Universität Stuttgart

Fakultät Informatik, Elektrotechnik und Informationstechnik

Institut für Parallele und Verteilte Systeme

Diplomarbeit Nr. 3132

Scalable Emulation of Network Links between Virtual Nodes

Markus Schirmer

Studiengang:	Informatik
Prüfer:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Betreuer:	Andreas Grau
begonnen am:	28. Dezember 2010
beendet am:	30. Juni 2011
CR-Klassifikation:	C2.4, C4, I6.8

Abstract

In these days, readiness to profit from non-local services is increasing continually. Non-local offers are gaining impressively in importance entailing satisfaction of needs of a steadily growing network-infrastructure as well as number of applications in order to make the most of this network-infrastructure. In order to be able to test such applications sufficiently, network-testenvironments are inquired which can emulate networktopology as large as possible and can handle numerous instances of the application to be tested. To this, it is necessary to work within these network-testenvironments with as many as possible different broadcastdomains. In this diplom thesis it is required to work under the emulation architecture TVEE of the project NET of the University of Stuttgart, Department Distributed Systems. In TVEE, the applications to be tested are carried out in virtual nodes. In order to get different broadcastdomains, it is necessary to have the possibility of grouping those virtual nodes. To this aim, TVEE employs VLAN-technique. On account of the definition, the VLAN-technique is limited to a relatively small number of VLAN-groups possible. Object of the present diplom thesis is to discover a method for a removal or enlargement of this restriction.

Zusammenfassung

Die Nutzung von nicht lokalen Diensten nimmt aktuell stetig zu. Die Bedeutung von online Angeboten wird immer größer. Daraus folgt auch der Bedarf nach einer stetig wachsenden Netzwerkinfrastruktur und einer ständig zunehmenden Zahl an Anwendungen, welche die Möglichenkeiten dieser Netzwerkinfrastruktur ausnutzen. Um solche Anwendungen ausreichend testen zu können, werden Netzwerktestumgebungen benötigt, die eine möglichst große Netzwerktopologie emulieren können und mit vielen Instanzen der zu testenden Anwendung zurecht kommen. Dazu ist es notwendig, innerhalb dieser Netzwerktestumgebungen mit möglichst vielen unterschiedlichen Broadcastdomains arbeiten zu können. In der Diplomarbeit soll unter der Emulationsarchitektur TVEE des Projekts NET der Universität Stuttgart Abteilung Verteilte Systeme gearbeitet werden und in TVEE werden die zu testenden Anwendungen in virtuellen Knoten ausgeführt. Es ist notwendig, diese virtuellen Knoten gruppieren zu können, um unterschiedliche Broadcastdomains zu erhalten. TVEE arbeitet zu diesem Zweck mit VLAN-Technik. Die VLAN-Technik ist aber aufgrund der Definition auf eine relativ kleine Anzahl möglicher Vlangruppen beschränkt. In dieser Diplomarbeit soll ein Weg gefunden werden, diese Beschränkung zu erweitern oder aufzuheben.

Danksagung

Ich möchte hier meinem Betreuer, Andreas Grau, Dank aussprechen. Er hat mich durch die Schwierigkeiten dieser Diplomarbeit geleitet. Seine Anregungen waren immer eine große Hilfe während meiner Arbeit. Er hat mir den Weg zu guter wissenschaftlicher Arbeit aufgezeigt.

Inhaltsverzeichnis

1 Einleitung **9**
- 1.1 Motivation . . . 9
- 1.2 Thema der Diplomarbeit . . . 11
- 1.3 Outline . . . 12

2 Verwendete Komponenten **13**
- 2.1 Virtualisierung . . . 13
  - 2.1.1 Hypervisor Virtualisierung . . . 15
  - 2.1.2 Betriebssystem-level Virtualisierung . . . 16
- 2.2 TVEE . . . 17
  - 2.2.1 Hardware des verwendeten Clusters . . . 18
  - 2.2.2 Knotenvirtualisierung . . . 19
  - 2.2.3 Zeitvirtualisierung . . . 19
- 2.3 Netzwerk . . . 21
  - 2.3.1 Sicherungsschicht . . . 21
  - 2.3.2 Ethernet . . . 22
    - MAC . . . 22
    - Multicast . . . 23
    - Tunnel . . . 23
    - Vlan . . . 23
  - 2.3.3 Vermittlungsschicht . . . 24
    - IP . . . 24
    - Multicast . . . 24
    - Tunnel . . . 25

3 Related Work **26**
- 3.1 Tunnel . . . 26
  - 3.1.1 MobiNet . . . 26
  - 3.1.2 Empower . . . 27
- 3.2 MAC Adressen Modifikation . . . 28
  - 3.2.1 V-eM . . . 28

3.3 Vlan . . . 30
3.3.1 TVEE . . . 30
3.4 Verschachteltes Vlan . . . 32

4 Entwurfskriterien **34**
4.1 Mögliche Vorgehensweisen . . . 34
4.2 Hardwareimplementierung . . . 35
4.3 MAC Adressen Modifikation . . . 36
4.4 MAC Multicast . . . 38
4.5 Tunnel . . . 40
4.6 Vlan . . . 40
4.7 Einschränkungen durch Hardware . . . 44

5 Konzept für Umsetzung der Vlanerweiterung **46**
5.1 Komponenten . . . 46
5.2 Design . . . 47
5.2.1 Zwei Ebenen Baumstruktur . . . 47
5.2.2 Zwei Ebenen Flat . . . 49
5.2.3 Sendevorgang . . . 51
5.3 Zusammenfassung . . . 52

6 Erzeugen der Vlangruppen **53**
6.1 Mögliche Ansätze . . . 53
6.2 Mehrfachsenden . . . 54
6.3 Direct . . . 55
6.4 Broadcast . . . 55
6.5 Gruppen ersetzen . . . 55
6.5.1 All . . . 58
6.5.2 Groups . . . 59
6.5.3 Large Groups . . . 59
6.5.4 Zusammenfassen von Vlangruppen . . . 60
6.6 Kombinationen . . . 60
6.6.1 Direct - Broadcast . . . 60
6.6.2 Large Groups - Direct . . . 60
6.6.3 Large Groups - Direct - Broadcast . . . 61
6.6.4 Middle Groups - Direct - Broadcast . . . 61
6.7 Komplexität . . . 61
6.7.1 Direct . . . 62
6.7.2 Broadcast . . . 62

7 Implementierung der Vlangruppenerstellung 63
7.1 Szenario . . . 63
7.2 Direct . . . 64
7.3 Broadcast . . . 64
7.4 Gruppen ersetzen . . . 65
7.4.1 Gemeinsame Teile aller Verfahren . . . 66
Gruppenkombinationen . . . 66
Gruppenvergleiche . . . 67
Prüffunktion . . . 68
Kostenberechnung . . . 68
Pfadprüfung . . . 69
Verbesserungen . . . 70
7.4.2 All . . . 70
7.4.3 Groups . . . 72
7.4.4 Large Groups . . . 72
7.4.5 Zusammenfassen von Vlangruppen . . . 73
7.5 Kombinationen . . . 73
7.5.1 Direct - Broadcast . . . 73
7.5.2 Large Groups - Direct . . . 74
7.5.3 Large Groups - Direct - Broadcast . . . 74
7.5.4 Middle Groups - Direct - Broadcast . . . 74

8 Evaluierung 76
8.1 Szenarien . . . 76
8.1.1 Szenariogenerator . . . 77
8.2 Ziel . . . 78
8.2.1 Overhead . . . 78
8.2.2 Benchmark . . . 78
8.2.3 Laufzeit . . . 78
8.3 Ergebnisse . . . 78
8.3.1 Direct . . . 79
8.3.2 Broadcast . . . 79
8.3.3 Gruppen ersetzen . . . 81
8.3.4 Kombinationen . . . 81
8.3.5 Direct - Broadcast . . . 81
8.3.6 Large Groups - Direct . . . 86
8.3.7 Large Groups - Direct - Broadcast . . . 86
8.3.8 Middle Groups - Direct - Broadcast . . . 89
8.4 Bewertung . . . 89

9 Zusammenfassung und Ausblick **91**
9.1 Zusammenfassung 91
9.2 Ausblick 92

Listings

7.1 Kostenberechnung Direct 64
7.2 Kostenberechnung Broadcast 65
7.3 Durchlaufen aller Gruppenkombinationen 66
7.4 Gruppenvergleich 67
7.5 Kostenberechnung 69
7.6 Erzeugung aller möglichen Gruppen aus einer Anzahl Hardwarerechner . 70
7.7 Zusammenfassen von Gruppen 73

