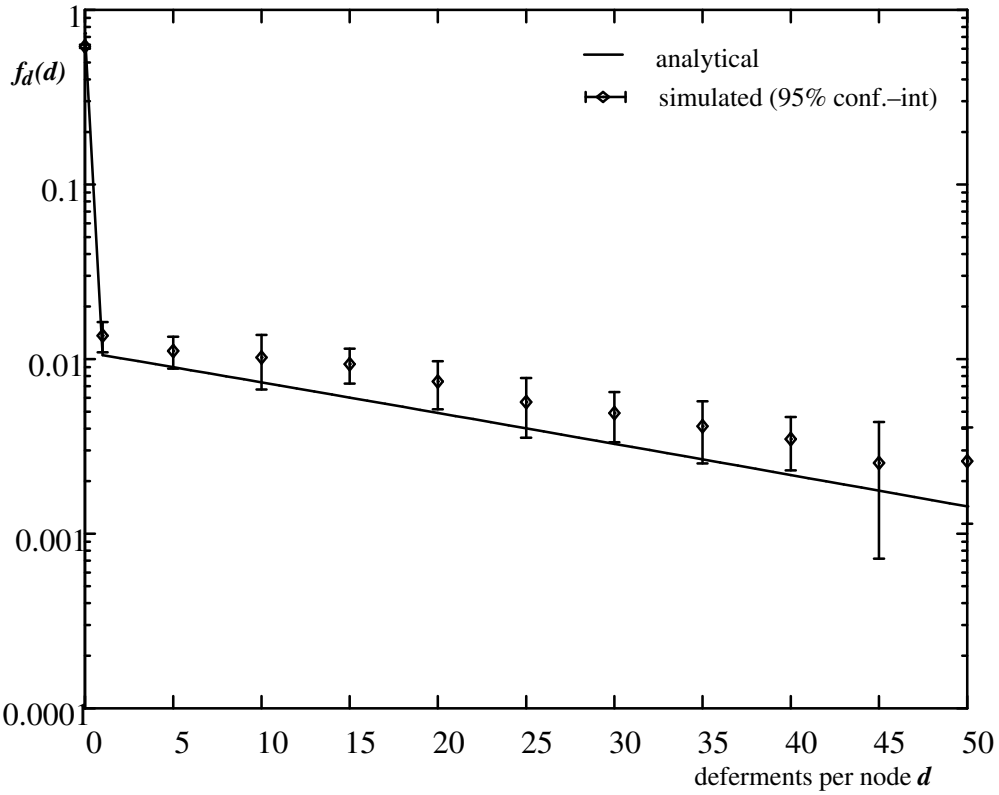
The pdf of d can be calculated using the same method as used for $f_c(c)$.

We get $d = 0$ deferred slots if: $z \leq T$

and d = n deferred slots if: $z = T + n$.

So it follows that:

$$f_d(d) = \left\{ \begin{array}{ll} \sum_{z=0}^{T} f_z(z) & for\ d = 0 \\ f_z(T + d) & for\ d > 0 \end{array} \right\}$$



pdf of the number of deferred slots ***d*** *per node and cycle for a BI ring with* ***L*** *= 2km,* ***S*** *= 1.4,* ***T*** *=10.*

4 Applications

4.1 Manipulation of the Transmit Counters in the Nodes

In the CRMA–II protocol as described in [v. As 92] the z–counters in each node are decremented by the value of the scheduling threshold T each time upon reception of the confirm frame: $z \rightarrow z–T$.

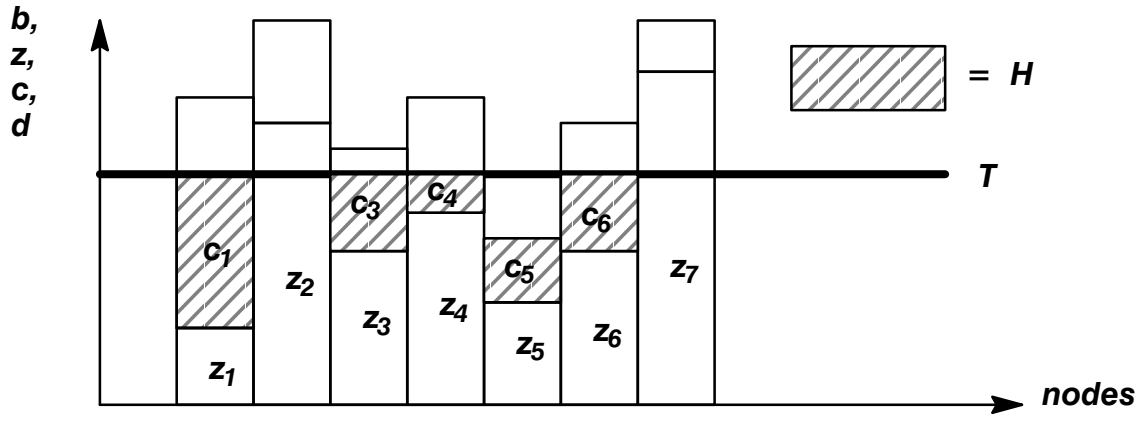
At a closer look one recognizes that this is nothing other than an implicit assumption about the value of the threshold: T must be equal to the mean value of z:

$$T=E(z))$$

otherwise sooner or later (the duration of a scheduling cycle being in the order of 10 ... 100 microseconds, see chapter 4.3 for details) overflows or underflows of some z–counters would occur.

As a consequence the threshold T cannot be chosen freely by the network management (resp. by an adaptive scheduler) in order to satisfy certain preferences concerning the desired values of confirms and deferments.

Another result is an incorrectnis in the calculation of T in the description of CRMA–II [v.As 92]: there only the total number of confirms H is preset by the network management and an individual T for each cycle is determined from H by a method that resembles the filling of a water reservoir by putting the transmission counts z_i side by side:



The common threshold T is then determined by:

$$H = \sum_{i=1}^{M} c_i\ (T)$$

where the case I that $z_i + b_i < T$ is respected and in the case that fewer than H slots are requested by the nodes the threshold should be set to twice T_L (measured in slots).

Using this kind of scheduling the probability of getting $T = E(z)$ is very small so that the transmit counters can always be expected to overflow of underflow.

A possible solution to this problem is to reset the z–counters to zero periodically after some cycles.

In our analysis we assume a reset of z every new cycle after the passing of the reserve command in order to reduce the complexity of the calculations. This also allows a free choice of the threshold T.

4.2 Optimum Scheduling Threshold

Since the CRMA–II mechanisms of reservation and deferment are only superimposed the basic buffer insertion access to ensure maximum fairness in throughput and delay it is desirable that these

mechanisms impair the excellent (but sometimes unfair) characteristics of buffer insertion as less as possible.

High numbers of confirms ***c*** and deferments ***d*** per node directly impair the possible overall system throughput:

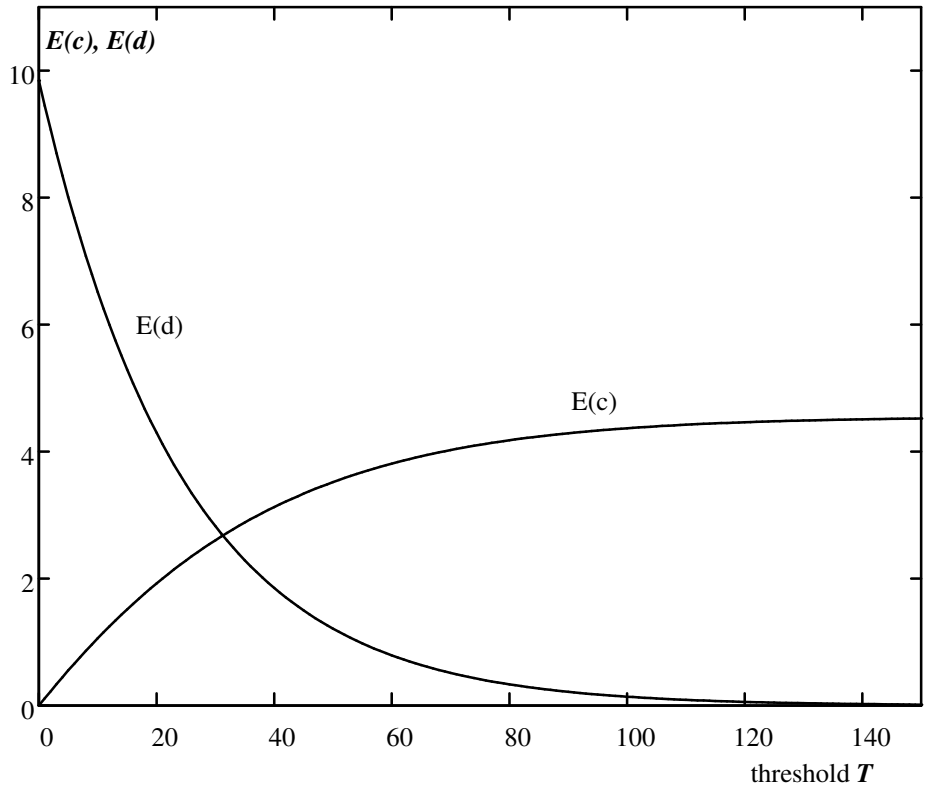
- Reserved slots can only be used (by discarding them at the input of a node) by nodes holding confirms. Other nodes have to pass them by on frame boundaries thus delaying the transmission of a packet out of the IB or TB.
- Nodes with a deferment counter greater than zero have to delay their access even if downstream nodes do not have slots to transmit anymore (scheduling reactions always happen delayed and thus cannot reflect the actual situation of the single nodes).

Thus an appropriate scheduling threshold has to be chosen carefully and it becomes evident that methods for determining ***T*** that consider only the resulting sum ***H*** of confirmed slots and not the deferments [v.As 92] may lead to suboptimal overall system behaviour.

4.2.1 Characteristics of the Scheduling Variables

The expectations of the confirms and deferments per node and cycle depending on the value of ***T*** can be determined from the equations of chapter 3 showing the basic characteristics of how fairness is achieved in CRMA–II:

$$E(c) = \sum_{c=0}^{c=T} c\, f_c(c) \qquad \text{and} \qquad E(d) = \sum_{d=0}^{\infty} d\, f_d(d)$$



*Expectations of **c** and **d** for a single node at various threshold values **T***
*(parameters: **S** = 1.4, **L** = 50km)*

At low levels of T nearly all transmitting nodes get deferred leading to a maximum value of $E(d)$ at $T = 0$ that equals the expectation of transmitted slots per node and cycle: $E_{max}(d) = E(z)$. For T going to infinity $E(d)$ becomes zero.
Similarly there exist no confirms for $T = 0$ and $E(c)$ reaches its maximum for $T \rightarrow \infty$ when all buffer length values lead to a corresponding number of confirms: $E_{max}(c) = E(b)$.
Concerning the two scheduler input variables it can be seen that $E(z)$ mainly depends on the duration of the scheduling cycles (dominated itself by the ring length T_L and the duration of the fairness calculations T_{calc}) while $E(b)$ is determined by the system load and independent of the ring length. Therefore no optimum solution of T can exist for all possible topologies and load scenarios.

The equations describing the scheduling variables that have been derived in the previous chapters can assist in the decision concerning the tradeoff between the consequences of reservations and deferments.

