6 Ergebnisse

Um *Mcntp* beurteilen zu können, empfiehlt es sich, *Mcntp* mit den Vorgaben aus Kapitel 3 zu vergleichen:

- Plattformunabhängigkeit
- 8-Bit-Clean
- Sparsamkeit bezüglich Bandbreite
- Ausfalltoleranz
- Transportsicherheit
- Benötigte Rechenzeit

6.1 Messungen und Tests

Alle Messungen wurden auf einem Teil des Backbones des Internetproviders Xlink in Karlsruhe in Zusammenarbeit mit den Partnerfirmen vor Ort (POPs) durchgeführt. Die untere Karte zeigt diesen Teil mit den entsprechenden Serverrechnern und Routern. Bei diesen Tests kam *Mcntp* auf unterschiedlichen Plattformen zum Einsatz.

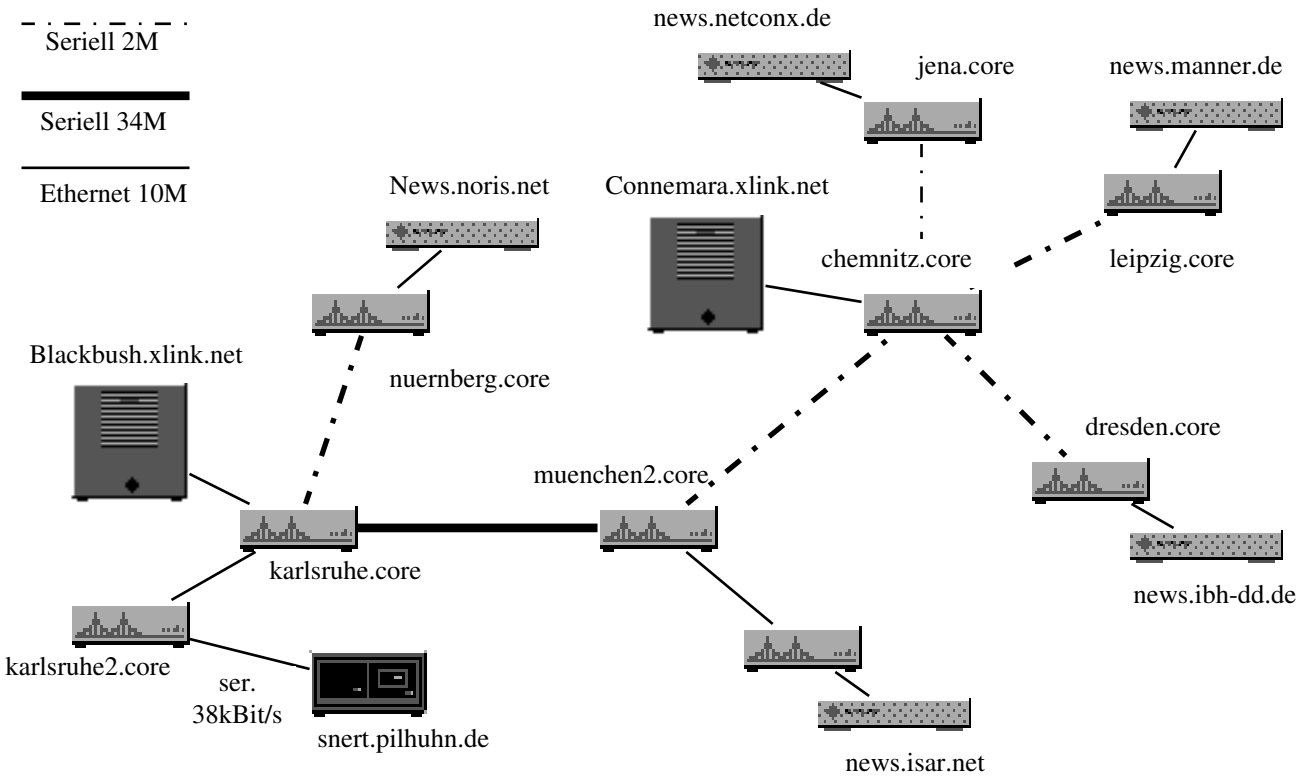

Abbildung 44: Versuchsbackbone

6.2 Plattformunabhängigkeit

Mcntp wurde auf die in Abschnitt 4.8 erwähnten Betriebssysteme bereits portiert. Da keine Funktionen verwendet werden, die für diese Betriebssysteme speziell sind, ist zu erwarten, daß auch Portierungen auf andere Betriebssysteme problemlos sind.