Kapitel 1

Einleitung

1.1 Motivation

Die Nutzung von nicht lokalen Daten wird in der heutigen Zeit immer wichtiger.
Ob im privaten Bereich beim Abruf von elektronischen Nachrichten (z.Bsp. Email, Instant Messaging, Google Kalender), bei der Nutzung von online Navigationssystemen (z.Bsp. Google Maps), beim Konsumieren von Musikstücken oder Videos (unter anderem Youtube), bei der Teilnahme an online Spielen (z.Bsp. Browserspiele, Massive Multiplayer Online Games) oder auch im geschäftlichen Bereich wie der innerfirmlichen Kommunikation und der Nutzung von zentral gespeicherten Daten (unter anderem Fileserver). Ebenso steigt die Anzahl und die Verbreitung der netzwerkfähigen Endgeräte, mit denen auf diese Daten zugegriffen wird, stark an.
All diese Geräte benötigen eine Netzwerkverbindung, ob mobiles Netzwerk, fest verlegtes LAN oder auch eine Internetverbindung. Für private Anwender stellt dies meist nur ein Problem in Hinsicht auf die Bandbreite der Internetverbindung dar. Sollte der Benutzer aber an abgetrennten Netzen z.Bsp. über VPN-Verbindungen teilnehmen entsteht die Notwendigkeit, die Netzwerkverbindungen zu Gruppen zusammenzufassen, sie voneinander abzuschotten.
Internet Provider, Anbieter von Internetinhalten, Universitäten oder große Unternehmen müssen mit einer großen Menge an Netzverbindungen zurecht kommen. Hier gibt es mehrere Fälle, in denen nicht nur die Bandbreite ein Problem darstellt, sondern auch eine Gruppierung der Netzwerkverbindungen aus administrativer oder sicherheitsthematischer Sicht sinnvoll ist. Eine Vielzahl unterschiedlicher Möglichkeiten bietet sich für die Gruppierung an.
Ein großes Unternehmen möchte seine PC-Arbeitsplätze vielleicht gerne in Gruppen, passend zu den vorhandenen Abteilungen verwalten. Die Sicherheit im Unternehmen wird durch diese Isolation von Arbeitsgruppen vereinfacht, da nur für die jeweilige Arbeitsgruppe relevante Daten und Endgeräte sichtbar sind. Um dies zu erreichen, kann das

Unternehmen auf OSI-Layer 3 intern IP-Subnetze für jede Abteilung vergeben. Probleme treten hier dann auf, wenn ein Arbeitsplatz oder mehrere in ein anderes Gebäude oder eine andere Zweigstelle umziehen müssen. Dann muss die Netzwerkinfrastruktur mit erhöhtem Aufwand angepasst werden.
Alternativ kann das Unternehmen auch auf OSI-Layer 3 oder 4 Tunnelverbindungen einrichten, um zusammengehörende Bereiche des Unternehmens zu verbinden.
Eine weitere Möglichkeit ist die Verwendung von Virtual Local Area Networks (Vlan). Vlans können Netzwerke auf OSI-Layer 2 in einzelne Netzsegmente aufteilen. Dann können PC-Arbeitsplätze mit relativ geringem Arbeitsaufwand innerhalb eines Unternehmens umziehen oder einzelne Arbeitsplätze schnell einer anderen Arbeitsgruppe zugeteilt werden. Vlans werden entweder den einzelnen Ports verwendeter Switches zugeteilt oder aber jedes versendete Datenpaket wird schon vom Absender mit einer Vlan-ID versehen, um die Zugehörigkeit zu einem bestimmten Vlan zu signalisieren. Ein großer Vorteil des Einsatzes von Vlans ist, dass aktuelle Netzwerkinfrastruktur wie Hardwareswitches die Vlantechnologie beherrschen und so die gesamte Verteilungsarbeit übernehmen können.
Sind mehrere große Unternehmen über den selben Internetprovider angeschlossen und alle diese Unternehmen haben ihre Arbeitsgruppen über mehrere Zweigstellen verteilt, das bedeutet, sie müssen die Gruppenzugehörigkeit durch das Netzwerk des Providers übertragen, kann dies bei der Verwendung von Vlan-IDs zu Problemen führen. Wenn alle Unternehmen mit der Vlan-ID 1 beginnen, dann werden mehrere Datenpakete mit der selben Vlan-ID im Providernetzwerk auftauchen. Bei Unicastpaketen stellt dies kein Problem dar, da der Empfänger bekannt ist, bei Broadcastpaketen jedoch ist die Vlan-ID relevant zur Findung des Empfängers. Hier bleibt nur die Tunnelung der Vlan-Pakete durch das Netz des Internetproviders.
Ebenfalls abhängig von der Möglichkeit Netzwerkverbindungen gruppieren zu können, ist der Bereich der Netzwerktestumgebungen. Hier werden Netzwerkprotokolle und Programme, die Netzwerke verwenden, auf Leistung und Schwachstellen getestet.
Die erhöhte Anzahl netzfähiger Endgeräte und die ansteigende Verwendung von nicht lokalen Daten über Netzwerke erfordert bei der Verwendung von Netzwerktestumgebungen auch die Berücksichtigung immer größerer Szenarien.
Netzwerktestumgebungen sind mittlerweile meist in der Lage mittels Virtualisierung eine große Anzahl an virtuellen Testknoten zu erzeugen die, je nach emulierter Netzwerkumgebung, in Gruppen aufgeteilt werden. Diese Gruppen lassen sich auf unterschiedliche Art und Weise realisieren. Wie bei großen Unternehmen können auch hier auf OSI-Layer 3 oder 4 Tunnel zwischen verbundenen Gruppen eingerichtet werden. Auch das Umschreiben von MAC-Adressen, wie es das Projekt V-em [AH06a] einsetzt, ist eine mögliche Vorgehensweise. Hier werden bei gesendeten Broadcasts die Ziel-MAC-Adressen in der letzten Softwarebridge der Virtualisierungssoftware mit den einzelnen Ziel-MAC-Adressen der gewollten Empfängerknoten überschrieben. Ebenso ist die Verwendung von Vlans eine mögliche Vorgehensweise. Die gesendeten Datenpakete der virtuellen Knoten können,

je nach gewünschter Gruppenzugehörigkeit, mit Vlan-IDs versehen werden. So ist das Gruppieren und Umgruppieren von virtuellen Knoten mit relativ geringem Aufwand möglich.
Ein Problem, das sowohl bei großen Unternehmen, Internet Providern sowie in Netzwerktestumgebungen bei der Verwendung von Vlans zu berücksichtigen ist, ist die Begrenztheit der Vlan-IDs auf zwölf Bit im Ethernetframe. Das entspricht 4096 unterschiedlichen Vlan-IDs. In allen drei Bereichen, besonders aber bei Netzwerktestumgebungen, ist dies unter Umständen nicht ausreichend.
Um Vlans in diesen Bereichen einsetzen zu können, bedarf es einer Erweiterung, die mehr als 4096 Gruppen ermöglicht.

1.2 Thema der Diplomarbeit

In dieser Diplomarbeit wird die Problematik der Gruppierung von Netzwerkverbindungen im Rahmen des Forschungsprojektes NET [NET11] der Universität Stuttgart näher betrachtet.
Bei NET wird ein Cluster eingesetzt, der es mittels Virtualisierungsverfahren ermöglicht eine große Anzahl an virtuellen Knoten zu erzeugen. Diese virtuellen Knoten können eingesetzt werden um beliebige Anwendungen auszuführen, da sie sich wie physikalische Knoten verhalten.
Um nun Netzwerkkommunikation zwischen den Knoten zu evaluieren, muss eine sinnvolle Netzwerkinfrastruktur erstellt werden. Das bedeutet, Gruppen aus virtuellen Knoten müssen erstellbar sein. Bei NET kommen hier bisher Vlans zum Einsatz, die aber aufgrund der Limitierung der Vlan-IDs nicht der vollen Leistungsfähigkeit des Clusters gerecht werden.

Um diese Einschränkung aufzuheben, wird in dieser Diplomarbeit ein Verfahren entwickelt, das normale Vlanverbindungen weiterhin unterstützt, somit weiterhin von der beschleunigten Hardwareverarbeitung der Vlans in den beteiligten Switches profitieren kann. Darüber hinaus aber die Möglichkeit schafft, mehr als 4096 Gruppen virtueller Netzwerkverbindungen zu erzeugen.
Das ermöglicht Netzwerkemulationsumgebungen mit realistischeren, da größeren, Netzwerkszenarien zu betreiben.

1.3 Outline

Die Kapitel dieser Diplomarbeit sind wie folgt aufgebaut:

Kapitel 2 führt die notwendigen Grundlagen und Komponenten ein, die für das Verständnis der Diplomarbeit förderlich sind. Es werden unter anderem Netzwerkprotokolle und die Emulationsarchitektur TVEE vorgestellt.

In Kapitel 3 werden Arbeiten vorgestellt, die mit der Thematik in Verbindung stehen. Die Unterschiedlichen Ansätze große Anzahlen von virtuellen Netzwerklinks zu verarbeiten liefert Interessante Anregungen.

Unterschiedliche Herangehensweisen an die Problematik Netzwerkverbindungen zu gruppieren werden in Kapitel 4 diskutiert. Unter anderem wird auf die Vlantechnik eingegangen.

In Kapitel 5 werden die in Kapitel 4 als am interessantesten bewerteten Ansätze detaillierter betrachtet und etwaige Probleme untersucht.