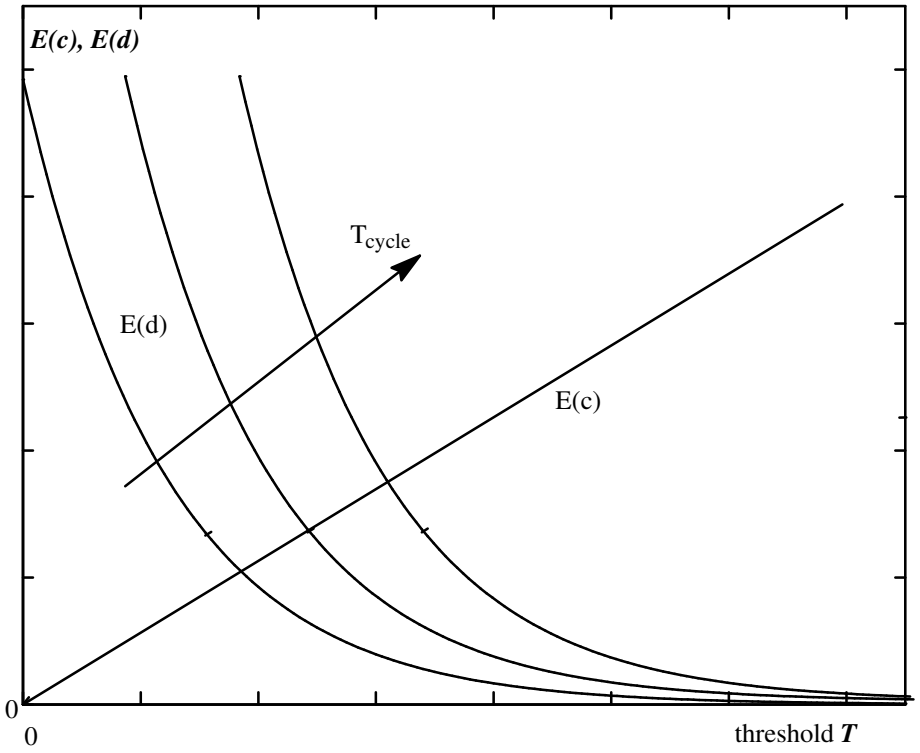
So to select an appropriate threshold at different load scenarios the network management could either use

- a lookup table to find T_{opt} for a certain S (corresponding to the desired levels of confirms and deferments) or
- an adaptive scheduling technique (e.g. that proposed in [Lem 93], see below).

In the following chapters of this analysis $T = E(z)$ is used for simplicity reasons unless otherwise stated.

4.2.2 Instability of Adaptive Scheduling at Overload Scenarios

In [Lem 93] an adaptive mechanism has been proposed that for each individual scheduling cycle determines an optimum scheduling threshold by first considering the sums of confirms and deferments of this cycle resulting form different values of T and then selecting the one T that minimizes the sum.
The instability of this technique in overloaded CRMA–II rings (that has also been shortly mentioned in [Meu 94]) resulting in unbounded duration of the scheduling cycles T_{cycle} can now be investigated here using the techniques of chapter 3.
Consider the influence of a slight rise of T_{cycle} on the scheduling variables:



Tendencies of E(c) and E(d) at overload

An increased T_{cycle} leads to a greater number z of slots transmitted by a node. At a constant value of T thus more deferments would result, corresponding to a shift of $E(d)$ to the right in the figures above.
At overload the number of confirms $E(c)$ depends only on T (the transmission buffer always being filled up) leading to the linear rise of $E(c)$ with T in the figure above. So with the shifted $E(d)$ the new minimum of $E(d)+E(c)$ is also shifted to the right side and the adaptive algorithm will increase T. This in turn will lead to more confirms and a longer marking time T_M of the scheduler so that T_{cycle} is further increased and the positive feedback circle is closed. Thus useless high values of T are determined only slowing down the reaction of the scheduler while at normal loads a stable equilibrium is guaranteed by $E(b)$ limiting $E(c)$).
A possible solution to avoid this instability is to impose an upper limit T_{max} on the allowed values of T. The figure of chapter 4.2.1 shows that under normal conditions the threshold T will always be in the order of the number of slots z transmitted per node and cycle. Since from chapter 3.4 it follows that:

$$E(z) = E(T_{cycle}) \cdot S/M$$

the equations for the distribution of T_{cycle} in chapter 4.3 allow the estimation of a reasonable T_{max}.

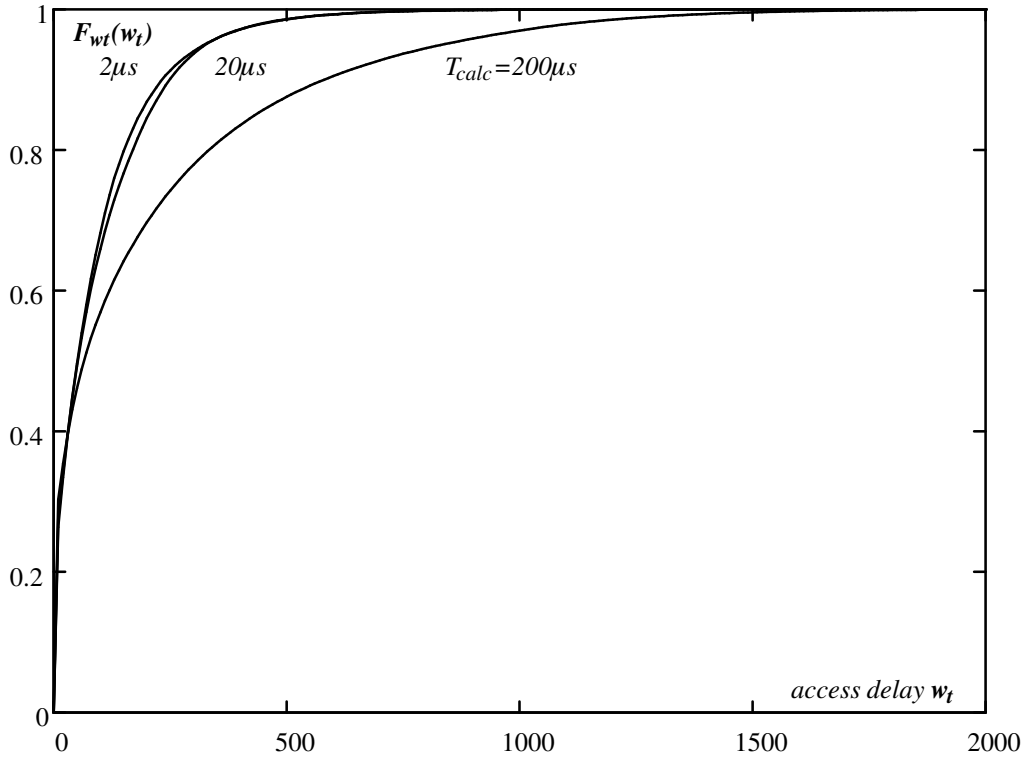
4.3 Determination of the Duration of Scheduling Cycles

4.3.1 Importance of the Duration of the Scheduling Cycle

As the reservation mechanism is responsible for maintaining the throughput fairness over all nodes the duration of a scheduling cycle directly determines the worst case access delay of a single node.

For example a node located directly behind a heavy user may be unable to access the ring by the normal use of gratis slots because all incoming slots are busy transporting packets of the heavy user.

So this node depends on reserved slots and deferments of the upstream node calculated by the scheduler with a reaction time determined the duration of the next scheduling cycle T_{cycle} . The following figure shows the probability distribution function (PDF) of the access delay w_t of a node positioned directly behind a heavy user node on a CRMA–II ring. The parameter is the duration of the scheduling calculations T_{calc} that directly influences T_{cycle} and $F_{wt}(w_t)$ (as it can be seen a reduction of T_{calc} below the order of the round trip delay T_L (10µs at L=2km) has no further positive effect of $F_{wt}(w_t)$, s. chapter 4.3.7 for details):



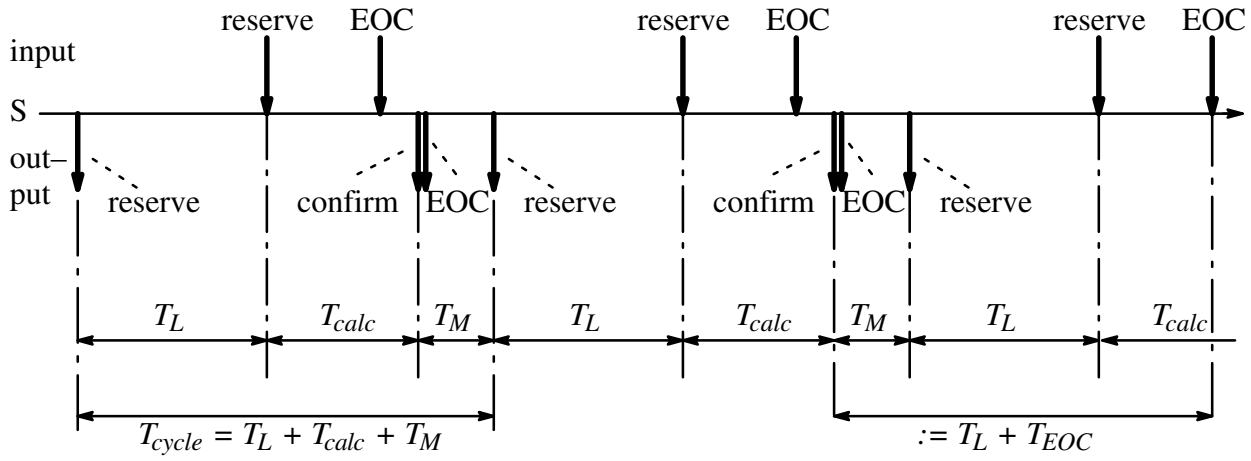
$F_{wt}(w_t)$ of a node behind a heavy user (i.e. having never an empty transmission buffer) at S=1.0 (all nodes together except the heavy user), M=20 and L=2km. Parameter is T_{calc}.

Therefore T_{cycle} is an important parameter of the network behaviour that has to be minimized as far as possible.

4.3.2 Components and Parameters of T_{cycle}

A closer look at the details of the scheduling cycles shows that the timing of the arrival of the EOC signal at the scheduler distinguishes two cases how T_{cycle} is composed:

1) EOC arrives when the scheduler is still computing the individual confirms and deferments of the nodes (arrival during T_{calc}):

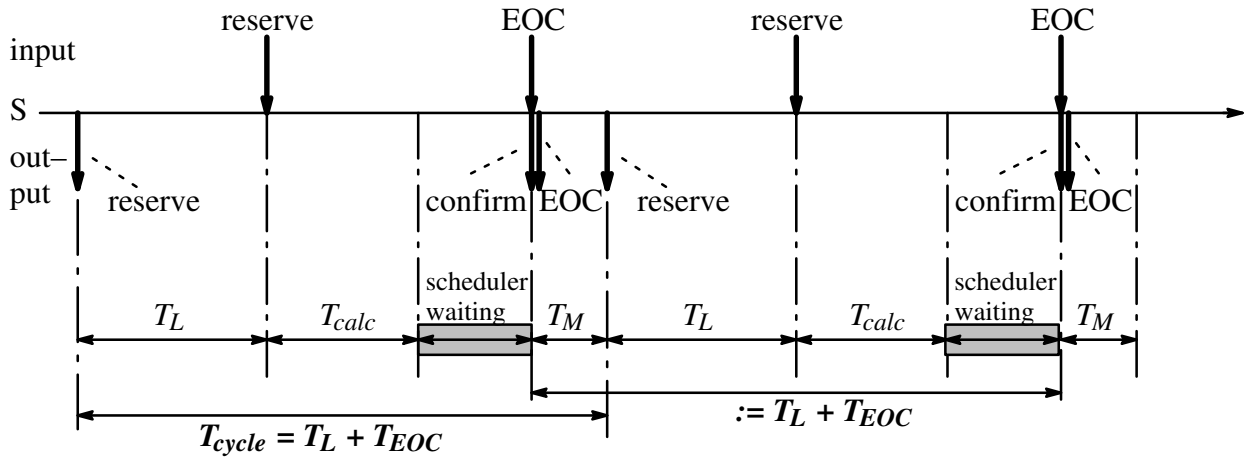


The above graphic shows the timing of the relevant input resp. output signals of the scheduler S (arrows pointing to resp. from the horizontal S–line).
After emitting a reserve frame the scheduler has to wait the round trip delay $\boldsymbol{T_L}$ for its return. The scheduler then calculates the confirms and deferments of the nodes (during calculation time $\boldsymbol{T_{calc}}$). In the considered case the EOC signal arrives during $\boldsymbol{T_{calc}}$ and the scheduler is able to emit a confirm frame directly after the completion of the scheduling calculations. The first marked slot contains the EOC signal.
So in this case the scheduling cycle has a duration of

$$T_{cycle} = T_L + T_{calc} + T_M.$$

For convenience in the later analysis we define here the time interval between the emission and the reception of the EOC signal as the sum $\boldsymbol{T_L + T_{EOC}}$ (the time necessary for the EOC signal to return to the scheduler will always be at least the round trip delay $\boldsymbol{T_L}$).