Mcxmit erwartet seine Eingabe über den Standardeingabekanal. Wenn nun der NTA nicht in der Lage ist, diesen selbst zu betreiben (*Channel-Feed* siehe 5.2.1), kann der NTA eine Datei mit den Pfadangaben und Message-IDs schreiben, die mcxmit dann über Umleitung der Standardeingabe zur Bearbeitung erhält. Dieser Mechanismus ist nicht auf das Unix Betriebssystem beschränkt, sondern funktioniert beispielsweise auch unter Windows NT[48] oder VMS.

Da mcrcv dem Empfänger NTA die Artikel via NNTP übermittelt, muß mcrcv nicht auf dem gleichen Rechner laufen, wie der Newsserver. Die Möglichkeit, Batches (siehe 4.5.2) zu schreiben, bietet die Möglichkeit, den Empfänger selbst auf Newssystemen zu betreiben, die keine NNTP-Unterstützung bieten[49].

Die einzige Bedingung für den Betrieb vom *Mcntp* , die erfüllt sein muß ist, daß IP Multicast auf den Rechnern unterstützt wird, auf denen mcxmit, mcrcv und yawgmoth laufen sollen. Diese Bedingung ist aber bei allen modernen Betriebssystemen erfüllt.

6.3 8-Bit clean

Mcntp ist 8-Bit clean. Dies bedeutet, daß keine Zeichen beim Transport verändert werden. Die verschickten Artikel kommen also beim Empfänger an, wie sie auf dem Senderhost verschickt wurden. Da bereits die Funktionenbibliothek *libmc* 8-Bit clean ist, war es auf Applikationsebene nicht nötig, manche Zeichen für den Transport umzukodieren.

6.4 Bandbreitenersparnis

Für den Betrieb im Backbone wurden mehrere *mcxmit* Prozesse generiert, die auf dem Newsserver *Blackbush-.xlink.net* die Artikel verteilen (siehe auch Abschnitt5.2.1). Zusätzlich wurden zwischen *Blackbush.xlink.net* und *Connemara.xlink.net*, sowie zwischen *Connemara.xlink.net* und den *news.ibh-dd.de* verzögerte NNTP-Feeds eingerichtet, um eventuell verlorengegangene Artikel nachzuliefern. Die verteilten Newsgruppen waren: Big-8[50] ohne comp.binaries.* und de.*.

6.4.1 Einfluß von Paketverlusten

Während mittels *Mcntp* nur Artikel verteilt wurden, die kleiner als 30.000 Bytes groß sind, wurden über den NNTP-Feed auch Artikel transportiert, die größer sind. Sowohl zwischen den Servern *Blackbush* und *Connemara*, als auch zwischen *Connemara* und *news.ibh-dd.de*, zeigte sich, daß die Anzahl der Artikel, die via NNTP transportiert wurden, im Bereich von 1% der Gesamtmenge lagen. Dies zeigt, daß die Annahme, daß nur mit wenig Paketverlust zu rechnen ist, sich bestätigt.