Die Problematik der Beschränkung des Hardwareswitches auf 4096 Vlan-IDs und Möglichkeiten diese so effizient es geht abzumildern werden in Kapitel 6 besprochen.

Kapitel 7 enthält die Implementierung von Verfahren um größere Anzahlen von Gruppen möglichst effizient auf kleinere Anzahlen von Vlan-IDs abzubilden.

In Kapitel 8 werden die in Kapitel 7 implementierten Verfahren mittels eines rechnerischen Kostenmodells evaluiert.

Kapitel 9 enthält eine Zusammenfassung und einen Ausblick auf weiterführende Arbeiten.

Kapitel 2

Verwendete Komponenten

In diesem Kapitel werden Komponenten und Verfahren betrachtet, die für die Durchführung der Diplomarbeit notwendige Voraussetzung sind. Da TVEE auf unterschiedliche Netzwerkschichten zugreift und für die Lösungsansätze dieser Diplomarbeit ganz spezielle Bereiche des Protokollstack notwendig sind wie z.Bsp. Multicast oder Vlan werden auch Grundlagen der Netzwerktechnik erwähnt. Hier wird in Anlehnung an das Lehrbuch von Andrew S. Tanenbaum [Tan03] ein vereinfachtes OSI-Modell betrachtet, welches auf die beiden Schichten 5 (Sitzungsschicht) und 6 (Darstellungsschicht) verzichtet.
Des weiteren wird die Hardware des unter NET [NET11] zum Einsatz kommenden Clusters betrachtet und die Emulationsarchitektur TVEE(Time Virtualized Emulation Environment), welche auf dem Cluster eingesetzt wird.
Zudem werden die unterschiedlichen Virtualisierungstechniken die unter TVEE verwendet werden und die hierfür notwendige Software näher betrachtet.
Diese Virtualisierungstechniken ermöglichen erst den Einsatz einer großen Zahl an virtuellen Knoten.

2.1 Virtualisierung

Eine virtuelle Maschine (VM) ist eine Software, die eine virtuelle Umgebung auf einem Rechner erzeugt, auf dem sie ausgeführt wird. Je nach virtueller Maschine kann innerhalb dieser Umgebung nur ein einzelner Prozess ausgeführt oder aber ein kompletter virtueller Rechner beherbergt werden. Die virtuelle Maschine macht die in ihr ausgeführten Prozesse unabhängig vom Betriebssystem des Hostrechners, der auf dem Hostrechner laufenden Anwendungen und, je nach virtueller Maschine, auch von der Hardware des Hostrechners.

Virtuelle Maschinen setzen auf unterschiedlichen Ebenen zwischen Hardware und Betriebssystem auf. Direkt in Hardware ist z.Bsp. IBM LPAR (Logic Partition, [SCDB+09]) realisiert. LPAR partitioniert die Hardware eines IBM Großrechners in virtuelle Systeme,

in denen unterschiedliche Betriebssysteme basierend auf der Hardwarearchitektur des IBM Großrechners ausgeführt werden können.
Virtuelle Maschinen, die zwischen Hardware und Betriebssystem angesiedelt sind, verwenden einen Virtual Machine Monitor (VMM), auch Hypervisor genannt, als Schnittstelle zwischen der Hardware und dem Betriebssystem z.Bsp. XEN[BDF+03]. Diese virtuellen Maschinen virtualisieren, sofern die darunterliegende Hardware dies unterstützt, die Hardware des Hostrechners für die oberhalb des Hypervisors ausgeführten Betriebssysteme und Anwendungen. Sollte die Hardware dies nicht unterstützen, wird alternativ eine Schnittstelle mittels Paravirtualisierung geschaffen.
Auf der Ebene des Betriebssystems gibt es mehrere unterschiedliche Ansätze. Zum einen gibt es Virtuelle Maschinen, die mittels eines VMM auf das Betriebssystem aufsetzen. Diese lassen sich noch in zwei weitere Gruppen unterteilen. Einerseits virtuelle Maschinen, die eine komplette Hardware emulieren und andererseits virtuelle Maschinen, welche die existierende Hardware des Hostrechners virtualisieren.
Mit den virtuellen Maschinen, die eine komplette Hardware emulieren können, lassen sich vom Hostrechner vollständig unabhängige Ausführungsumgebungen schaffen. Dies ermöglicht das Ausführen von Anwendungen unter einem anderen Betriebssystem auf einer vollständig unterschiedlichen Hardwarearchitektur als der Hostrechner, auf dem die virtuelle Maschine ausgeführt wird, bietet. Dies leisten z.Bsp. Qemu[Bel05] und Bochs[Boc]
Die virtuellen Maschinen, die die Hardware des Hostrechners virtualisieren, ermöglichen das Ausführen beliebiger Betriebssysteme aber basierend auf der vom Hostrechner zur Verfügung gestellten Hardware z.Bsp. VMWare[vmw] oder VirtualBox[vira].
Die andere Gruppe auf Betriebssystemebene sind die virtuellen Maschinen, die Betriebssystem-level Virtualisierung durchführen. Diese erstellen mehrere voneinander unabhängige Container innerhalb des Betriebssystems des Hostrechners. Da alle Container somit den selben Betriebssystemkernel verwenden, ist es nicht möglich, innerhalb der Container unterschiedliche Betriebssysteme zu installieren. Beispiele hierfür sind OpenVZ[ope], Virtuozzo[virb].

Des weiteren gibt es noch virtuelle Maschinen, die nur für einen einzelnen Prozess eine virtuelle Umgebung erzeugen. Die JavaVM [jav] ist so eine virtuelle Maschine. Sie ist Teil der Programmiersprache Java und ist in der Lage, den Bytecode von Javaprogrammen auszuführen. Die JavaVM dient also als Schnittstelle zwischen Javaprogrammen und Betriebssystem/Hardware. Das ermöglicht Javaprogramme unabhängig vom verwendeten Betriebssystem auszuführen.

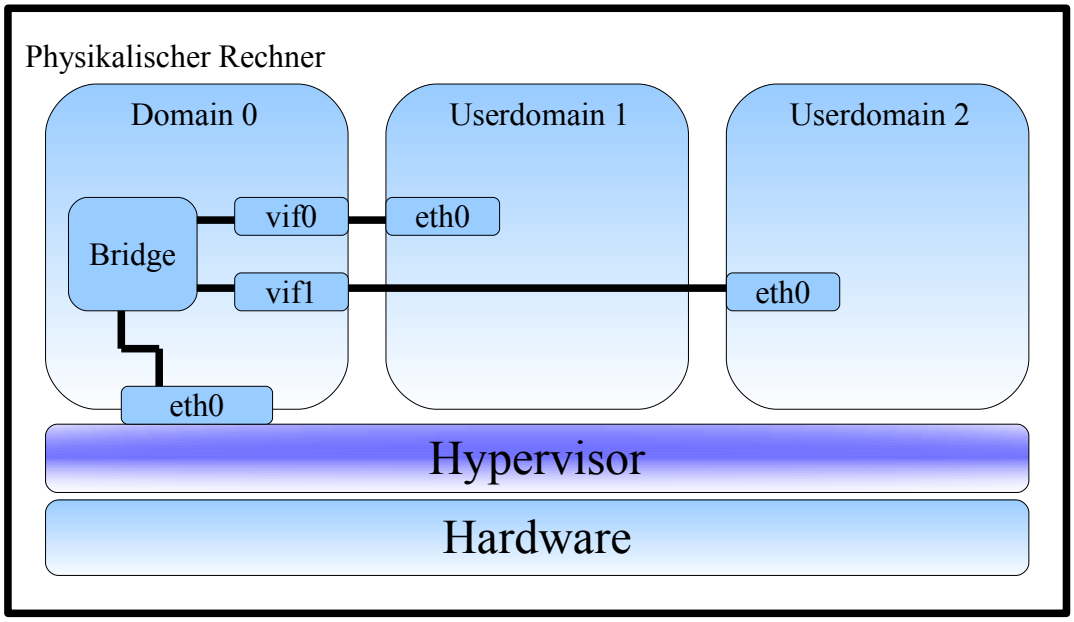
Auf dem von NET [NET11] verwendeten Cluster kommen zwei virtuelle Maschinen zum Einsatz, XEN [BDF+03] und OpenVZ [ope].