2) EOC arrives after the completion of the fairness calculations:



In this case the EOC signal has been delayed by certain processes (see detailed analysis below) so heavily that it arrives after $\boldsymbol{T_{calc}}$. Therefore the scheduler has to wait actionless before it can complete

the current cycle.
In this case we have a cycle duration of

$$T_{cycle} = T_L + T_{EOC}.$$

As a result we can say that $\boldsymbol{T_{cycle}}$ is determined by the comparison of $\boldsymbol{T_{EOC}}$ on the one side and the sum of $\boldsymbol{T_M + T_{calc}}$ on the other:

case 1: $T_{EOC} < T_M + T_{calc} \Rightarrow T_{cycle} = T_L + T_M + T_{calc}.$
case 2: $T_{EOC} > T_M + T_{calc} \Rightarrow T_{cycle} = T_L + T_{EOC}.$

Characteristics of the above variables:
$\boldsymbol{T_L}$ is only determined by the physical ring topology;
$\boldsymbol{T_{calc}}$ is a fixed parameter depending on the number of nodes and the implementation of the scheduler;
$\boldsymbol{T_M}$ depends on the sum of confirms ***cc*** in the current cycle and is therefore a random variable described by the equations of chapter 3.6;
$\boldsymbol{T_{EOC}}$ is a random variable determined by internal processes of the CRMA–II protocol (see below).

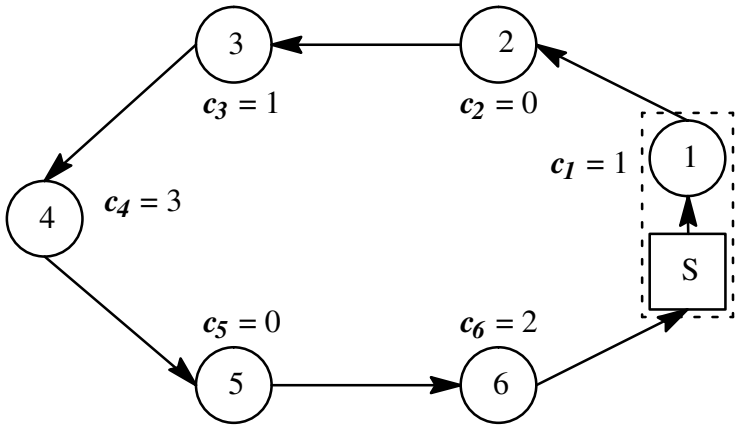
4.3.3 Effects determining T_{EOC}

Resulting from the above definition $\boldsymbol{T_{EOC}}$ is independent of the physical round trip $\boldsymbol{T_L}$ delay of the ring and is determined only by the time the EOC signal is delayed in the single nodes of the network. As mentioned before a node receiving the EOC signal is only allowed to retransmit it if it had the possibility to access a number of reserved slots corresponding to the confirms it got in the current scheduling cycle. A node having pending confirms holds the EOC signal until its confirm counter gets zero.
So two effects delay the retransmission of the EOC signal:

4.3.3.1 Enqueueing of BR–slots in the Insertion Buffers:

During $\boldsymbol{T_M}$ the scheduler marks a number of gratis slots equivalent to the sum of confirms during this cycle as reserved:
FG slots get FR,
BG slots get BR.
As this train of reserved slots travels along the ring the FR slots are either used (by discarding them) by nodes with confirm counter greater than zero or bypassed by nodes with zero counter values.
The BR slots are enqueued in the IB of each station after the scheduler until they reach their destination where they get FR by destination removal of the load. Therefore each BR slots gets delayed by the waiting times in the IBs of nodes between scheduler and destination address. This effect tears apart the train of reserved slots and nodes that depend on originally BR slots that get FR on their way over the ring have to wait longer for the arrival of those slots.
Within the train of reserved slots the BR slot with the highest destination address (***DA***) experiences the greatest queueing delay. And the node that turns out to be the user of this reserved slot (after it has been emptied by the destination node) will hold the EOC signal for the longest time.
Note: since our definition of $\boldsymbol{T_{EOC}}$ excludes the always present round trip delay $\boldsymbol{T_L}$ the basic propagation delays between the nodes do not have to be considered here.

Example:
Consider the following CRMA–II ring with $\boldsymbol{M}$ = 6 nodes:



We assume that at the end of T_{calc} the scheduler has computed confirm counter values c_i as shown in the figure. During the marking phase (after the return of the EOC signal) the scheduler marks cc incoming slots as reserved:

$$cc = \sum_{i=1}^{M} c_i = 7$$

Thus the train of reserved slots at the scheduler output may look like:

slot a	b	c	d	e	f	g
FR	BR DA = 3	BR DA = 5	BR DA = 5	FR	FR	BR DA = 2

(slot a is trans–mitted first)

Starting at node 1 this train is torn apart:
Slots **b**, **c**, **d**, and **g** are enqueued in the IB of node 1 (that also discards slot **a** and emits a BG).
In node 2 slot **g** is emptied (BR –> FR) and bypassed over the insertion buffer (IB) while slots **b**, **c**, and **d** are again enqueued.
So in the end slots **c** and **d** experience the greatest queueing delay (in the IBs of stations 1 to 4).
Note: The delay of the slots **b** and **g** cannot be greater than the delay of slots **c** and **d** because in the IBs of nodes 1 (in the case of slot **g**) and 1 and 2 (in the case of slot **b**) they are enqueued directly together with **c** and **d** and thus experience an identical delay.

In the following we will refer to this component of T_{EOC} as T_{BR}.

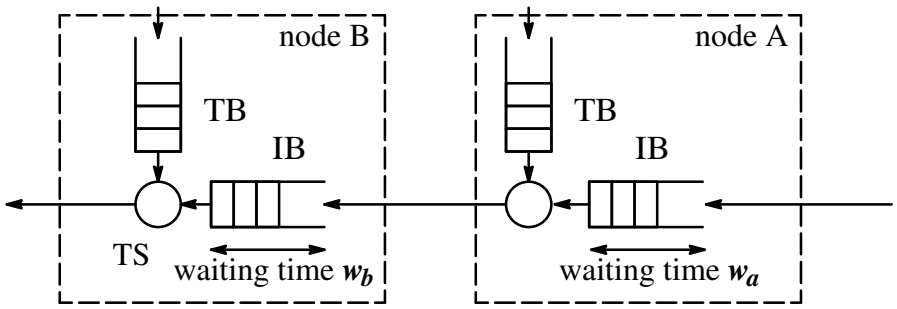
The distribution of $\mathbf{T_{BR}}$ can be calculated in three steps:

1) The pdf of the waiting time w of a busy reserved (BR) slot in the insertion buffer (IB) of a single CRMA–II node is expressed using the formula derived in chapter3.2:

$$f_w(w) = \frac{1-\varrho}{1-\varrho_r}\delta(w) + \left(1 - \frac{1-\varrho}{1-\varrho_r}\right)\frac{1}{\bar{x}}e^{-w/\bar{x}}$$

2) In the next step we calculate the pdf of the waiting time that a BR slot experiences by passing through several IBs until it reaches its destination node:

For this purpose we first consider a BR slot that passes through two IBs (of node a resp. b):



With the assumption that the waiting times w_a and w_b in the IBs are independent random variables having the pdf $f_w(w)$ we get the distribution of $w_c = w_a + w_b$ as the convolution of $f_{wa}(w_a) = f_w(w)$ and $f_{wb}(w_b) = f_w(w)$:

$$f_{wc}(wc) = f_w(w) * f_w(w).$$

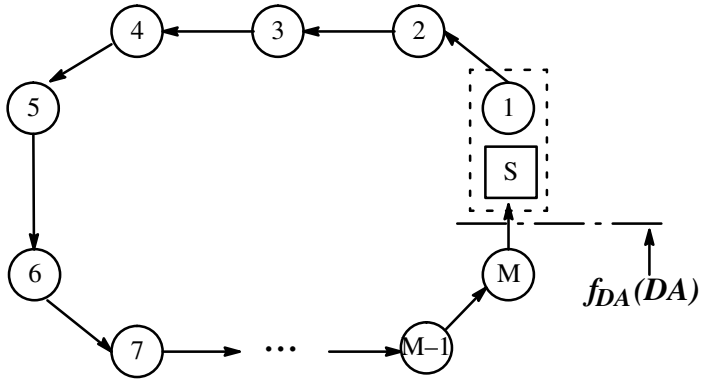
A BR slot with a destination address of $DA = n$ has to pass through $n-1$ IBs between scheduler and destination node. The distribution of the delay w_{tot} experienced by this BR slot in the $n-1$ IBs therefor can be calculated by $(n-1)$ times convoluting $f_w(w)$:

$$f_{w_{tot}}(w_{tot}) = (f_w(w))^{*n}$$

3) finally we determine the pdf of T_{BR} by considering the waiting time of the IB slot with the maximum destination address within a cycle:

For the calculation of the pdf of this component of T_{EOC} we need the distribution of the max DA within a scheduling cycle: $f_{DAmax}(DA_{max})$.