[48] Für Windows NT kann es sein, daß dazu die GNU-Win32 Werkzeuge von http://www.cygnus.com/misc/gnu-win32/ benötigt werden.
[49] C News beispielsweise wird ohne NNTP-Unterstützung verteilt.
[50] comp.*, news.*,rec.*,sci.*,soc.*,humanities.*,misc.*,talk.*

Um herauszufinden, wie groß der Paketverlust zwischen *Blackbush.xlink.net* und *Connemara.xlink.net* in der Praxis ist, wurde der verzögerte NNTP Feed von Karlsruhe nach Chemnitz auf exakt die gleiche Artikelmenge eingestellt, wobei der NNTP-Feed mit fünf Sekunden Verzögerung betrieben wurde. Hierbei zeigte sich folgendes deutliches Bild:

Artikelgröße	Artikel verschickt	Artikel via *Mcntp* erhalten	Artikel nachgeliefert	Paketverlust
<30.000 Bytes	12.000	11.992	8	0.6 Promille
<60.000 Bytes	28.000	27.936	64	2.2 Promille

Tabelle 10: Artikelverluste in der Praxis bei unterschiedlichen Artikelgrößen

Es zeigt sich also, daß im praktischen Gebrauch der Artikelverlust unter einem Prozent[51] und damit die Voraussetzung gegeben ist, *Mcntp* effizient einzusetzen. Der etwas höhere Verlust bei größeren Artikeln läßt sich damit erklären, daß zum einen hier die Wahrscheinlichkeit höher ist, daß einzelne Fragmente des Pakets verloren gehen, und zum anderen damit, daß die Netzwerkpuffer im Kern des Zielsystems nicht groß genug waren, um mehrere große Pakete, die nacheinander ankamen, zu puffern.

Nebenbei: da *Mcntp* die herkömmliche Verteilung über NNTP nicht ersetzt, sondern ergänzt, können Artikel, die via *Mcntp* nicht zugestellt wurden, über NNTP nachgeliefert werden. Es gehen also insgesamt keine Artikel verloren.

6.4.2 Datenkompression

Wie in Abschnitt 4.4 beschrieben, ist mcxmit in der Lage, die Artikel vor dem Versand zu komprimieren. Die Auswertung der von mcxmit generierten Statistikdaten führt dabei zu folgenden Bild:

		Ohne Kompression	Mit Kompression
Anzahl versendete Artikel		33044	1037905
Artikeldaten in Bytes	Bytes	68105941	2260938548
Verschickt total	Bytes	71372945	1349773580
Differenz	Bytes	-3267004	911164968
Durchschn. Artikelgröße	Bytes	2061	2178
Durchschn. verschickt (pro Art.)	Bytes	2159	1300
Kompressionsfaktor		1	1.7

Tabelle 11: Einfluß von Datenkompression auf die Artikelgrößen

Durch die Kompression der Artikel vor dem Versand kann also das übertragene Volumen auf ca. 60 % des Originalvolumens reduziert werden – wie sich dies auf die zur Übertragung notwendige Datenrate auswirkt, zeigt die nächste Tabelle :

Die Anzahl der Artikel entspricht dabei mit 100.000 der Anzahl derer, die aktuell[52] am Tag via *Mcntp* verteilt werden.

Mit diesen Daten lässt sich nun berechnen, wie die Einsparung auf der Leitung zwischen München und Chemnitz ist, wenn die Newsserver *news.manner.de* in Leipzig, *news.ibh-dd.de* in Dresden und *news.netconx.de* in Jena statt

[51] Natürlich unter der Voraussetzung, daß auf den Leitungen selbst keine Paketverluste auftreten
[52] Stand 22.7.1997

		Ohne Kompression	Mit Kompression
Artikelgröße	Bytes	2000	2000
Artikel pro Tag		100.000	100.000
Artikelvolumen ohne *Mcntp*-Header	MBytes	190,7	190,7
Artikelvolumen mit *Mcntp*-Header	MBytes	200,1	200,1
Volumen verschickt	MBytes	200,1	124,6
Datenrate	kBit/s	19	12

Tabelle 12: Einfluß von Datenkompression auf die Datenrate

via NNTP aus Karlsruhe , via *Mcntp* versorgt werden und nur die Artikel, die verloren gegangen sind, via NNTP nachgeliefert werden (natürlich können diese beiden Server auch direkt vom Server *Connemara.xlink.net* versorgt werden, was diese Einsparung auf der Leitung von München nach Chemnitz auch bringen würde, aber dieser neue Server steht erst seit kurzem zur Verfügung. Außerdem lässt sich das Rechenmodell auch auf andere Bereiche im Backbone anwenden, wo kein Server wie *Connemara.xlink.net* zur Verfügung steht (vgl. auch 3.8 auf Seite 31).