2.1.1 Hypervisor Virtualisierung

Auf dem von NET [NET11] verwendeten Cluster wird die Emulationsarchitektur TVEE, Time Virtualized Emulation Environment eingesetzt. Diese setzt 2 Virtualisierungsebenen ein, die Hypervisorvirtualisierung und die Betriebssystem-level Virtualisierung.
Für die Hypervisor-level Virtualisierung kommt XEN [BDF+03] zum Einsatz. XEN ist in der Lage, mit seinem Hypervisor direkt auf der Hardware des Hostrechners aufzusetzen, ohne ein vorinstalliertes Betriebssystem zu benötigen. Der Hypervisor stellt eine Schnittstelle zwischen der Hardware und den auf dem Hypervisor aufsetzenden Betriebssystemen dar.
XEN kennt zwei unterschiedliche Möglichkeiten, die Hardware den Gastbetriebssystemen zur Verfügung zu stellen. Die erste Möglichkeit besteht darin, falls die zugrunde liegende Hardware Hardwarevirtualisierung unterstützt, die vorhandene Hardware zu virtualisieren und dem Gastbetriebssystem auf diese Weise zur Verfügung zu stellen. Hierbei wird die komplette Hardware des Hostrechners virtualisiert und dem Gastbetriebssystem zur Verfügung gestellt. Da die komplette Hardware berücksichtigt werden muss, entsteht bei diesem Verfahren ein großer Virtualisierungsoverhead, aber das eingesetzte Gastbetriebssystem muss dadurch nicht modifiziert werden, was wiederum einen Vorteil darstellt.

Die zweite Möglichkeit ist die von XEN angebotene Paravirtualisierung. Bei der Paravirtualisierung wird die Hardware des Hostrechners nicht exakt nachgebildet sondern, XEN bildet mit dem Hypervisor eine Schnittstelle zwischen Hardware und Gastbetriebssystem nach, die auf möglichst geringen Virtualisierungsoverhead ausgelegt ist und auf hohe Leistung abzielt. Um die Paravirtualisierung einsetzen zu können, müssen deshalb die Gastbetriebssyteme in der Hardwareabstraktionsebene und bei den Gerätetreibern angepasst werden.
Aufgrund des geringeren Overheads und der Leistungsvorteile wird XEN unter TVEE im Paravirtualisierungsmodus betrieben.
Aufsetzend auf den Hypervisor erstellt XEN mehrere virtuelle Container, genannt Domains (siehe Abbildung 2.1). Diese Domains unterteilen die zur Verfügung stehenden Ressourcen gleichmäßig. Der erste Container, Domain0, dient zu administrativen Zwecken. Unter anderem wird sämtlicher Netzwerkverkehr, den die anderen Domains erzeugen, über eine virtuelle Bridge in der Domain0 geleitet. In den Containern Domain-User0 und folgenden können beliebige, auch unterschiedliche, Betriebssysteme installiert werden. Je nach eingesetzter Virtualisierungsmethode modifiziert oder nicht.

Abbildung 2.1: XEN

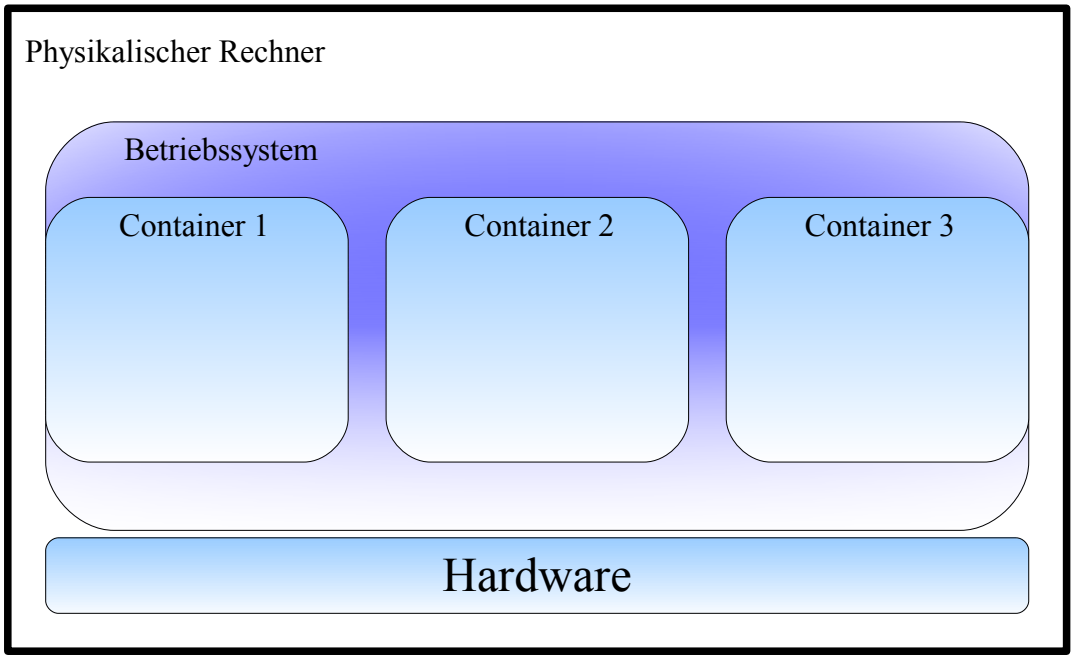


2.1.2 Betriebssystem-level Virtualisierung

Die zweite Virtualisierungsebene in TVEE besteht aus der Betriebssystem-level Virtualisierungssoftware OpenVZ [ope].
OpenVZ ermöglicht es innerhalb eines Betriebssystems mehrere voneinander isolierte Container zu erstellen. Diese Container teilen die Systemresourcen des Betriebssystems unter sich auf. Die Container werden bei OpenVZ als Virtual Environment (VE) oder Virtual Private Server (VPS) bezeichnet(siehe Abbildung 2.2).
OpenVZ basiert auf einem modifizierten Linuxkernel der die Virtualisierung, Isolierung und das Management der Systemresourcen und Subsysteme ermöglicht.
In den Containern sind unterschiedlichste Systemresourcen verfügbar. Jeder Container verfügt unter anderem über sein eigenes Dateisystem, seine eigenen Prozesse, eigene Benutzer, Netzwerk mit einer eigenen virtuellen Netzwerkkarte Somit verhalten sich die Container wie eigenständige physikalische Rechner.

Eine Einschränkung ist, dass keine unabhängigen Betriebssysteme installiert werden können, da ja alle Container im selben Hostbetriebssystem laufen und somit den selben Kernel verwenden. Das bringt auch den Vorteil eines sehr geringen Virtualisierungsoverheads mit sich. Zudem kann die Netzwerkkommunikation zwischen den einzelnen Containern über das Hostbetriebssystem erfolgen und benötigt somit auch weniger Kontextwechsel als bei

Abbildung 2.2: OpenVZ

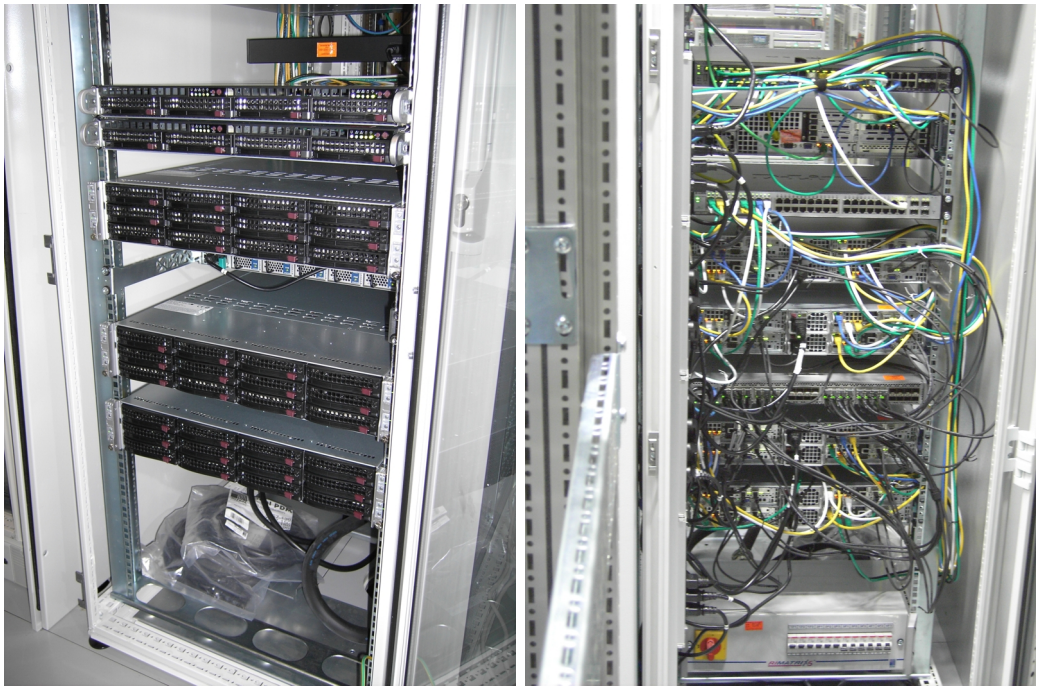


anderen Virtualisierungsmethoden. Bei diesen muss zwischen verschiedenen Betriebssystemen hin und her gesprungen werden, was einen größeren Rechenaufwand mit sich bringt.