In the case of small rings with the expectation of the sum of confirmed slots per cycle is in the order of the mean packet length we can approximate $f_{DAmax}(DA_{max})$ by $f_{DA}(DA)$. Where $f_{DA}(DA)$ denotes the distribution of the destination address of busy slots at the scheduler input port:



The distribution of ***DA*** can be determined by considering the packets that move into the scheduler S:

packets from node number:	destined to node number:
M	1, 2, 3, ..., M−1
M−1	1, 2, 3, ..., M−2
...	...
4	1, 2, 3
3	1, 2
2	1

So we have altogether

(M–1)M/2

source–destination address combinations. And for the destination address ***DA*** we have a share of ***M–DA*** moving through the scheduler so that we get:

$$f_{DA}(DA) = 2\frac{M - DA}{M(M-1)}$$

For large rings with an expected number of confirms greater than the packet length we have several packets that get marked during a scheduling cycle. Thus in the distribution of DA_{max} high values of DA_{max} will get more emphasized. This partly counteracts the declining tendency of $f_{DA}(DA)$ so that it is a good approximation to assume a uniform distribution of $f_{DAmax}(DAmax)$ for large rings:

$$f_{DAmax}(DAmax) = \frac{1}{M-1}$$

The corresponding distribution of DA_{max} is then used the weigh the convolution terms constituting $f_{TBR}(T_{BR})$:

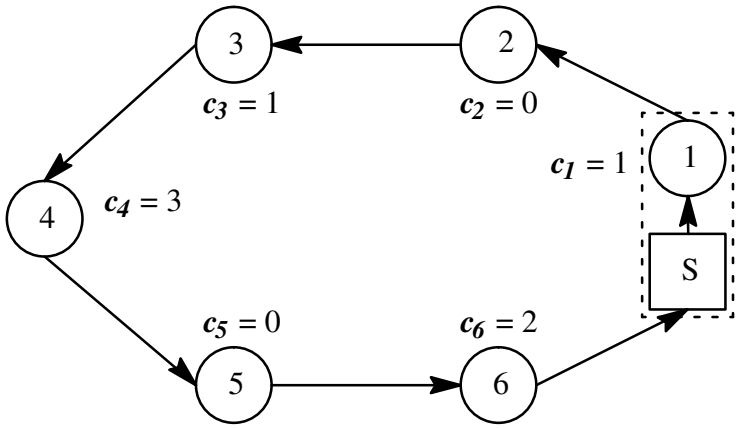
$$f_{TBR}(T_{BR}) = \underbrace{f_{DAmax}(1)\ \delta(T_{BR})}_{\text{DAmax = 1 –> BR slot passes through 0 IBs}} + \underbrace{\sum_{i=1}^{\infty} f_{DAmax}(i+1) \cdot \left(f_w(T_{BR})\right)^{*i}}_{\text{DAmax = } i+1 \text{ –> BR slot passes through } i \text{ IBs}}$$

4.3.3.2 Overtaking of Nodes

This second effect makes nodes wait even longer before they pass on the EOC signal. Due to the fact that in CRMA–II also BG slots can become marked by the scheduler a situation may occur where a node i still has a confirm counter c_i > o when all reserved slots of the current cycle have passed. This effect is illustrated in the following example:

After T_{calc} (and reception of EOC) the scheduler generates the following train of reserved slots:

Slots **a** and **d** are accessed (FR –> BG) in nodes 1 and 3. In node 4 slots **b** and **c** are emptied and can be accessed immediately. But after that there are no more accessible reserved slots left for node 4 to satisfy its third confirm.
Node 4 therefore stays with $c_4 = 1$ after all remaining reserved slots have passed by and holds the



slot a	b	c	d	e	f	g
FR	BR DA = 3	BR DA = 3	FR	BR DA = 6	BR DA = 5	BR DA = 5

(slot a is trans–mitted first)

EOC signal: node 4 has been ”overtaken”.
The remaining reserved slots **e**, **f**, and **g** then move on to nodes 5 and 6 where all are emptied (BR –> FR). Two of the resulting FR slots (**e** and **f**) are accessed by node 6.
Slot **g** now moves on through the scheduler and nodes 1, 2, and 3 until node 4 can discard it at last. Now node 4 re–emits the EOC signal that reaches the scheduler without any further delay ($c_5 = c_6 = 0$).
So the event of one or more overtaken nodes increases T_{EOC} by exactly one round trip delay T_L.
Note: Nodes after the overtaken one still delay the remaining BR slots by enqueueing them so that the first effect stays untouched by the second: the slots still needed by the overtaken node are further delayed (we therefore have an addition of a fixed T_L to T_{EOC}).

The probability P_o that one or more nodes are being overtaken during a scheduling cycle therefore depends not only on the number of nodes (M), the distribution of the sum of confirms during a cycle ($f_{cc}(cc)$), and the distribution of the share of busy slots ($f_h(h)$) within the marked slots (that all could easily be determined using the equations of chapter 3, $f_h(h)$ having a binomial distribution with the parameter ϱ_r and cc being the number of trials).
But P_o also depends on the actual sequence of BR and FR slots within the train of marked slots in interaction with the pattern how the sum of confirms cc of the actual cycle is distributed over the nodes of the network. And this interaction converts the exact determination of P_o into a very ugly combinatorical problem since all possible combinations of the number of confirms c_i of the single nodes (adding up to cc) and all possible combinations of the destinations addresses of the busy slots have to be considered together with their corresponding probabilities.

For low to medium system loads ($S \lesssim 1.0$) it is nevertheless possible to give a quite good approximation for P_o that still reflects properly the influence of all interesting network parameters (M, T, S, and L):

A closer look at the mechanisms of overtaking shows that the number of nodes M, the number of confirms cc, and the number h of busy marked slots (BR slots) within a cycle are orthogonal determi-

nants of P_o, i.e. for a fixed number of nodes and a certain combination of ***cc*** and ***h*** within a cycle we always get the same P_o for this cycle regardless of the other system parameters (***S***, ***L***, etc.).

Up to medium system loads the relative share of those scheduling cycles that contain so many confirmed slots that their P_o differs significantly from $P_o(cc=h=1)$ is rather small so that it is justified to consider only ***cc***=***h***=1:

$$P_o(cc \geq 1,\ h \geq 1) \approx P_o(cc=1,\ h=1).$$

And a large share of cycles does not contain any reserved slots at all with the consequence of no overtaking in these cycles.
So we can approximate:

$$P_o \approx P_o(cc=1,\ h=1) \cdot (1 - f_{cc}(0)).$$

We get $f_{cc}(cc=0)$ from the equations of chapter 3 and $P_o(cc=1,\ h=1)$ can be calculated using the following considerations:

In the case of only one BR slot and no FR slots overtaking occurs if the destination address of the BR slot is greater than the address of the node holding the corresponding confirm:
e.g. for a ring with ***M*** = 20 nodes (1..20) and a destination address ***DA*** = 4 of the BR slot we have an overtaking if the address of the node holding the corresponding confirm is in the range of 1..3.
Thus having the ***DA*** distributed with

$$f_{DA}(DA) = 2\frac{M - DA}{M(M - 1)}$$

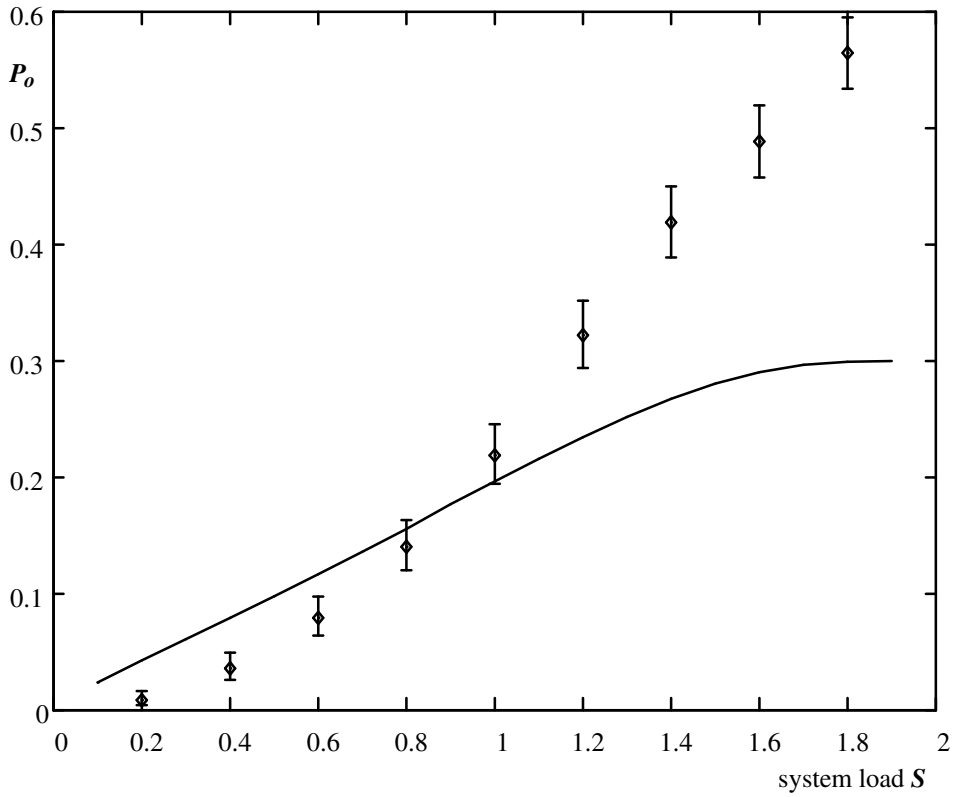
and the conditional probability of overtaking for a certain ***DA*** of

$$p_{DA} = \frac{DA - 1}{M}$$

we get for all possible ***DA*** = 1 .. ***M***–1 of the BR slot:

$$P_o(1,1) = \sum_{DA=1}^{M-1} \left(2\frac{M - DA}{M(M - 1)} \frac{DA - 1}{M} \right) = 2 \sum_{DA=1}^{M-1} \frac{(M - DA)(DA - 1)}{M^2(M - 1)}$$

The range of ***S*** where this approximation can be used is shown in the following diagram:



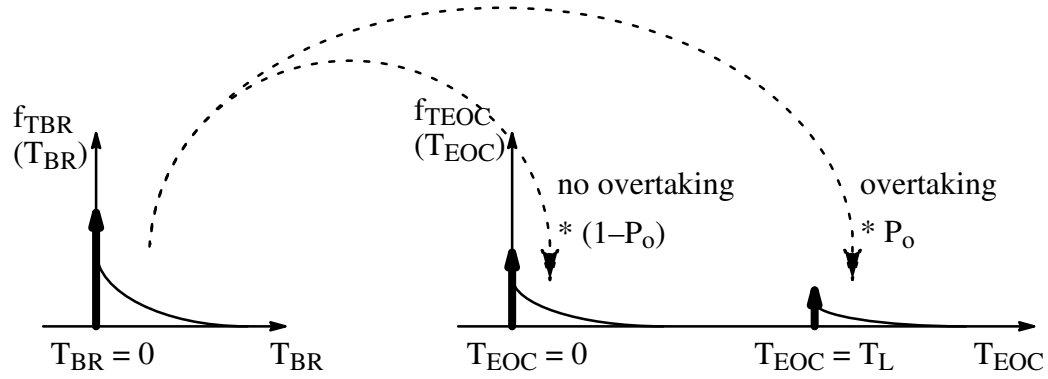
calculated and simulated overtaking probability P_o at different loads S

For higher values of S a much better accuracy of P_o can be achieved by embedding a miniaturized simulation within the calculation of $f_{Tcycle}(T_{cycle})$ that evaluates properly and very fast the above mentioned combinatorical problem (see chapter 4.3.3.4 for details).