Wenn die Artikel nur mittels NNTP von Karlsruhe nach Dresden, Leipzig und Jena gelangen, beträgt die Datenmenge 200 MB pro Tag und Feed also 600 MB am Tag (bzw. 56,9 kBit/s) auf der Leitung von München nach Chemnitz.

Unter der Annahme, daß nur 95 % aller Artikel via *Mcntp* transportiert werden und die restlichen 5 % der Artikel ca. 15 % des Volumens ausmachen [53], ergibt sich die Datenmenge, die transportiert wird, wie folgt (Tabelle 13):

	Volumen	Bandbreite
Artikel via *Mcntp*	106 MB	10,1 kBit/s
Rest via NNTP (3*)	90 MB	8,5 kBit/s
Gesamt	196 MB	18,6 kBit/s
Einsparung	404 MB	38.3 kBit/s

Tabelle 13: Rechenbeispiel: tatsächliche Einsparung

Wie man sieht, ist beim Newstransport via NNTP eine 64kBit Leitung schon nahezu ausgelastet. Wenn die Artikel via *Mcntp* ausgeliefert werden, und nur fehlende oder (für *Mcntp*) zu große Artikel via NNTP nachgeliefert werden, ist die Leitung nicht einmal zur Hälfte belastet.

6.4.3 Einfluß von Verzögerungen bei NNTP

Als nächstes wurde die Verzögerung der Auslieferung zwischen *Blackbush.xlink.net* und *Connemara.xlink.net*, *Connemara.xlink.net* und *Micky.ibh-dd.de* = *news.ibh-dd.de*, sowie *Blackbush.xlink.net* und *Micky.ibh-dd.de* variiert, wie dies in Tabelle 14 auf Seite 73 dargestellt ist.

Bei dieser Messung wurden den jeweiligen Empfängern von den jeweiligen Sendern parallel zur Übertragung via *Mcntp* die selben Artikel via NNTP zugestellt. Diese Zustellung erfolgte mittels nntplink, wobei die Übertragung via NNTP um verschiedene Zeitspannen verzögert wurde. Bei einer Verzögerung von kleiner gleich einer Sekunde ist der Transport via NNTP zwischen *Blackbush* und *Connemara* deutlich schneller, als der Transport via *Mcntp*. Dies hängt vorallem damit zusammen, daß *Mcntp* viel Zeit auf das Komprimieren der Artikel und vorallem auf

[53] Diese Zahlen wurden unter der Berücksichtigung der Tatsache gemacht, daß die meisten Artikel, die großër als 30 oder gar 60 kBytes sind, in alt.* zu finden sind. Außerdem wurde Tabelle 5 auf Seite 16 berücksichtigt.

		Blackbush an Connemara		Connemara an Micky	
Verzöge-rung	Via NNTP gesendet	via NNTP akzeptiert	Quote	Via NNTP akzeptiert	Quote
0s	1000	960	96%	234	23,4 %
1s	1000	967	96,7%	72	7,2 %
2s	1000	336	33,6%	49	4,9 %
3s	1000	158	15,8%	35	3,5 %
4s	1000	53	5,3%	10	1,0 %
5s	1000	2	0,2%	1	0,1 %

Tabelle 14: Effekt von verzögerten NNTP-Verbindungen

die Erzeugung und Verifikation der digitalen Signatur verwendet. Ab fünf Sekunden Verzögerung sind dann alle Artikel (bis auf wenige, die auf dem Transportweg verlorengegangen sind) via *Mcntp* geliefert worden.