2.2 TVEE

Im Rahmen des Forschungsprojektes NET[NET11] der Abteilung Verteilte Systeme des Instituts für Parallele und Verteilte Systeme (IPVS) der Universität Stuttgart wird ein Cluster eingesetzt, auf dem eine Emulationsarchitektur, die Time Virtualized Emulation Environment (TVEE) verwendet wird. Diese Emulationsarchitektur ermöglicht es, eine sehr große Zahl an Instanzen einer Software in einer sehr großen emulierten Netzwerkumgebung zu evaluieren. Um den Cluster, auf dem TVEE zum Einsatz kommt, optimal auszulasten, verwendet TVEE sowohl Knoten- [GMHR08] als auch Zeitvirtualisierung [GHR09]. Da beide Virtualisierungen für die ausgeführte Anwendung vollständig transparent sind, bedarf es keiner Modifikationen an der Anwendung.

Abbildung 2.3: NET Cluster



2.2.1 Hardware des verwendeten Clusters

Der Cluster, der im Projekt NET [NET11] eingesetzt wird, besteht aus 16 PCs mit jeweils acht CPUs. Es handelt sich hierbei um jeweils zwei XEON Quadcore CPUs mit 2,26GHz Taktfrequenz und 24 GByte Ram pro Rechner(siehe Abbildung 2.3).
An die Netzwerktopologie sind die einzelnen Rechner jeweils über drei Netzwerkkarten angebunden, wobei eine für den administrativen Zugriff auf die Rechner mit 1Gbit und zwei für die eigentlichen Testläufe mit 10GBit dimensioniert sind. Dadurch haben administrative Zugriffe über die 1Gbit Karte auf die einzelnen Rechner keinen störenden Einfluss auf aktuell laufende Testmessungen, welche über das 10GBit-Netzwerk laufen.
Die Netzwerkkommunikation erfolgt über einen hoch performanten Switch mit Vlan-Unterstützung, der es ermöglicht, beliebige Rechner zu einer Topologie zusammenzuschalten.
Als Betriebssystem kommt auf dem Cluster Rocks Cluster Distribution mit CentOs Linux und Kernel Version 2.6.18 mit XENErweiterung für XEN[BDF+03] Version 3.1.2 zum Einsatz.

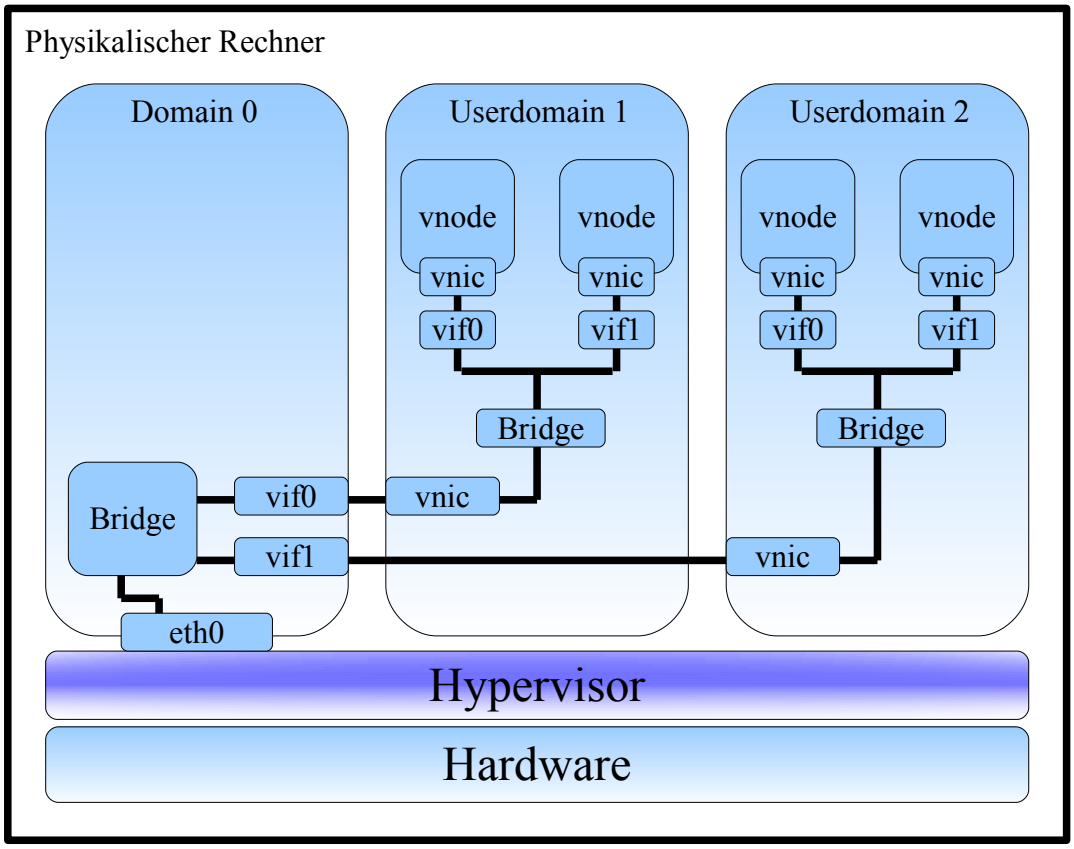
2.2.2 Knotenvirtualisierung

Die Knotenvirtualisierung [GMHR08] von TVEE besteht aus zwei Ebenen. Dies ist so realisiert, dass auf jedem Hardwareknoten (pNode) eine Instanz von XEN [BDF+03] ausgeführt wird. XEN setzt direkt auf der Hardware des pNode auf ohne ein zuvor installiertes Betriebssystem zu benötigen(siehe Abbildung 2.4). Um Virtualisierungsoverhead einzusparen und um die Leistung des Gesamtsystems zu erhöhen, wird XEN im Paravirtualisierungsmodus ausgeführt. Hier wird nicht die Hardware des Hostrechners für die Gastbestriebssysteme in den XEN Containern mittels Virtualisierung zur Verfügung gestellt, sondern der Hypervisor von XEN erzeugt eine auf geringen Overhead und auf Leistung optimierte Hardwareemulation die eine Anpassung im Gastbetriebssystem erfordert. Bei TVEE kommt hier Linux zum Einsatz mit einem speziell modifizierten Kernel. Die Anpassungen im Hardwareabstraktionslayer und in den Gerätetreibern des eingesetzten Linux haben aber keinen Einfluss auf die zu testende Software. Diese kann unmodifiziert verwendet werden.
Die zweite Ebene der Virtualisierung ist so umgesetzt, dass innerhalb des in XEN laufenden Gastbetriebssystems eine Betriebssystem-level Virtualisierung durchgeführt wird. Dies geschieht durch OpenVZ Container, die das XEN Gastbetriebssystem mittels virtuellem Routing in einzelne von einander isolierte Partitionen unterteilen. In die virtuellen Knoten (vNodes). Diese isolierten Container besitzen alle eine vollständige Arbeitsumgebung und verhalten sich wie „reale“ Hardwareknoten. Die Arbeitsumgebung enthält unter anderem eigene Prozesse, eigene Benutzer, einen eigenen Speicherbereich, eine oder mehrere eigene virtuelle Netzwerkkarten, einen eigenen Netzwerkstack bestehend aus Netzwerk- Transport- und Anwendungslayer. Auch kann jeder vNode separat gestartet und auch neugestartet werden. Da alle vNodes den selben Kernel des XEN Gastbetriebssystems verwenden, kann die Komunikation zwischen den vNodes über eine virtuelle Netzwerkkarte und eine Softwarebridge des XEN Gastbetriebssystems erfolgen. Dieses virtuelle Routing spart kostbare Systemresourcen und erzeugt weniger Overhead gegenüber der Kommunikation in anderen virtuellen Umgebungen, die jeweils vollständige Betriebssysteme pro vNode verwenden müssen [GMHR08, MGWR07].
Des weiteren unterstützt TVEE auch eine dynamische Neuanordnung der aktuell im Cluster aktiven vNodes je nach Lastsituation auf den einzelnen Hardwareknoten und kann so eine bessere Auslastung des Clusters erreichen [GHR11].

2.2.3 Zeitvirtualisierung

Unter TVEE wird Zeitvirtualisierung [GHR09] verwendet. Diese ermöglicht es, die virtuelle Zeit innerhalb der virtuellen Maschine XEN zu beschleunigen oder zu verlangsamen. Dadurch kann bei zu starker Belastung des Clusters durch Verlangsamung der virtuellen Zeit die Last reduziert werden, genauso wie bei nicht vollständiger Auslastung des Clusters durch Beschleunigen der virtuellen Zeit die Auslastung des Clusters verbessert werden kann.

Abbildung 2.4: TVEE schematische Sicht



Dies wurde erreicht, indem der Hypervisor von XEN so modifiziert wurde, dass er die virtuelle Zeit in einer Userdomain von XEN verändern kann. Dies geschieht mittels eines Faktors genannt TDF (Time Delation Factor). Mit Hilfe des für TVEE entwickelten Protocol for Latency Aware Changing of Epochs (PLACE) ist es möglich, den TDF dynamisch, je nach Auslastung des Clusters, simultan auf allen eingesetzten pNodes anzupassen und dadurch zu jedem Zeitpunkt eines Experiments eine bessere Auslastung des Clusters zu erreichen.