4.3.4 Approximative Algorithm for Determining the Distribution of T_{cycle}

The pdf of T_{cycle} is now computed in the following three steps:

1. Calculation of the pdf $f_{TBR}(T_{BR})$ with the formulas discussed in chapter 4.3.3.1.
2. Calculation of the pdf of the end–of–cycle time T_{EOC}:
   Without overtaking $f_{TEOC}(T_{EOC})$ would be distributed like $f_{TBR}(T_{BR})$ while in the case of overtaking T_{EOC} is extended by one round trip delay T_L. So $f_{TEOC}(T_{EOC})$ consists of $f_{TBR}(T_{BR})$ weighted with $1–P_o$ for the case of no overtaking and $f_{TBR}(T_{BR})$ shifted by T_L. and weighted with P_o for the case of overtaking:



3. In the main loop over all possible values of the sum of confirms cc for a cycle the two cases how T_{cycle} can be composed (as discussed in chapter 4.3.2) are evaluated:

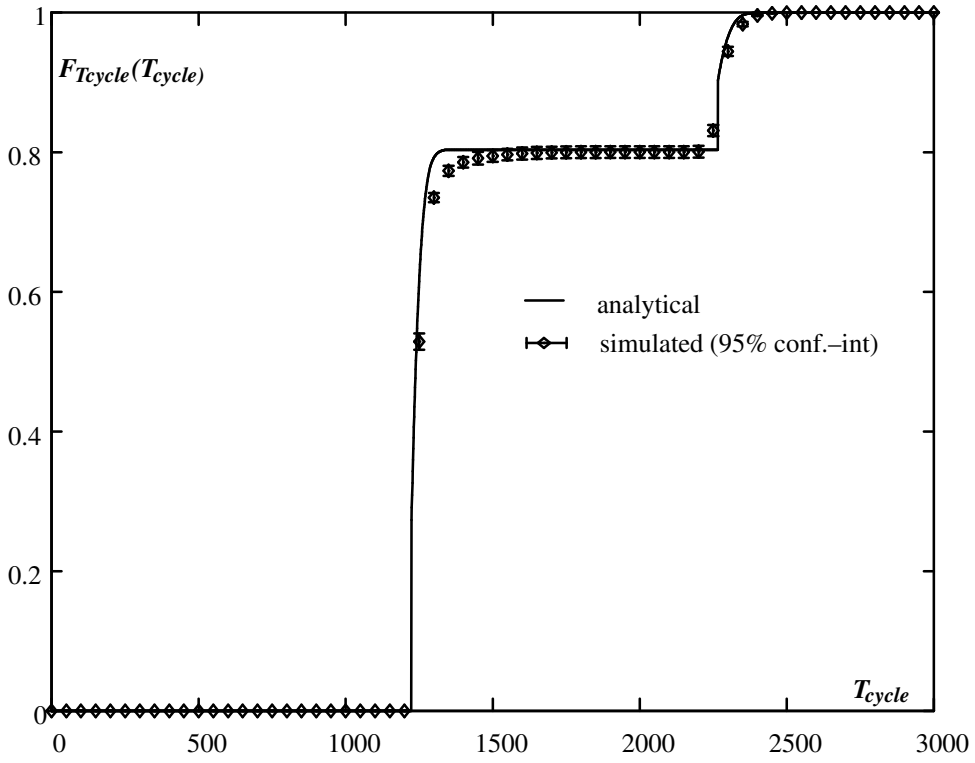
   case 1: $T_{EOC} < T_M + T_{calc}$ $\Rightarrow T_{cycle} = T_L + T_M + T_{calc}$
   case 2: $T_{EOC} > T_M + T_{calc}$ $\Rightarrow T_{cycle} = T_L + T_{EOC}$

   Since the marking time T_M normalized to T_{slot} equals the sum of confirms cc of the current cycle we get the following structure of the main loop (described here in a pseudo–C–syntax):

```
f_Tcycle [T_L] += f_cc [0];              /* no marking, overtaking and T_BR for cc = 0   */
for (cc = 1; cc <= cc_max; cc++){
        T_c+m = T_calc + cc;                          /* time for sched. calc. and marking */
        for (T_EOC =0; T_EOC < T_c+m ; T_EOC ++){     /* comparison (2 cases)              */
                f^cc_TEOC[T_c+m] += f^cc_TEOC[T_EOC];
                f^cc_TEOC[T_EOC] = 0;
                }
        for (T_EOC =0; T_EOC < T_EOCmax; T_EOC ++){ /* weighting with prob. f_cc (cc)    */
                                                    /* and shifting by basic delay T_L   */
                f_Tcycle[T_EOC + T_L] += f^cc_TEOC[T_EOC] · f_cc[cc];
                }
        }
}
```

Notation and limits:
– $f^{cc}_{TEOC}(TEOC)$ means the pdf of the random variable (rv) ***TEOC*** with the rv ***cc*** as a parameter.
– The necessary upper bound cc_{max} of the main loop can be estimated by considering the distribution $f_{cc}(cc)$ and choosing a reasonable value, e.g. $f_{cc}(cc_{max}) \leq$ 1e–3.
– The limit T_{EOCmax} of the second inner loop over T_{EOC} can be derived by the consideration that the delay in the IB of a single node cannot be greater than the length l_{IB} of the IB measured in slots. As this l_{IB} will not be greater than the maximum length of a packet to be stored within this buffer the upper bound for T_{EOCmax} is M–times l_{IB} plus T_L (the exponential decay of $f_w(w)$ and its convolutions often allow much smaller values of T_{EOCmax}).

The following figure shows a comparison of this algorithm with a simulated CRMA–II ring at a medium load:



$F_{Tcycle}(T_{cycle})$ analytical and simulated for a CRMA–II ring with L = 50 km at a system load of S = 1.0

4.3.5 A More Exact "Semi–Analytic" Way to Determine the Distribution of T_{cycle}

An exact determination of $f_{Tcycle}(T_{cycle})$ that is valid up to high values of the overall system load S has to use exact expressions for both
– the distribution of DA_{max} the max. destination address of the packets within the train of marked slots and
– the overtaking probability P_o .
As mentioned in the previous chapters both cannot be calculated with reasonable expenses in terms of calculation time in a pure analytic way.
So in this chapter it is shown how a small and very fast (for usual confidence intervals it is executed much faster than the rest of the analysis) "mini–simulation" can be embedded into the analysis to represent the otherwise nearly untractable combinatoric processes that generate the probability P_o and the distribution $f_{DAmax}(DA_{max})$.

This semi–analytical way consists of three parts:

- a time and delay oriented part using the formulas of chapter 3 to derive the basic distributions $f_c(c), f_w(w)$ etc.
- a combinatorical part that is carried out by a small simulation,
- and the concluding calculation of $f_{Tcycle}\ (T_{cycle})$.

Since here the time and delay oriented part is done analytically only a completely combinatorical simulation of the reservation based access subsystem is necessary instead of a time consuming event or process oriented simulation of the whole CRMA–II network.
This approach can be compared with some mixed mode simulation methods for digital circuits where the transient behaviour and delay of the inputs and output of the circuits is modelled by analog circuitry while the logical behaviour of the elements is described by boolean equations.

In the simulation part the combinatorical aspects of the scheduling and processing within the nodes are modelled in the following way:

For each scheduling (= simulation) cycle

- first the random values of the confirms c_i for each node are generated using the distribution $f_c(c)$ derived before in the analytical part;
- then a virtual train of cc (sum of c_i) marked sots is created using
  – the distribution $f_h(h)$ of the number of busy slots within the cc reserved slots,
  – the distribution of the packet length (measured in slots), and
  – the distribution of the destination addresses at the input of the scheduler.
  At this time the maximum destination address for this cycle is selected and entered into an array ordered by cc and h of the cycle.
- In the last step the accessing and processing of the train of reserved slots is done for each node according to the CRMA–II protocol: i.e. BR slots get FR if the actual node i is their destination and a number c_i of FR slots get BG.
  If an overtaking is detected the current cycle is stopped.
  After each cycle the event of overtaking or no overtaking is recorded into a second array from which P_o is calculated later (also ordered by cc and h).

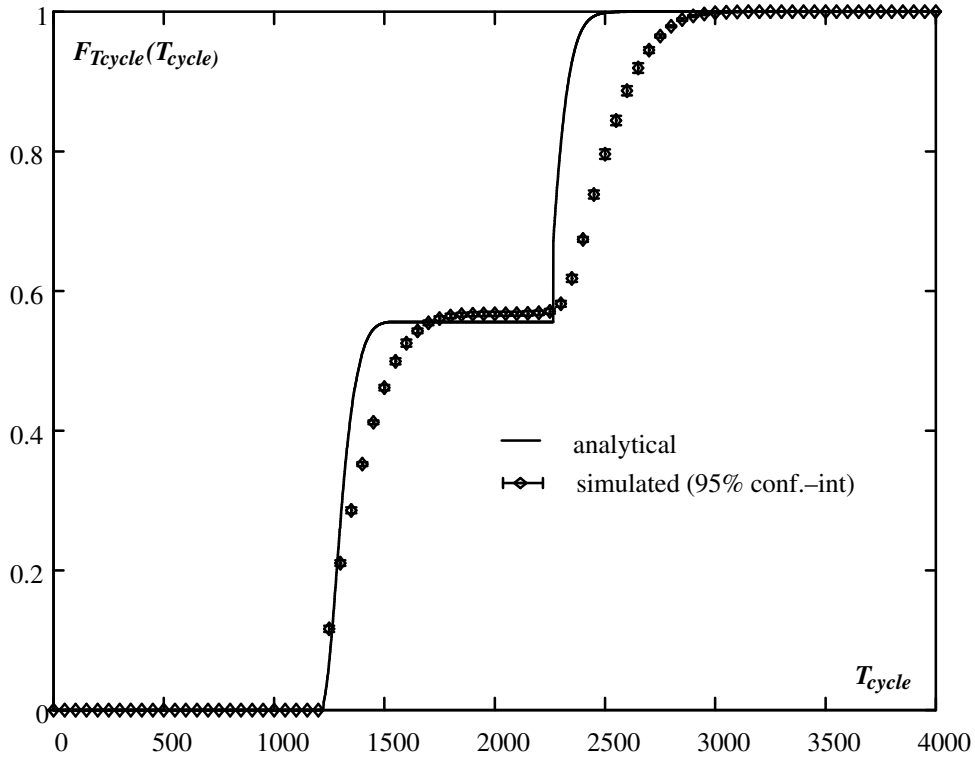
In the third part of this "semi–analytical" procedure the analytically derived expressions for the time oriented part of the CRMA–II behaviour are combined with the simulated values of $f_{DAmax}(DA_{max})$ and P_o :

- The distribution of T_{BR} is calculated using $f_w(w)$ and its convolutions, and the simulated distribution of DA_{max} as shown in chapter 4.3.3.1. Here it must be taken into account that the distribution of DA_{max} depends on the parameter h (its own distribution being dependent on the value of cc) so that for each combination of h and cc there exists a different distribution of DA_{max} .
- The distribution of the cycle length itself is then calculated as in the previous chapter using again a main loop over the values of the sum of confirms cc. But this time the dependence of T_{EOC} on h has to be taken into account, leading to an additional inner loop over h:

```
f_Tcycle [T_L] += f_cc [0];                 /* no marking, overtaking and T_BR for cc = 0      */
for (cc = 1; cc <= cc_max; cc++){
        f^cc_TEOC [0] += f^cc_h [0];        /* no overtaking and T_BR for h = 0               */
        for(h = 1 ; h <= cc;  h++){         /* determination of f^cc_TEOC                     */
                                            /* in a loop over h                               */
                for(T_BR = 0; T_BR <=  T_BRmax ; T_BR ++){
                        f^cc_TEOC[T_BR] += (1–P_o[cc,h]) · f^h_TBR[T_BR] · f^cc_h [h];  /* no overt. */
                        f^cc_TEOC[T_BR + T_L] += P_o[cc,h] · f^h_TBR[T_BR] · f^cc_h [h]; /* overt.   */
                }
        }
        T_c+m = T_calc + cc;                            /* time for sched. calc. and marking */
        for (T_EOC =0; T_EOC < T_c+m ; T_EOC ++){       /* comparison (2 cases)              */
                f^cc_TEOC[T_c+m] += f^cc_TEOC[T_EOC];
                f^cc_TEOC[T_EOC] = 0;
                }
        for (T_EOC =0; T_EOC < T_EOCmax; T_EOC ++){     /* weighting with prob. f_cc (cc)    */
                                                        /* and shifting by basic delay T_L   */
                f_Tcycle[T_EOC + T_L] += f^cc_TEOC[T_EOC] · f_cc [cc];
                }
        }
}
```

The upper limits of the loops are calculated as in the approx. determination in chapter 4.3.4 (T_{BRmax} being equal to T_{EOCmax} without the basic ring propagation delay T_L).