Die Anzahl der Artikel, die *Connemara* bei *Micky* einliefern kann, selbst wenn der NNTP-Transfer ohne Verzögerung betrieben wird, ist deutlich kleiner, als zwischen *Blackbush* und *Connemara*. Dies hat damit zu tun, daß die via *Mcntp* verteilten Artikel auf beiden Hosts dekodiert werden müssen. Erst danach kann *Connemara* den Artikel via NNTP weiterleiten. In dieser Zeit ist der Artikel bereits auf *Micky* angekommen und wird auch dort dekodiert.

Bei beiden Messungen fällt auf, daß erst bei einer Verzögerung von ca. fünf Sekunden fast alle Artikel via *Mcntp* übertragen werden. Es scheint, als ob die Router auf dem Weg die Artikel teilweise recht lange zwischenpuffern. Dies ist allerdings schwer meßbar, da die Router selbst keine Vorrichtungen dafür haben.

6.4.4 Einfluß von Crosspostings

Crosspostings sind Artikel, bei denen mehr als eine Gruppe im *Newsgroups:* Header eingetragen ist. Damit erscheint der Artikel dem Leser, als ob er in beide Gruppen gepostet wurde, wird aber nur einmal übertragen und auch nur einmal im Dateisystem abgelegt.

Da bei *Mcntp* verschiedene Newsgruppen getrennt übertragen werden können, kann es passieren, daß ein Artikel, der ein Crossposting in zwei Gruppen ist, damit auch zweifach übertragen wird (z.B. Ein Artikel ist ein Crossposting nach *de.rec.motorrad* und *rec.motorcycles.harley*. Es werden *de.** und *rec.** getrennt verteilt, wie in Abschnitt 5.2.2 gezeigt.). Damit wird die doppelte Bandbreite zum Transport benötigt[54].

Dem NNTP-Server angeboten	489154 Artikel	100%
Vom NNTP-Server akzeptiert	461213 Artikel	94,28%
Vom NNTP-Server zurückgewiesen	27908 Artikel	5,72%

Tabelle 15: Duplikate aufgrund von Crosspostings

Die Werte aus Tabelle 15 stammen vom Host *Connemara* und wurden aus den Statusmeldungen generiert, die mcrcv produziert. Es wurden also 5% der Artikel wegen Crosspostings doppelt von Karlsruhe nach Chemnitz übertragen. Der Bandbreitenbedarf der Artikel ist also im Gegensatz zu den Daten in Tabelle 13 noch 5% höher und beträgt 10,6 kBit/s anstelle von 10,1 kBit/s. Dies ist gegenüber dem Wert von 56,9 kBit/s, die ein reiner NNTP-Feed aus Karlsruhe benötigen würde, immer noch sehr gut.

[54] Da beide Artikelkopien die gleiche Message-ID haben, wird nur einer von beiden vom Empfängernewssystem akzeptiert. Somit entstehen keine Duplikate beim Empfänger.

Selbst dann, wenn die Empfänger in Leipzig, Dresden und Jena ausschließlich von Chemnitz aus versorgt würden und nur der Newssserver in Chemnitz Artikel via NNTP aus Karlsruhe erhalten würde, wäre der Bandbreitenbedarf mit 16,8 kBit/s auf der Strecke Karlsruhe Chemnitz immernoch deutlich höher. Dies kommt daher, daß NNTP im Gegensatz zu *Mcntp* die Artikel vor dem Versand nicht komprimieren kann.

Der Versand der Artikel via *Mcntp* ist also, obwohl manche Artikel mehrfach übertragen werden, wesentlich effektiver als die Übertragung via NNTP.