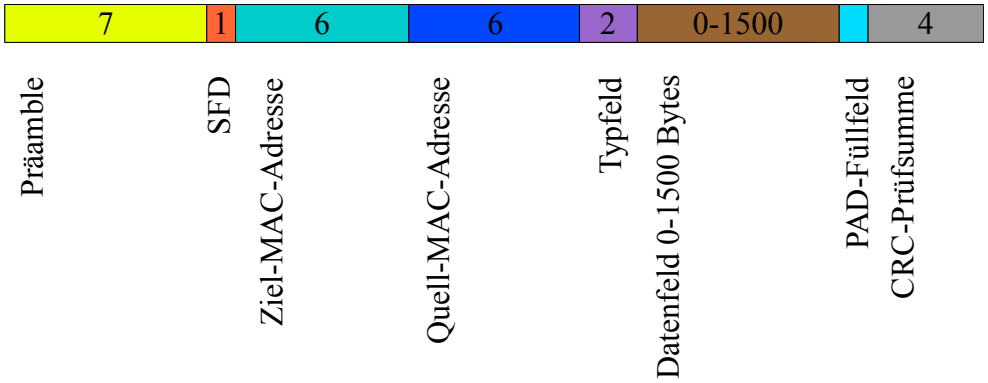
2.3 Netzwerk

Da TVEE auf unterschiedliche Netzwerkschichten zugreift und für die Lösungsansätze dieser Diplomarbeit ganz spezielle Bereiche des Protokollstack notwendig sind wie z.Bsp. Multicast oder Vlan werden auch Grundlagen der Netzwerktechnik erwähnt. Der von NET [NET11] eingesetzte Cluster verwendet ein LAN (Local Area Network), um die einzelnen Rechner miteinander zu verbinden. Es gibt unterschiedlichste Möglichkeiten, ein solches LAN aufzubauen und Daten über ein solches LAN zu übertragen. Die Organisation IEEE (Institute of Electrical and Electronics Engineers) hat hierfür einige Standards definiert [IEEb]. Der Cluster verwendet Ethernet IEEE 802.3 [IEE08], das unterschiedliche Verfahren Netzwerkverbindungen zu gruppieren ermöglicht. Des weiteren ist es auch auf den höheren Netzwerkschichten möglich, Netzwerkverbindungen zu gruppieren.

2.3.1 Sicherungsschicht

Die Sicherungsschicht versucht, eine effiziente und zuverlässige Kommunikation zwischen physikalisch verbundenen Rechnern zu gewährleisten. Hierfür werden die von der Vermittlungsschicht erhaltenen Pakete in Rahmen gekapselt bevor sie übertragen werden. Auf der Sicherungsschicht kommen verschiedene Protokolle zum Einsatz, die diese Rahmen verwalten. Eine ausführliche Übersicht kann hier [Tan03] gefunden werden.
Um Netzwerkverbindungen zu gruppieren, ist es innerhalb der Sicherungsschicht unter anderem möglich, mit Hilfe von MAC Multicasts (Media Access Control) Nachrichten zu verschicken, die nur bestimmte Rechner erhalten sollen. Des weiteren ist es möglich Tunnel einzusetzen, um die Frames gezielt an bestimmte Rechner zu verschicken z.Bsp. L2TP(Layer Two Tunneling Protocol) [RFC99].

Abbildung 2.5: Ethernetheader



2.3.2 Ethernet

Bei Ethernet IEEE 802.3 [IEE08] sind alle Rechner über ein Medium miteinander verbunden. Sollte ein Bus oder ein über einen HUB verbundenes Netzwerk zugrunde liegen, werden alle Nachrichten, die ein Rechner sendet von allen anderen Rechnern empfangen. Wird der HUB durch einen Switch ersetzt, wird eine gesendete Nachricht nur noch an den gewollten Empfänger übertragen. Um mit einem Medium für Mehrfachzugriff arbeiten zu können, sind einige Protokolle notwendig. Diese werden unter dem nächsten Punkt genauer betrachtet.

MAC

Ein Teil der Sicherungsschicht ist die MAC Teilschicht (Media Access Control). Zuerst einmal wird jeder Netzwerkkarte ein 48 Bit Schlüssel zugeteilt, die sogenannte MAC Adresse(siehe Abbildung 2.5). Mit ihrer Hilfe sind alle Rechner, die an einem LAN beteiligt sind, eindeutig identifizierbar. Nun können, falls kein Switch im Einsatz ist, alle anderen Rechner nicht gewollte Nachrichten anhand der MAC Adresse identifizieren und bereits in der Netzwerkkarte verwerfen. Falls es gewünscht wird eine Nachricht an alle am selben Netz angeschlossenen Rechner zu senden, dann kann dies mittels einer Broadcastnachricht geschehen. Hierfür ist eine extra MAC Adresse reserviert, sie besteht nur aus 1er Bits. In der MAC Teilschicht sind auch mehrere Protokolle angesiedelt, die den Zugriff mehrerer Rechner auf ein gemeinsames Medium regeln, wie z.Bsp. ALOHA oder CSMA (Carrier Sense Multiple Access) in verschiedenen Ausführungen. Diese sind hier ausführlich beschrieben [Tan03].

Multicast

Falls nun in einem LAN nicht ein Rechner adressiert werden soll und auch nicht alle an das LAN angeschlossenen Rechner sondern eine ausgewählte Gruppe, dann gibt es die Option eine Multicastnachricht zu verschicken. Hierfür sind MAC Adressen mit dem Prefix „01 00 5e“ reserviert. Rechner, die solche Multicastnachrichten empfangen möchten, müssen sich bei dem Router des LAN (falls dieser Multicastnachrichten unterstützt) für eine Multicastgruppe anmelden. Dafür wird IGMP (Internet Group Management Protocol)[IETa] verwendet, welches auf IP aufsetzt. Normalerweise übernimmt der lokale Router die Aufgabe die Nachrichten zu vervielfältigen und den Empfängern zukommen zu lassen. Das führt dazu, dass im lokalen LAN Multicastnachrichten doch wieder an alle Rechner mittels Broadcast geschickt werden müssen, da ein HUB oder Switch die reservierten MAC Adressen nicht einem bestimmten Rechner zuordnen können. Einige Switches sind aber in der Lage bei IGMP Packeten durch „snooping“[IETc], also analysieren der Packetheader, die MAC Rechneradressen auszulesen und so interne Tabellen zu führen, welche lokalen Rechner in welcher Gruppe eingeschrieben sind. Dadurch wird ein Flooding an alle im LAN an diesen Switch angeschlossenen Rechner vermieden. Der Switch muss nur wissen, an welchem Port Nachrichten zum Router geschickt werden, denn dorthin muss er alle Multicastnachrichten weiterleiten.

Tunnel

Ein Tunnel ermöglicht es, Nachrichten aus dem lokalen LAN direkt in ein anderes, weiter entferntes, nicht mit dem ersten LAN direkt verbundenes, LAN zu schicken. Dazu eignet sich beispielsweise das L2TP(Layer Two Tunneling Protocol) [RFC99]. Hier werden lokale Nachrichten mittels eines L2TP Access Concentrator (LAC) in eine L2TP Datennachricht verpackt und über das Internet an das ZielLAN übertragen. Dort wird dann das Ausgangspacket von einem L2TP Network Server (LNS) wieder ausgepackt und in dem ZielLAN weitergeleitet. Version 3 von L2TP[IETb] kann nicht nur über UDP tunneln sondern auch direkt IP verwenden.

Vlan

Vlans (Virutal Local Area Network)[IEEa] ermöglichen das Gruppieren von Rechnern in Broadcastdomänen, auch wenn diese Rechner nicht direkt miteinander über ein LAN verbunden sind. Die Zugehörigkeit zu einem Vlan kann per Software verändert werden, ist somit flexibel zu handhaben.
Das ermöglicht eine Isolierung von Gruppen untereinander, das der Netzwerkverkehr nur innerhalb einer Vlangruppe übertragen wird. Vlan erleichtert also nicht nur die Administration sondern bringt auch Sicherheitsaspekte mit sich.
Es gibt zwei Varianten, wie die Gruppenzugehörigkeit der einzelnen Rechner verwaltet werden kann. Zum einen können in den Switches der LANs, in denen sich Mitglieder einer

