

Idealerweise parametrisieren Sie die Programme so, dass über die Kommandozeile die relevanten Parameter eingestellt werden können (welche sind das?).

- Port(4711)
- Buffer-Größe
- Anzahl Pakete

Java-Programme starten:

-java RX default: java RX 4711 1024, wobei 4711 der Port und 1024 die Buffer-Größe ist.

-java TX default: java TX 1000 0, wobei 1000 die Anzahl der Pakete ist.

C-Programme starten:

-. /RX default: ./RX 4711 512, wobei 4711 der Port und 512 die Buffer-Größe ist.

-. /TX default: ./TX 4711 100 0, wobei 4711 der Port ist und 100 die Anzahl der Pakete.

## Messungen

TX.java->RX.c

41	Java to C						
42	# test	# packets	buffer size	delay		Durchn. Packets lost	
43	10	100	1024	0		0	0,00
44	10	1000	1024	0		0	0,00
45	10	10000	1024	0		646,1	6,46
46							
47	10	100	1024	100		0	0,00
48	10	1000	1024	100		0	0,00
49	10	10000	1024	100		0	0,00
50							

## TX.java->RX.java

51	Java to Java						
52	# test	# packets	buffer size	delay		Durchn. Packets lost	
53	10	100	1024	0		0	0,00
54	10	1000	1024	0		269,9	26,99
55	10	10000	1024	0		6906,1	69,06
56							
57	10	100	1024	100		0	0,00
58	10	1000	1024	100		0	0,00
59	10	10000	1024	100		1807,1	18,07
60							
61	10	100	1024	500		0	0,00
62	10	1000	1024	500		0	0,00
63	10	10000	1024	500		1777,9	17,78
64							
65	10	100	1024	1000		0	0,00
66	10	1000	1024	1000		0	0,00
67	10	10000	1024	1000		1591,8	15,92
68							
69	10	100	1024	2000		0	0,00
70	10	1000	1024	2000		0	0,00
71	10	10000	1024	2000		0	0,00

## TX.c->RX.c

	A	B	C	D	E	F	G
1	C to C						
2	# test	# packets	buffer size	delay		Durchn. Packets lost	% of lost packets
3	10	100	1024	0		0	0,00
4	10	1000	1024	0		8,2	0,82
5	10	10000	1024	0		2154,3	21,54
6							
7	10	100	1024	100		0	0,00
8	10	1000	1024	100		6,9	0,69
9	10	10000	1024	100		2086,6	20,87
10							
11	10	100	1024	300		0	0,00
12	10	1000	1024	300		0	0,00
13	10	10000	1024	300		1426,3	14,26
14							
15	10	100	1024	500		0	0,00
16	10	1000	1024	500		0	0,00
17	10	10000	1024	500		0	0,00
18							

## TX.c->RX.java

19	C to Java						
20	# test	# packets	buffer size	delay		Durchn. Packets lost	
21	10	100	1024	0		0	0,00
22	10	1000	1024	0		618,3	61,83
23	10	10000	1024	0		8629,2	86,29
24							
25	10	100	1024	100		0	0,00
26	10	1000	1024	100		600	60,00
27	10	10000	1024	100		8455,8	84,56
28							
29	10	100	1024	500		0	0,00
30	10	1000	1024	500		367,8	36,78
31	10	10000	1024	500		6193,3	61,93
32							
33	10	100	1024	1000		0	0,00
34	10	1000	1024	1000		50,5	5,05
35	10	10000	1024	1000		3103,4	31,03
36							

### Folgerungen:

Über alle Test hinweg lässt sich eindeutig erkennen das der Delay zwischen den Paketen zu insgesamt zu weitaus weniger Paketverlusten führt. Mittels vergrößerung des Delays lässt sich bei allen Tests, außer C zu Java, der Paketverlust bei 100,1000 und auch 10000 Paketen auf bis zu 0% verringern.

Als schlechtester Fall sticht C->Java hervor, selbst mit einem Delay von 1000 findet trotzdem noch ein Paketverlust von 31% der 10000 Pakete, dies liegt daran das C viel schneller die Pakete produziert als Java sie konsumieren kann. Ohne Delay kommt man sogar auf einen Verlust von ganzen 86% der 10000 Pakete womit fast keine erkennbare Datei mehr ankommen kann. Bei ganz wenig Paketen kommt es noch zu keinem allzu großen Overflow bei Java weshalb keine Pakete verloren gehen.

Als bester Fall lässt sich Java->C erkennen, da bereits bei einem Delay von 100 keine Pakete mehr verloren gehen. Wie vorhin erwähnt arbeitet Java langsamer als C, daher kommt es zu wenig Paketverlusten da kein Overflow beim RX.c entsteht was zu einem geringem Paketverlust führt.