6.5 Ausfalltoleranz

Der Newstransport ausschließlich über *Mcntp* ist nicht ausfallsicher, da kein gesichertes Multicastprotokoll verwendet wird. Wenn aber, wie es in dieser Arbeit beschrieben wurde, ein NNTP-Feed später den Empfängern die Artikel, die sie möglicherweise erhalten möchten, nochmals anbietet, ist der Newstransport sehr ausfallsicher. Der limitierende Faktor ist dabei der NNTP-Server auf dem Sendersystem. Fällt dieser aus, können weder Artikel via *Mcntp*, noch via NNTP verschickt werden.

Die zum Transport der Newsartikel via *Mcntp* verwendeten Programme (mcxmit, mcrcv und yawgmoth) haben sich im Praxisbetrieb als stabil laufend gezeigt. Es treten keine Abstürze auf und die Programme yawgmoth und mcrcv laufen vom Start bis zum Herunterfahren des Systems ohne zutun durch. mcxmit wird mehrmals täglich vom NTA neu gestartet[55], deswegen sind hier keine so großen Laufzeiten am Stück zu erwarten.

Der Newstransport via *Mcntp* läuft im Backbone von Xlink nun seit mehr als einem Monat stabil ohne Störungen oder Ausfälle.

6.6 Transportsicherheit

Durch die Verwendung von digitalen Signaturen ist *Mcntp* gegen eine unbefugte Einspeisung von Artikeln gewappnet. Artikel werden nur dann von einem Empfänger akzeptiert, wenn sie von einem bekannten Sender signiert wurden. Der Aufwand, den zur Bildung der Signatur benutzen RSA-Schlüssel cryptograhisch zu brechen ist relativ hoch. Vorallem in Hinblick darauf, daß es weltweit viele Newsserver gibt, die offen für alle zugänglich sind.

Die digitale Signatur bietet nur die Gewißheit, daß ein Artikel von einem bekannten Sender kommt und während des Transports nicht verändert wurde. Die Signatur bietet aber keinen Schutz gegen unbefugtes Mitlesen von privaten Artikeln, wie sie z.B. bei der Verteilung von privaten Newsgruppen via *Mcntp* auftreten.

Die Verschlüsselung der Artikel ist noch nicht in mcxmit und mcrcv implementiert. Es ist aber bereits ein Feld im Protokollkopf vorgesehen (siehe Abbildung 31 in Abschnitt 4.4.1 auf Seite 45), so daß es nicht sehr viel Aufwand ist, dies zu mcxmit und mcrcv hinzuzufügen.

6.7 Benötigte Rechenzeit

Die von *Mcntp* benötigte Rechenzeit hängt in erster Linie von der Rechenleistung des verwendeten Prozessors und der Qualität der verwendeten RSA-Bibliothek ab. Eine Version, bei der die interne Arithmetik in Assembler kodiert ist, läuft schneller ab, als eine Version, bei der die Routinen in der Programmiersprache C programmiert wurden.

[55] Dies ist eine Folge der inneren Abläufe von INN.

Das Programm redemo, welches der rsaeuro Bibliothek beiliegt, benötigt beispielsweise 10% mehr Rechenzeit, um eine Datei zu verschlüsseln, wenn es mit der C und nicht mit der Assembler Version übersetzt wurde[56].

mcxmit, welches die digitale Signatur erzeugt, muß hierzu den RIPEMD Fingerprint verschlüsseln (siehe auch 2.4.1 und 4.4.2). Dies ist aufwendiger, als das Entschlüsseln, das in mcrcv zur Verifikation der Signatur nötig ist.

mcxmit benötigt auf *Blackbush* (Sun Ultra Sparc System) bei vier Artikeln pro Sekunde (dies entspricht 345.000 Artikeln am Tag) ca. 30% CPU. Die RSA Bibliothek ist hierbei eine C Version, da in der Dokumentation zur Bibliothek steht, daß die Sparc Assembler Version nicht auf allen CPUs korrekt läuft. Es wurden im Testbetrieb schon bis zu acht Artikel pro Sekunde versandt.

Auf *Connemara*, ebenfalls eine Sun Ultra Sparc, benötigen die Empfängerprozesse zusammen maximal 10% CPU – typischerweise weniger als 2%.