Gruppe befinden, oder auch in den Bridges, die zwei LANs mit Mitgliedern verbinden, Tabellen vorgehalten werden. In den Tabellen kann dann entweder jedem Port eine Vlan-ID zugewiesen werden oder die Vlan-IDs können anhand der MAC Adressen der Mitglieder vergeben werden oder unterschiedlichen Schicht 3 Protokollen oder IP-Adressen werden unterschiedliche Vlan-IDs zugewiesen.
Alternativ dazu können auch im Ethernetheader Vlantags gesetzt werden, in denen eine Vlan-ID vom jeweiligen Sender eingetragen wird, welche dann von Switch/Bridge ausgelesen wird. Sollte der Sender keine Vlan-ID generiert haben, kann dies von Switch/Bridge anhand von Port/MAC des Senders nachgeholt werden. Für diese Variante von Vlan gibt es eine Erweiterung des Ethernetheaders um einen Vlan Protokoll- Identifier und drei Bit für ein Prioritätsfeld und zwölf Bit für die Vlan-ID und ein Bit für CFI(Canonical Format Indicator) welcher angibt ob sich im Nutzdatenbereich ein 802.5 Frame befindet oder ein 802.3 Frame. Die zwölf Bit der Vlan-ID ermöglichen theoretisch 4096 unterschiedliche Vlan-IDs. Drei dieser IDs sind aber reserviert, „0, 1 und 4095“. Es bleiben also 4093 Vlan-IDs zur freien Verfügung.
Auch kann auch jeder Sender gleichzeitig Mitglied in mehreren Vlangruppen sein. Er muss nur unterschiedliche Vlan-IDs im Ethernetheader angeben, sobald er eine Nachricht versendet

2.3.3 Vermittlungsschicht

Die Vermittlungsschicht hat die Aufgabe, ein Paket vom Sender zum Empfänger zu übertragen. Dies kann über mehrere Zwischenstationen, wie Router, geschehen. Es handelt sich also um eine Endpunkt zu Endpunkt Übertragung. Hierfür existieren in der Vermittlungsschicht eine Menge Routingprotokolle. Einige kann man hier nachlesen [Tan03].

IP

IP (Internet Protokoll Datagram) ist das im Internet verwendete Paketformat. Es besteht aus zwei Teilen, einem Header und einem Nutzdatenteil[Def81]. Mit IP können Rechner direkt adressiert werden oder es ist möglich einen Broadcast zu schicken. Damit dieser Broadcast nicht alle Rechner im Internet erreicht, bietet das IP Protokoll die Möglichkeit, mit Hilfe von Subnetzmasken Rechner in Gruppen zusammenzufassen und Broadcasts somit nur an bestimmte Rechner auszuliefern.

Multicast

Sollen nun nicht einzelne Rechner und auch nicht alle Rechner eines Subnetztes mit einer Nachricht erreicht werden gibt es die Möglichkeit, die Nachricht als Multicast[IET89] zu verschicken. Multicasts gehen nur an eine vorher ausgewählt Menge von Rechnern. Damit

ein Rechner Mitglied in einer Multicastgruppe werden kann muss er sich mittels des IGMP (Internet Group Management Protocol) [IETa] bei seinem lokalen Router anmelden. Damit Multicast funktionieren kann, muss der Router Multicastnachrichten unterstützen. Jede Multicastgruppe erhält eine der reservierten IP-Adressen im Bereich von 224.0.0.0 bis 239.255.255.255. Die Adressen 224.0.0.0, 224.0.0.1 sind reserviert. Mit 224.0.0.1 erreicht man alle Multicast benutzenden Rechner in einem direkt verbundenen Netzwerk, mit 224.0.0.2 erreicht man alle Router in einem direkt verbundenen Netzwerk, die an Multicast beteiligt sind.
Damit, wenn die Nachricht im LAN des Empfängers angekommen ist, diese nicht als Broadcast gesendet werden muss, gibt es eine Abbildungsfunktion für IP Multicastadressen auf MAC Multicastadressen. Da der Adressbereich für IP Multicast größer ist als der reservierte Bereich in den MAC Adressen kann es vorkommen, dass einige IP Adressen auf die selbe MAC Adresse übertragen werden.

Tunnel

Beim Tunneln werden IP Pakete eingepackt und dann an einen weiter entfernten Zielrechner in einem anderen Subnetz verschickt. Dort werden die Pakete wieder ausgepackt und dann lokal im dortigen Subnetz zugestellt. Dieses Verfahren eignet sich auch für VPN (Virtual Private Networking). Mögliche Protokolle hierfür sind Version 3 von L2TP[IETb] oder auch GRE (Generic Routing Encapsulation)[rfc].

Kapitel 3

Related Work

In diesem Kapitel werden andere Projekte vorgestellt, die sich mit dem Gruppieren von Netzwerkverbindungen beschäftigen. Bei diesen Projekten handelt es sich um Netzwerktestumgebungen. Somit müssen diese Projekte eine größere Anzahl an Gruppen verarbeiten können. Die Projekte haben dafür unterschiedliche Ansätze. Alle sind in der Lage sehr große skalierbare Testumgebungen zu erstellen. Alle Projekte setzen Virtualisierungstechniken ein, um große Zahlen an virtuellen Testknoten zu erschaffen. V-eM [[AH06b, AH06a]] verwendet XEN und überträgt die Nachrichten indem die MAC Zieladressen umgeschrieben werden. Bei MobiNet[MRBV05] werden die Testknoten und die Netzknoten getrennt gehalten. Übertragungen finden hier mittels eines Tunnels durch den Netzbereich der Testumgebung statt. Empower[ZN04] geht ähnlich vor. TVEE verwendet hingegen Vlans. Diese unterschiedlichen Ansätze werden hier vorgestellt.

3.1 Tunnel

3.1.1 MobiNet

MobiNet[MRBV05] ist eine verteilte Emulationsumgebung für mobile Anwendungen. Sie hat einige grundlegende Elemente von ModelNet[REFRENZ] übernommen. Mit MobiNet ist es möglich, beliebige Netztopologien zu emulieren und Protokolle oder Anwendungen auf diesen Netztopologien zu testen.

Als Grundlage verwendet MobiNet einen Emulationscluster, der in zwei Bereiche unterteilt wird. Einen „inneren“ Bereich und einen „Randbereich“. Im Randbereich kann handelsübliche PC Hardware zum Einsatz kommen, ebenso Notebooks oder sogar PDAs. Alles was als Anwendungshardware für die zu testende Applikation geeignet ist. Falls nun größere Szenarien benötigt werden, bietet MobiNet auch die Möglichkeit, mittels Virtualisierung Virtual Edge Nodes (VN) zu erzeugen. Mehrere VNs laufen dann auf

einem Randbereichsknoten und ermöglichen die mehrfache Ausführung der zu testenden Anwendung auf einem Randbereichsknoten.

Die Nachrichten, die ein Randbereichsknoten an einen anderen senden möchte, muss dieser durch den inneren Bereich des Clusters schicken. Im inneren Bereich des Clusters kommen Hardwareknoten mit großem Arbeitsspeicher zum Einsatz, die einen modifizierten FreeBSD Kernel besitzen. Diese inneren Knoten emulieren das gewünschte Verhalten der emulierten Netzwerktopologie. Diese Emulation wird erreicht, indem der FreeBSD Kernel um mehrere Module erweitert worden ist.

Nachdem eine Nachricht von einem Randbereichsknoten einen Inneren Knoten erreicht, wird es mittels ipfw an die Erweiterungsmodule von MobiNet weitergegeben. Zuerst muss das Routingmodul von MobiNet anhand einer Tabelle und der gewünschten emulierten Topologie entscheiden, wie viele Hops emuliert werden müssen. Dann wird die Nachricht durch unterschiedliche „pipes“ geschickt. Diese pipes emulieren die Verzögerung, die während eines Hops zu erwarten wäre. Hat eine Nachricht alle pipes durchquert, wird sie an den Zielrandbereichsrechner ausgeliefert.
Die unterschiedlichen pipes, die für die Emulation der Hops in der emulierten Netzwerktopologie notwendig sind, werden gleichmäßig auf alle Hardwareknoten im inneren Bereich verteilt, um auch die Last auf alle Hardwareknoten gleichmäßig zu verteilen. Um bewegliche mobile Geräte emulieren zu können, ist das Mobilitymodule von MobiNet in der Lage, die Routinginformationen für alle Randbereichsknoten dynamisch nach einem vorher definierten Bewegunsprofil anzupassen. Sollte sich dadurch die Anzahl der Hops zwischen zwei Randbereichsknoten ändern, werden Nachrichten einfach durch mehr pipes geleitet.

Damit Broadcasts nur die in einer Broadcastdomain befindlichen Randbereichknoten erreicht, muss das Routingmodul von MobiNet dafür Sorge tragen, dass diese Nachrichten nur an die korrekten Randbereichsknoten weitergeleitet wird. Der innere Bereich ist somit auch für die Vervielfältigung der Broadcastnachricht für alle erreichbaren Knoten zuständig.
Gruppierungen werden also anhand von Tabellen im Routingmodul realisiert. Das bedeutet, die Randknoten übergeben ihre Nachricht an den inneren Bereich der sie dann je nach Tabelleneintrag durch den inneren Bereich tunnelt bis hin zum Zielrandknoten.

3.1.2 Empower

EMPOWER (Emulation the Performance of wide area networks)[ZN04] ist eine verteilte Emulationsumgebung für mobile und kabelbasierte Anwendungen. Auch Empower ist in der Lage beliebige Netztopologien zu emulieren und Protokolle oder Anwendungen darauf zu evaluieren.