The following figure shows a comparison of the results of the semi–analytic approach with a simulated CRMA–II ring: this time for a high load of S = 1.7 near the limits of the system:



$f_{Tcycle}(T_{cycle})$ *semi−analytical and simulated for a CRMA−II ring with L = 50 km at S = 1.7near the throughput limit of this system*

The remaining deviations between the analysis and the simulation that show only up at higher loads are produced by the following second order effects:

- At high loads the expectation of T_{cycle} is increased by P_o and T_{BR} above the value of $T_{calc} + T_L$ that was originally assumed for the determination of $f_z(z)$. As the consequence the real expectation $E(z)$ rises above the value used for setting the threshold T in the simulation. The resulting increase of deferments reduces the usable system throughput.
- At high loads the possible system throughput is further decreased by the large number of reserved slots that result from both a high threshold $T = E(z)$ and large buffer lengths $E(b)$.

These reductions of the possible system throughput resulting an increased queueing delay of the BR slots have not yet been introduced in the analytical descriptions. This is reflected in the figure above by the fact that the analytical curve shows a higher gradient than the simulated one.

Still the semi–analytic approach reflects the internal effects of CRMA–II and their results in a very realistic manner.

4.3.6 Results and Conclusions From the Analysis

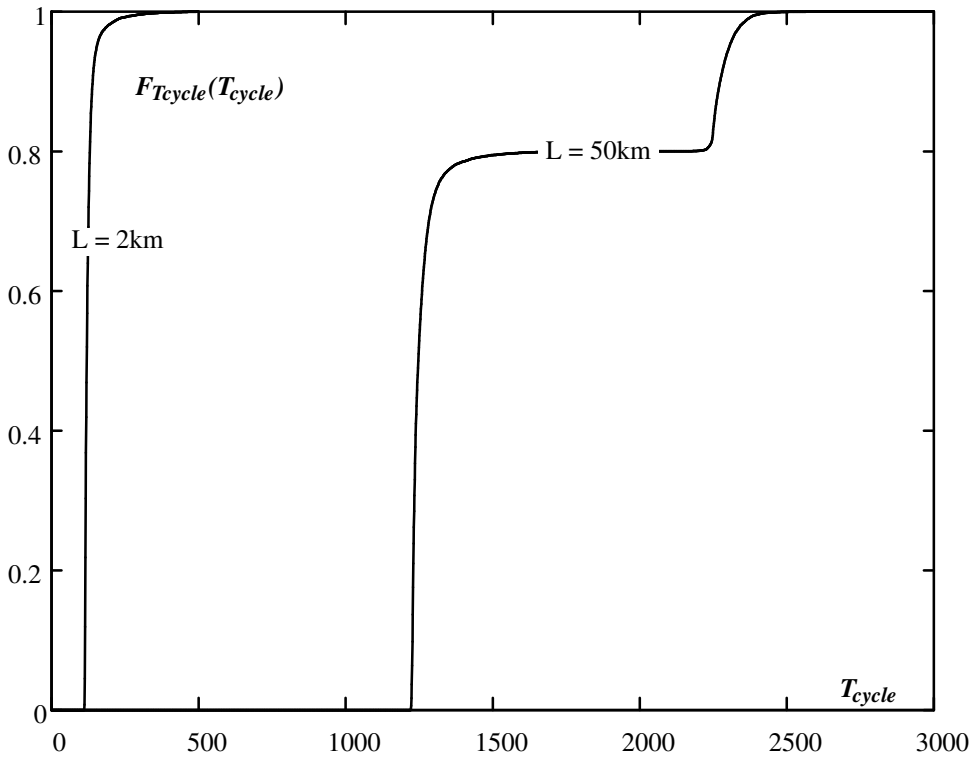
In a scenario of a ”normal user” node lying directly downstream a ”heavy user” node that constantly fills up its insertion buffer the ”normal user” node can only help himself from starvation by relying on the confirmed access granted by the scheduler via marked slots and on the deferment of the heavy user. So the duration and distribution of T_{cycle} determine the time–oriented aspects of fairness within the CRMA–II network.
Therefore it is desirable to have T_{cycle} distributed with a mean value and a variance both being as low as possible.
A closer look at the figures calculated by the formulas of the previous chapters verifies the prevalence of the above stated two main effects that influence T_{cycle}:

- The overtaking of confirmed nodes by the train of reserved slots:
  This effect leading to an increase of T_{cycle} by the round trip delay time T_L shows up already at low to medium loads ($P_o \approx 20\%$ for a ring of 20 nodes at S=1)
- The enqueueing of BR slots in the insertion buffers of intermediate nodes between the scheduler and their destination node:
  the resulting component T_{BR} of T_{cycle} shows an exponential decay that extends up to a far away limit only imposed by the maximum IB length of the nodes (corresponding to M times the max. packet length).

Depending on the physical size of the CRMA–II ring this leads to the following relation between the both undesirable results:

- In small rings the exponential decay caused by T_{BR} outweighes and hides the results of the overtaking.
- In large rings T_{cycle} is mainly extended by the overtaking and the resulting additional ring latency T_L so that here also an increased variance of T_{cycle} is to be expected.

Since overtaking only becomes visible if the additional delay is greater than T_{calc} a round trip delay of $T_L = T_{calc}$ is the limit between the two cases.



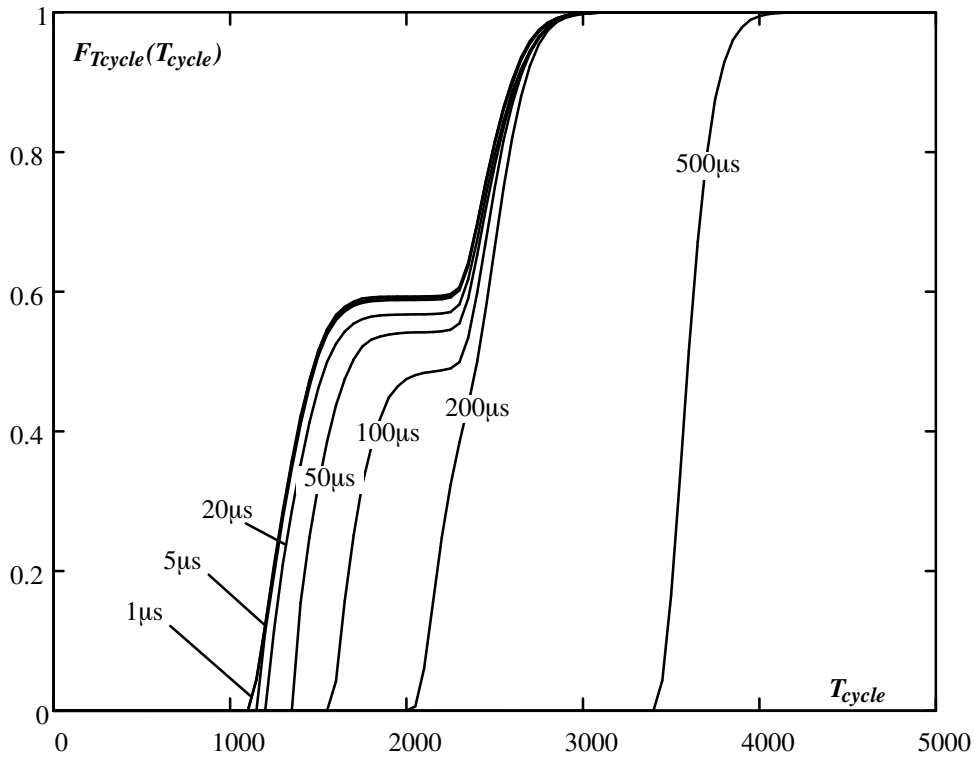
simulated $F_{Tcycle}(T_{cycle})$ for CRMA–II rings with L = 2km and L = 50km at S = 1.0

Of the possible solutions for these problems two alternatives are most interesting:

- Usage of a fast hardware based implementation of he scheduler:
  By building the scheduler in hardware the calculation time T_{calc} of the scheduling process can be reduced from the order of tenths of microseconds (for solutions employing fast standard microprocessors) to a few microseconds [Lem 93]. This in turn has a direct positive influence on the minimum values of T_{cycle}.
  But since neither the overtaking nor the enqueueing part of T_{BR} are addressed by this measure no basic improvements are to be expected (s. ch. 4.3.7 for details)
- A slight modification of the CRMA–II protocol that allows marking of only free slots is promising to be more useful (s. ch. 4.3.8 for details).

4.3.7 Necessary Scheduling Speed

The following figure shows the simulated distribution of the cycle duration for schedulers with different calculation time T_{calc}. Considering the components and effects that constitute T_{cycle} it becomes clear that decreasing T_{calc} below the expected delay $E(T_{BR})$ of the BR slots in the IBs has a neglectable effect on the shape of $F_{Tcycle}(T_{cycle})$. And as long as T_{calc} is below the order of T_L the overtaking reduces significantly the gained reduction (by using a faster scheduler) of T_{cycle}.



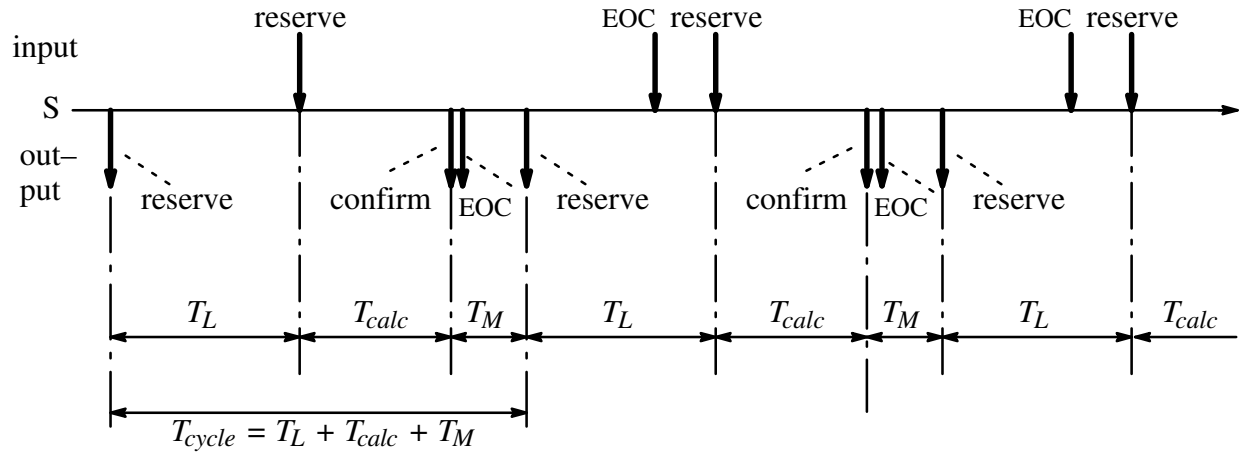
$F_{Tcycle}(T_{cycle})$ for a CRMA−II ring with a length of L = 50km (corresponding to T_L = 250µs) for different values of T_{calc} at S = 1.7.

The consequence is that only for short rings a hardware implementation of the scheduler is reasonable while for MANs the scheduler can be implemented using one of the faster standard microprocessors.