Die mcxmit Prozesse benötigen relativ viel CPU Leistung – vorallem auch relativ zu den mcrcv Prozessen. Da aber die Newstransferagents im Normalfall genug CPU-Leistung zur Verfügung haben[57], ist dies nicht weiter störend. Falls die CPU-Leistung nicht ausreicht, alle Artikel mit nur einer CPU zu verarbeiten, können die Artikel auch von mehreren Maschinen aus verteilt werden.

Mcntp ist auf jeden Fall leistungsfähig genug für das aktuelle Newsvolumen und hat noch Luft für zukünftige Entwicklungen.

Ein interessanter Effekt im Zusammenhang mit der Benutzung von *Mcntp* auf Senderservern liegt darin, daß die nötige Eingabe-/Ausgabeleistung zu den Festplatten hin sinkt. Dies geschieht, obwohl *Mcntp* zusätzlich zu den vorhandenen NNTP-Verbindungen betrieben wird. Beim Transport via NNTP wird derselbe Artikel nicht gleichzeitig an alle Empfänger verteilt, sondern diese Zeitpunkte variieren. Damit muß der Artikel mehrfach von der Festplatte in den Arbeitsspeicher geladen werden. Beim Versand via *Mcntp* geschieht dies nur einmal (bzw. wenn der Artikel sich noch im Plattenpuffer befindet, garnicht – siehe auch Abschnitt 2.3.4). Nachfolgende NNTP-Transfers, die benutzt werden, um verlorengegangene Artikel nachzuliefern, müssen nur noch für die Artikel, die tatsächlich übertragen werden, auf die Festplatte zugreifen.

6.8 Probleme

Eine kleine Unschönheit gibt es leider auch im Zusammenhang mit *Mcntp*: Artikel, die auf einem Host produziert werden, der die entsprechende Newsgruppe via *Mcntp* empfängt, werden wieder zu diesem Host zurücktransportiert. Beispiel: Es wird ein Artikel auf *Connemara* erzeugt. Dieser Artikel wird von *Connemara* an *Blackbush* weitergeleitet. *Blackbush* verteilt den Artikel via *Mcntp* und dieser wird wieder zu *Connemara* transportiert.

Die einzige Möglichkeit dies zu verhindern, die aber nicht gut skaliert und deshalb in der Praxis nicht einsetzbar ist, wäre, alle Artikel direkt auf dem Server zu posten, der diese später via *Mcntp* verteilt. Allerdings verhindert dies nur, daß ein Host einen Artikel, der auf ihm gepostet wurde, diesen nochmals via *Mcntp* erhält. Dieses Vorgehen verhindert nicht, daß Netzwerkbandbreite mehrfach benötigt wird, weil zum Posten auf dem Senderhost eine NNTP-Verbindung dorthin gemacht werden muß, um den Artikel einzuliefern. Der Unterschied zu einer Transportverbindung via NNTP besteht nur in den Kommandos die dafür genutzt werden.

Dieses Problem ist allerdings in der Praxis nicht so groß, wie es hier aussieht. Die in Tabelle 15 gezeigten Duplikate sind alle Duplikate - es kann nicht festgestellt werden, ob diese durch Crosspostings erzeugt werden, die

[56] auf i486dx2/80

[57] Die Newstransferagents benötigen vorallem sehr viel Bandbreite für die Festplattenzugriffe. CPU Leistung wird von den NTAs nur wenig gefordert.

mehrfach verschickt wurden oder dadurch, daß Artikel, die *Connemara* an *Blackbush* geschickt hat, wieder via *Mcntp* bei *Connemara* angekommen sind.

6.9 Zusammenfassung

Dieses Kapitel beschreibt das Verhalten von *Mcntp* im praktischen Einsatz. *Mcntp* erfüllt die in Kapitel 3 geforderten Eigenschaften und läuft stabil im Praxisbetrieb bei dem Internetserviceprovider Xlink.