4.3.8 Modification of CRMA–II to Reduce the Duration of the Scheduling Cycles

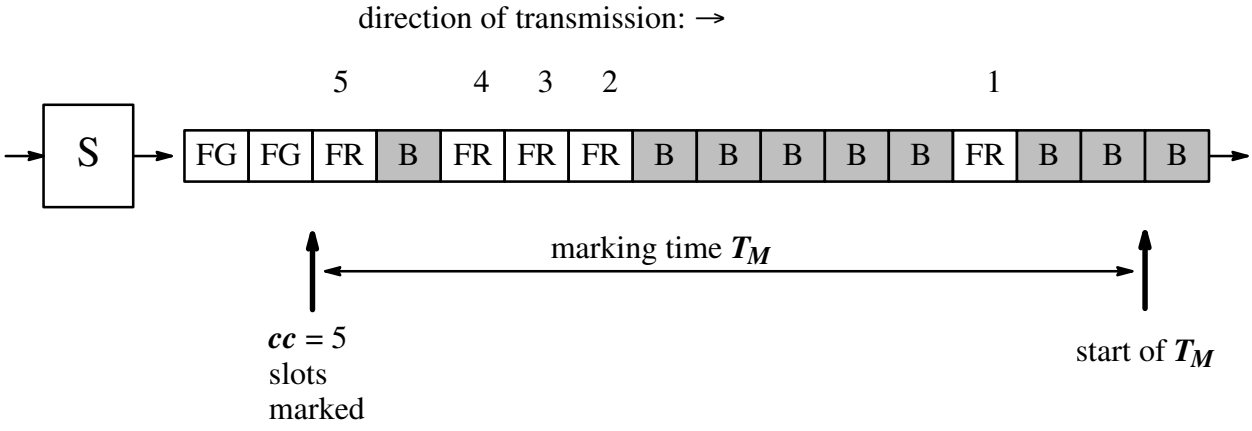
Since both overtaking and enqueueing are caused by the presence of busy reserved slots it would be desirable to allow only free reserved slots to get marked at the scheduler.

The resulting scheduling cycles then always have the same structure:

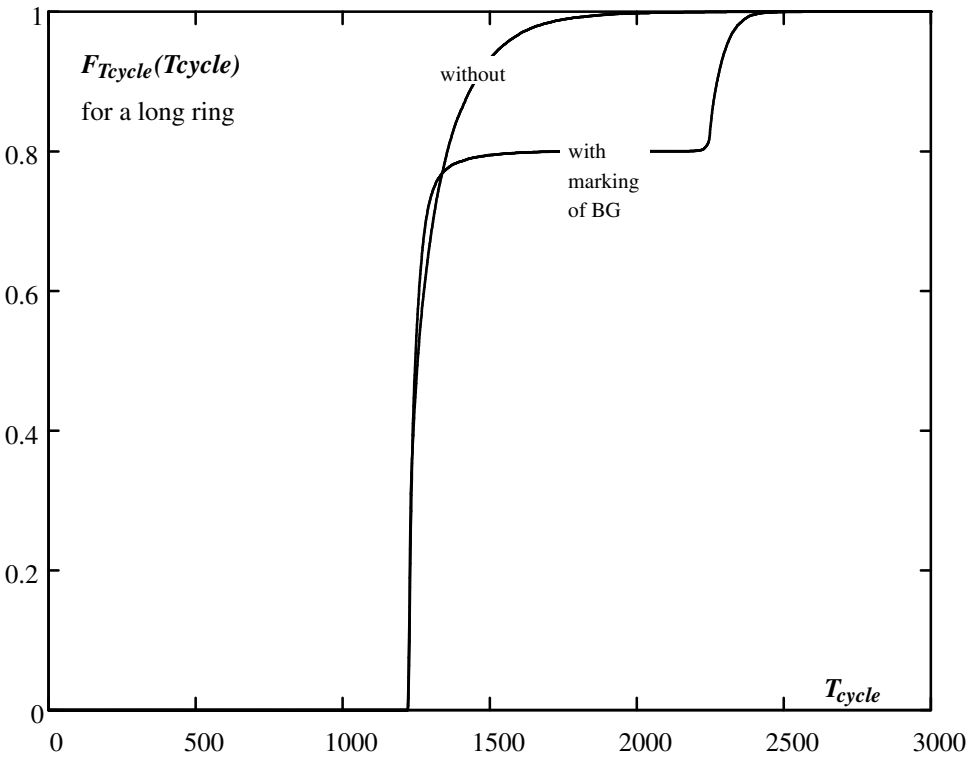


Thus the duration of T_{cycle} is nearly fixed, the only random variable being the marking time T_M.

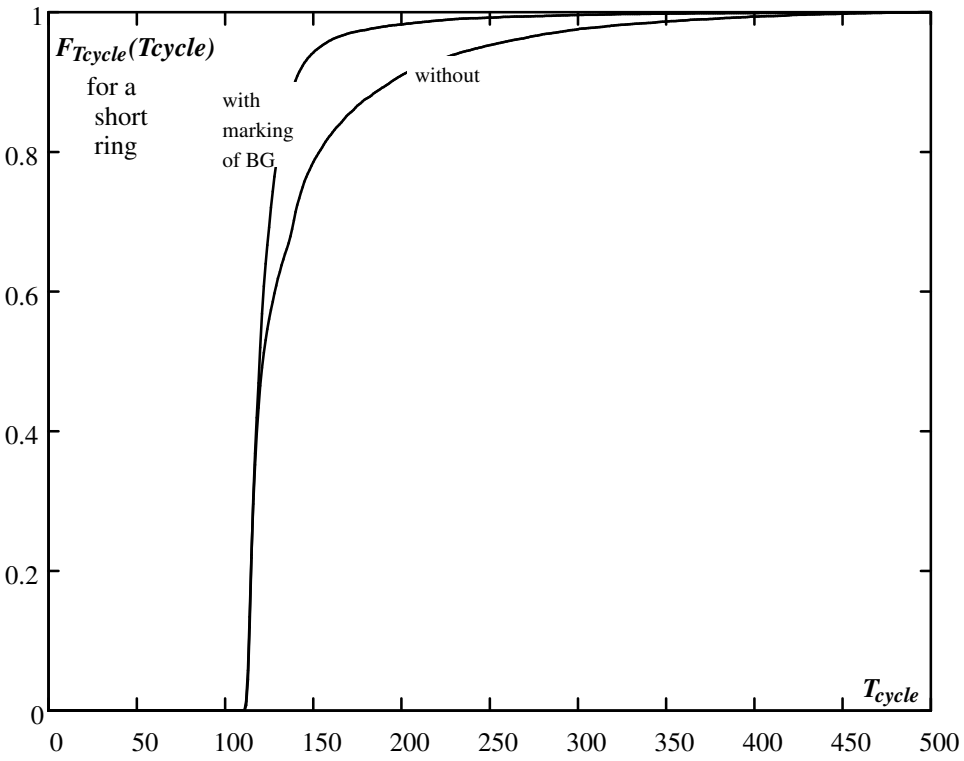
As an expense for the now gained simplicity and speed of the cycles an increased marking time has to be taken into account because during the marking of cc free slots a certain number of busy slots (depending on the system load S) also pass the scheduler.
E.g. for the case of $cc = 5$ the slots at the output of the scheduler S may look like:



Therefore it can be expected that for short CRMA–II rings and high loads the extended distribution of T_M outweighes the positive effect (on the distribution of T_{cycle}) gained by eliminating of the overtaking. This is shown in the following two figures:



PDF of T_{cycle} with and without the marking of BG slots for L = 50km at S=1.0

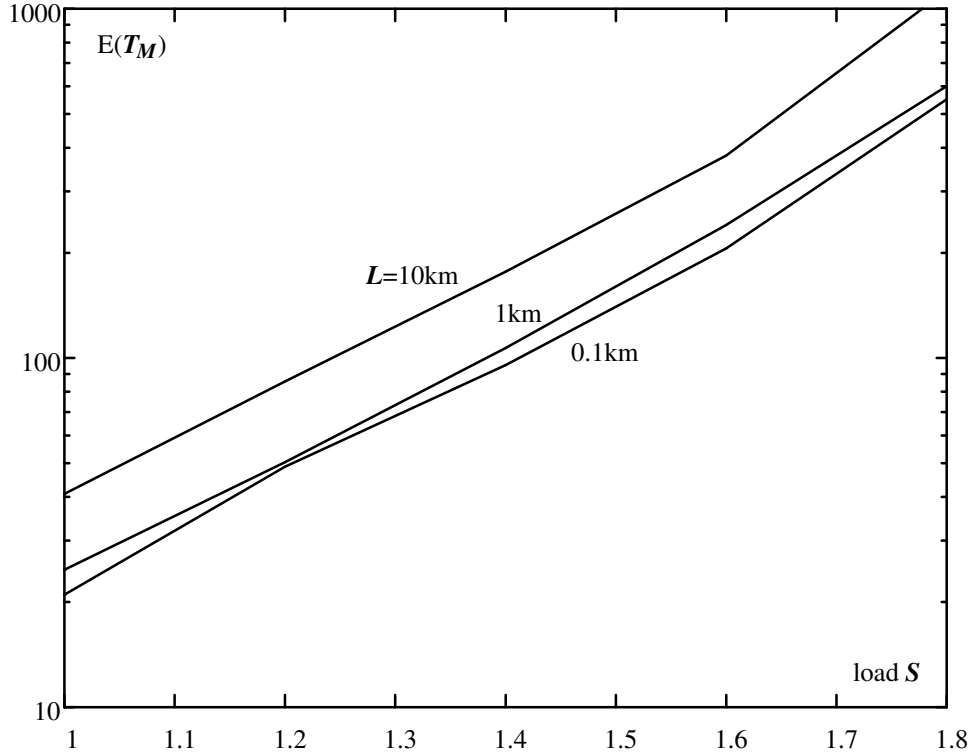


PDF of T_{cycle} with and without the marking of BG slots for L = 2km at S=1.0

In the original CRMA–II the delay of the BR slots and the overtaking extend the duration T_{cycle} of the scheduling cycles. This increase will be in the order of the round trip delay T_L.
In the modified CRMA–II when only FG slots get marked the increased marking time extends T_{cycle} at high loads.
So the modification of the protocol makes only sense if the increased marking time T_M of the modified protocol is less than T_L.

The correct decision between the two protocol alternatives can either be taken in advance by the network management using the formulas of the previous chapters (expecting an overall system load in a certain range) or an adaptive enhancement of the scheduler can be used:

1) At a given ring length L the critical overall system load S_c from that on it is preferable to mark also BG slots can be estimated using the following considerations:
When only FG slots get marked the resulting marking time T_M strongly depends on the load S and is only slightly influenced by the round trip delay T_L . The reasons are:



Expectation of T_M over load S for several ring lengths L (modif. protocol: only FG get marked)

– the absence of markable FG slots extends the marking time:

$$E(T_M) = E(cc) \cdot \frac{1}{1 - S/2} = E(c) \cdot M \cdot \frac{1}{1 - S/2}$$

– an increased load S leads to a higher optimal threshold $T_{opt} \approx E(z)$ that in turn increases the marking time since $E(b)$ also rises with S.
– an increased T_L at a given load and the resulting higher T_{cycle} also increase $E(z)$ and T_{opt} but $E(c)$ only rises slightly because it is limited by $E(b)$.
So S_c an be estimated by:

$$T_M \;=\; E\big(c\big(S, T_L, T..\big)\big) \cdot M \cdot \frac{1}{1 - S/2} \stackrel{!}{=} T_L$$

2) The alternative approach is to use an adaptive mechanism within the scheduler that decides for each cycle whether to mark also BG slots:
At the start of the whole system the scheduler first derives the round trip delay $\boldsymbol{T_L}$ from the re–arrival time of its reserve command.
This value (measured in units of the transmission time of a slot) is then loaded at the beginning of the marking phase into an additional marking time counter MTC and the number of slots to mark is loaded into the marking counter MC.
Then each time then a slot arrives at the scheduler the MTC is decremented by one. And as long as MTC $\geq$ 0 this slot only gets marked if it is an FG slot (with the MC being decremented correspondingly). When MTC reaches zero also the following BG slots get marked (until MC is zero).
So this protocol enhancement automatically leads to scheduling cycles with a minimum duration without the need for modifying the access mechanisms in the CRMA–II nodes.
Note: the scheduler itself has to have the necessary decision speed to distinguish between the different slot types even yet at the unmodified protocol since during the marking phase (at high loads and scheduling thresholds) also overtaking FR slots can reach its input if the number of marked slots gets greater than the round trip delay measured in slots.

4.3.9 Dual Ring Networks

The equations of chapter 3 apply to CRMA–II dual ring networks as well. But since in the case of two counter rotating rings the packets are always routed on the shortest way to their destination a different distribution $f_{DA}(DA)$ of the destination addresses at the scheduler has to be used when the overtaking probability and the queueing delay of the BR slots are calculated:

E.g. for an odd number of nodes M the scheduler on the one ring with a transmission direction corresponding to a rising node index notices only the following source destination combinations:

packets from node number:	destined to node number:
M	1, 2, 3, ..., (M−1)/2−1, (M−1)/2
M−1	1, 2, 3, ..., (M−1)/2−1
...	...
(M+1)/2+1	1

So we have altogether

$$\frac{(\frac{M+1}{2} - 1)\frac{M+1}{2}}{2}$$

source–destination address combinations. And for the destination address DA we have a share of $(M+1)/2-DA$ moving through the scheduler so that we get:

$$f_{DA}(DA) = 4\frac{M + 1 - 2DA}{M^2 - 1} \qquad for\ DA = 1\ ..\ (M - 1)/2$$

For an even number of nodes M we get:

$$f_{DA}(DA) = 4\frac{M + 1 - 2DA}{M^2} \qquad for\ DA = 1\ ..\ M/2$$

Using the same considerations as in chapter 4.3.3.2 we get now for the probability of overtaking in the case of only one marked slot (that is BR):

$$P_o(1,1) = 4\sum_{DA=1}^{(M-1)/2}\frac{(M + 1 - 2DA)(DA - 1)}{M^2(M - 1)} \qquad for\ an\ odd\ M$$

and:

$$P_o(1,1) = 4\sum_{DA=1}^{M/2}\frac{(M + 1 - 2DA)(DA - 1)}{M^3} \qquad for\ an\ even\ M$$

So that for the case of a dual (and counter rotating) ring network we get a reduced overtaking probability ($f_{cc}(cc)$ and $f_h(h)$ having the the same distributions as in the single ring case if the small feedback influence of the reduced overtaking and enqueueing on T_{cycle} is neglected). E.g. for M=20 we have the following probabilities for cc=h=1 :

– single ring: $P_o(1,1) = 0.3$
– double ring: $P_o(1,1) = 0.1425$

Since the maximum destination address of busy slots at the scheduler is no longer $M-1$ (as in the case of a single ring) but $(M-1)/2$ for an odd M resp. $M/2$ for an even M the enqueueing delay T_{BR} of BR slots in the IBs is also reduced in a dual ring topology.

So dual ring networks offer (at nearly doubled hardware expenses) not only a higher system throughput (up to 8 times the link capacity compared to 2 times for a single ring) but also show a better behaviour concerning the duration of the scheduling cycles.

4.4 Further Applications

The mathematics of chapter 3 and 4 could be extended further to cover also the following topics:

- **Investigation of asymmetric load scenarios:**
  Up to now only balanced CRMA–II rings (i.e. each node transmitting the same average load onto the ring) have been considered. But the above presented formulas and methods allow also the analysis of scenarios where one or more nodes within the network transmit considerably more (or less) than the rest. The same is true to situations where the destination addresses of the packets are no longer uniformly distributed (e.g. some nodes within the ring communicate more with each other than with the rest, or client server arrangements). The reason for the tractability of these scenarios with the presented methods is the separation of the interesting distributions into their main components down to the level of single nodes and their buffers. After the determination of the corresponding loads at these buffers the resulting waiting times and queue lengths are then combined via convolution and weighted addition. This detailed way allows therefore to treat also the asymmetric scenarios simply by calculating the corresponding loads of the buffers.

- **Evaluation of the improvements (and costs) induced by the introduction of bypassing:**
  In earlier version of CRMA–II as presented in June 1991 [v.As 91] the nodes treated arriving scheduler command frames and not usable reserved resp. deferred slots by enqueueing them into their insertion buffers (IBs) like ”normal” busy slots. These slots increased the load of the IBs and the corresponding waiting times although their presence in the IB was not of any use for any of the other nodes. The result was a suboptimal delay / throughput behaviour of the network [Zin 92].
  Therefore the improved version of CRMA–II [v. As 92] allowed the nodes to bypass these slots around their IBs (scheduler commands on slot boundaries and the other unusable slots on frame boundaries). But these measures on the one hand further complicate the protocol and on the other hand also have their own negative effects on its performance: no longer transmission of packets in only contiguous slots and the resulting need for a more expensive reassembly at the receiving node because of bypassing on slot boundaries; even bypassed slots occupy the outgoing link of the node thus reduce its transmission capacity.
  In this case the analytical methods can be applied to address the question of costs vs. improvements and to assess under which circumstances what kind of bypassing makes the most sense.

These and the investigation of further improvements of the CRMA–II protocol (such as individual marking be all the nodes in contrast to the actual marking that is done by the scheduler alone) are subjects of further study.

List of Symbols (in lexical order):

a	probability for the transition b → b–1 during T_{slot}
b, $f_b(b)$	number of slots in TB and IB and its pdf
c, $f_c(c)$	number of confirmed slots per node and cycle and its pdf
c_i	confirms of node i
cc, $f_{cc}(cc)$	sum of confirms per cycle and its distribution
d, $f_d(d)$	number of deferred slots per node and cycle and its pdf
e, $f_e(e)$, $F_e^*(s)$	waiting time in the AB and its pdf and $\mathcal{L}$–transform
f, $f_f(f)$, $F_f^*(s)$	waiting time in the ring branch of an BI node and its pdf and $\mathcal{L}$–transf.
g, $f_g(g)$, G	number of packets arriving into TB during a cycle, pdf and expectat.
h, $f_h(h)$	number of BR slots within the cc marked slots
k, $f_k(k)$	number of packet arrivals into TB during T_{slot} and its pdf
m, $f_m(m)$	number of packets in a virtual M/M/1 queueing system and its pdf
s, $f_s(s)$	packet length (in full slots) and its pdf
w, $f_w(w)$, $F_w^*(s)$	waiting time in the IB and its pdf and $\mathcal{L}$–transform
w_t, $f_{wt}(w_t)$, $F_{wt}(w_t)$	access delay of a TB packet and its pdf and PDF
x, $f_x(x)$, $\overline{x}$	packet length (normalized to T_{slot}) and its pdf and mean
z_i	probability for the transition b → b+i during T_{slot}
z, $f_z(z)$	number of slots transmitted per node and cycle and its pdf
$B^*(s)$	$\mathcal{L}$–transform of the pdf of the service time
DA, $f_{DA}(DA)$	destination address of a busy slot at the scheduler input and its pdf
DA_{max}, $f_{DAmax}(DA_{max})$	max. dest. addr. of a busy slot at the scheduler input and its pdf
E(A)	expectation of random variable A
H	desired sum of confirms per cycle
L	ring length
M	number of nodes in the CRMA–II ring
P_o	probability of overtaking during a cycle
R	link bit rate
S	system load
T, T_{opt}	scheduling threshold and its optimum
T_{BR}, $f_{TBR}(T_{BR})$	waiting time of a BR slot in IBs between source and dest.
T_{calc}	duration of the fairness calculations
T_{cycle}, f_{Tcycle}, F_{Tcycle}	duration of the scheduling cycles and its pdf and PDF
T_{EOC}	time needed by the EOC signal to move around the ring (excl. T_L)
T_L	round trip delay
T_M, $f_{TM}(T_M)$	marking time (measured in T_{slot}) and its pdf
T_{slot}	transmission time of a slot
$\delta(A)$	Kronecker delta function for variable A
λ_r	arrival rate into the IB of a node
λ_t	arrival rate into the TB of a node
π	probability of a packet arriving into TB during T_{slot}
ϱ_r	load into the IB of a node
ϱ_t	load into the IB of a node
ϱ	load on a link
σ	probability for the arrival of a slot into IB during T_{slot}

Abbreviations and Acronyms:

pdf	probability density function
AB	additional buffer
AS	additional server
ATM	Asynchronous Transfer Mode
BG	busy gratis slot
BI	Buffer Insertion
BR	busy reserved slot
EOC	end of cycle
FG	free gratis slot
FR	free reserved slot
IB	insertion buffer
LAN	Local Area Network
MAC	Medium Access Control
MAN	Metropolitan Area Network
MC	marking counter
MTC	marking time counter
PDF	probability distribution function
RB	receive buffer
RS	receive switch
TB	transmit buffer
TS	transmit switch

References

[v. As 91] v. As, H.R. / Lemppenau, W.W. / Zafiropulo, P. / Zurfluh, E.:
CRMA–II: A Gbit/s MAC Protocol for Ring and Bus Networks With Immediate Access Capability,
Proc. of EFOC/LAN 91, London, June 19–21, 1991

[v. As 92] v. As, H.R. / Lemppenau, W.W. / Zafiropulo, P.:
Performance of CRMA–II: A Reservation–Based Fair Media Access Protocol for Gbit/s LANs and MANs,
Proc. of EFOC/LAN 92, Paris, June 24–26, 1992

[Bac 93] Bach, C. Grebe, A.:
Comparison and performance evaluation of CRMA–II and ATMR,
Proc. of EFOC&N 93, The Hague, June 30 – July 2, 1993

[Bux 83] Bux, W. / Schlatter, M.:
An Approximate Method for the Performance Analysis of Buffer Insertion Rings,
IEEE Transactions on Communications, Vol. COM–31, No. 1, Jan. 1983, p. 50–55

[Geh 94] Gehring, H. et al.:
An approach for delay–sensitive services in CRMA–II,
Proc. of EFOC&N 94, Heidelberg, Germany, June 22 – 24, 1994

[Ham 86] Hammond, J.L. / O'Reilly, P.J.P:
Performance Analysis of Local Computer Networks,
Addison–Wesley, 1986

[Kl I 75] Kleinrock, L.:
Queueing Systems, Vol. I: Theory,
J. Wiley & Sons, 1975

[Kl II 76] Kleinrock, L.:
Queueing Systems, Vol. II: Computer Applications,
J. Wiley & Sons, 1976

[Lem 93] Lemppenau, W.W. / v. As, H.R. / Schindler, H.R.:
ATM Implementation of the CRMA–II Dual Ring LAN and MAN,
Proc. of EFOC&N 93, The Hague, June 30 – July 2, 1993

[Mar 92] Marsan, M. A. et al.:
Performance of the CRMA–II protocol: a simulation study,
Proc. of EFOC/LAN 92, Paris, June 24–26, 1992

[Meu 93] Meuser, T.:
Performance evaluation of the CRMA–II Protocol in Gbit/s LANs,
Proc. of EFOC&N 93, The Hague, June 30 – July 2, 1993

[Meu 94] Meuser, T.:
Enhancement of the CRMA–II protocol by distributed marking,
Proc. of EFOC&N 94, Heidelberg, Germany, June 22 – 24, 1994

[Pap 84] Papoulis, A.:
Probability, Random Variables and Stochastic Processes,
McGraw–Hill, 1984

[Zin 92] Zinsinger, N.:
Analytische und Simulative Untersuchung des CRMA–II Protokolls,
Diploma Thesis at Lehrstuhl für Kommunikationsnetze, Technical University Munich, Munich, July 1992