Empower verwendet mehrere PCs mit handelsüblicher Hardware. Diese werden in zwei Gruppen unterteilt. Eine Gruppe besteht aus den Testknoten, auf denen die zu testende Anwendung ausgeführt wird. Diese Testknoten haben eine Netzwerkkarte, über die sie mit dem LAN verbunden sind. Die andere Gruppe besteht aus den Emulationsknoten, diese haben mehrere vier Port Netzwerkkarten. Auf jedem Emulationsknoten werden nun virtuelle Router ausgeführt, die jeweils mit den Netzwerkkarten verbunden werden. Soll nun eine bestimmte Netztopologie emuliert werden, so wird jeder Router der Topologie auf einen virtuellen Router abgebildet. Dieser virtuelle Router erhält alle Eigenschaften des Routers in der Topologie inklusive der Anzahl der Netzwerkkarten mit denen er verbunden wird. Nun können auf einem Emulationsknoten mehrere virtuelle Router betrieben werden. Begrenzt wird es durch die Last auf dem Knoten und durch die Anzahl an möglichen Netzwerkports. Jeder virutelle Router verhält sich jetzt wie ein realer Router mit eigener Routingtabelle.

Um die Eigenschaften der zu emulierenden Netzwerktopologie darstellen zu können, wird jedem Netzwerkport ein Virtual Device (VD) vorgeschaltet. Diese VDs greifen den Netzwerkverkehr auf sobald er an dem Netzwerkport ankommt, leiten ihn dann durch mehrere interne Module bevor der Netzwerkverkehr wieder zurück auf die Hardwareebene gegeben wird. Die VDs enthalten unterschiedliche Module z.Bsp. für Verzögerung, MTU, Bandbreite, Verlust, Bitfehler, Aus-der-Reihe-Übertragung.

Bei der Verwendung mobiler Endgeräte werden diese wie Router als Virtual Mobile Nodes (VMN) auf die Emulationsknoten abgebildet. Jede VMN hat eine eigene Routingtabelle mittles der jede VMN selbst entscheiden kann ob sie eine Verbindung von einer anderen VMN zulassen darf oder ob sie das Paket verwerfen muss. Dadurch werden Pakete nur an bestimmmte Rechner weitergereicht.
Gruppierungen sind hier also mittels Routingtabellen in den virtuellen Routern in den Emulationsknoten möglich, die dafür sorgen, dass Broadcasts nur bestimmte Rechner erreichen können. Wie bei MobiNet sendet der „äußere“ Knoten eine Nachricht ab und der innere Bereich führt die Netzwerkemulation durch und liefert dann das Paket am Ziel ab.

3.2 MAC Adressen Modifikation

3.2.1 V-eM

V-eM[AH06b, AH06a] ist ein Netzwerkemulationscluster, der in der Lage ist beliebige Netzwerktopologien zu emulieren. In diesen Netzwerktopologien können dann beliebige Protokolle oder Anwendungen evaluiert werden.
Der Cluster besteht aus handelsüblicher PC Hardware und Netzwerkswitches.

Jeder Hardwareknoten des Clusters führt eine virtuelle Maschine aus, verwendet wird XEN[BDF+03]. Mit Hilfe von XEN ist es möglich, mehrere virtuelle Container auf einem Hardwareknoten zu erstellen, was das Betreiben mehrerer virtueller Knoten (vNode) auf einem Hardwareknoten (pNode) ermöglicht. Jeder vNode hat unter anderem seinen eigenen Kernel, Speicher, Netzwerkstack, Userzugriffsmöglichkeit. In jedem vNode kann ein eigenes Betriebssystem mit eigenen Anwendungen installiert werden. Um den Virtualisierungsoverhead zu verringern, wird XEN im Paravirtualisierungsmodus betrieben.
XEN erstellt einen Container zu administrativen Zwecken, Domain0, und mehrere Userdomains in denen eigene Betriebssysteme installiert werden können. Jede Userdomain hat eine oder mehrere eigene virtuelle Netzwerkkarten, die über ein virtuelles Interface mit der Domain0 verbunden sind. Der Netzwerkverkehr läuft nun über eine Softwarebridge in Domain0 zwischen den virtuellen Netzwerkkarten der Userdomains oder über eine weitere virtuelle Netzwerkkarte in Domain0, die über den XEN Hypervisor mit der physikalischen Netzwerkkarte verbunden ist. Um gleiche Laufzeiten zwischen den vNodes gewährleisten zu können, wird sämtlicher Netzwerkverkehr zuerst aus XEN über die physikalische Netzwerkkarte herausgeleitet und danach über eine zweite virtuelle Netzwerkkarte wieder zurück.
Unicast Nachrichten werden so mittels ARP (Address Reolution Protocol) korrekt ausgeliefert. Multicastnachrichten werden auch korrekt verarbeitet. Nur einige bestimmte Multicastadressen, z.Bsp. die Adresse in die sich alle Router automatisch eintragen, sind problematisch. Hier würden auch Router Nachrichten erhalten, die in einer möglichen Netzwerktopologie diese Nachrichten nicht erhalten sollten. Diese speziellen Multicastübertragungen werden in V-eM manupuliert. Hierfür sind bei V-eM die virtuellen Brücken in der XEN Domain0 so modifiziert worden, dass sie die MulticastMACadresse mit der gewollten Zieladresse überschreiben und dann erst weiter geben.
Broadcasts stellen so aber immer noch ein Problem dar. Falls also ein Szenario verwendet wird, in dem Broadcasts eine Rolle spielen, werden intern allen Broadcastlinks MulticastMACAdressen zugewiesen. Die Broadcastadresse wird dann von der virtuellen Brücke mit der entsprechenden MulticastMACAdresse überschrieben. Broadcasts werden dann als Multicast an die jeweilige Multicastgruppe gesendet. Der verwendete Switch lernt dann die Gruppenzugehörigkeit mittels IGMP-snooping (siehe Kapitel zwei) und kann die Pakete korrekt verteilen.
Gruppierungen werden also mittels Multicastgruppen gebildet.

3.3 Vlan

3.3.1 TVEE

Die Emulationsarchitektur TVEE (Time virtualized emulation Evnvironment) ist eine verteilte Netzwerkemulationsumgebung, die eine beliebige Netzwerktopologie emulieren und Programme und Anwendungen anhand dieser emulierten Netzwerktopologie evaluieren kann.
Als Grundlage verwendet TVEE einen Cluster. Auf jedem Hardwareknoten dieses Clusters wird nun eine Virtualisierung auf zwei Ebenen gestartet. Zuerst wird die virtuelle Maschine XEN ausgeführt. XEN setzt mit Hilfe des Hypervisors direkt auf der Hardware auf und erstellt eine Administrations-Domäne Domain0 und Userdomänen. Innerhalb der Userdomänen wird die zweite Virtualisierungsebene mit Hilfe der Virtuellen Maschine OpenVZ realisiert. Diese unterteilt das in XEN laufende Gastbetriebssystem Linux in einzelne von einander isolierte Container, die virtuellen Knoten (vNodes).
Jeder dieser virtuellen Knoten besitzt eine oder mehrere virtuelle Netzwerkkarten. Diese sind mit einer virtuellen Brücke im Gastbetriebssystem in der XENdomäne verbunden und können so miteinander kommunizieren. Diese virtuelle Brücke ist wiederum mit einer oder mehreren virtuellen Netzwerkkarten in der XEN Domäne verbunden. Zu diesen virtuellen Netzwerkkarten in der XENUserdomäne korrespondieren virtuelle Interfaces in der Domain0, die mit einer virtuellen Brücke in der Domain0 verbunden sind. Die Softwarebridge in der Domain0 ist über einen nativen Gerätetreiber mittels des Hypervisors mit der physikalischen Netzwerkkarte verbunden(siehe Abbildung 3.1).
Aufgrund dessen, daß die vNodes mittels virtueller Brücken verbunden sind, hängen alle am selben LAN. Falls nun einige vNodes zu einer Gruppe zusammengefasst werden sollen, gibt es mit Broadcastsendungen innerhalb dieser Gruppe Probleme. Bei TVEE wird hier Vlan verwendet, um dieses Problem zu lösen. Alle vNodes, die in der selben Gruppe sind, verwenden die selbe Vlan-ID im Ethernetheader. Der Switch, der alle Hardwareknoten miteinander verbindet, versteht Vlan und übermittelt alle Nachrichtenpakete entsprechend.

Somit hat der Ansatz von TVEE mit Vlans gegenüber den Ansätzen von MobiNet und Empower den klaren Vorteil, dass der Hardwareswitch die Verteilung von Broadcastnachrichten übernehmen kann und kein Rechner den Aufwand des Mehrfachsendens einer Nachricht zu erbringen hat.
Das Verfahren von V-eM ist viel versprechend, nur haben heutzutage viele Switches eine sehr begrenzte Implementierung an MAC Multicastadressen, die meistens vergleichbar mit der Begrenzung an implementierten Vlangruppen ist, und somit ebenso wie der Ansatz von TVEE auch hier eine Beschränkung der Gruppenanzahlen existiert.

Abbildung 3.1: TVEE mit Vlandevices

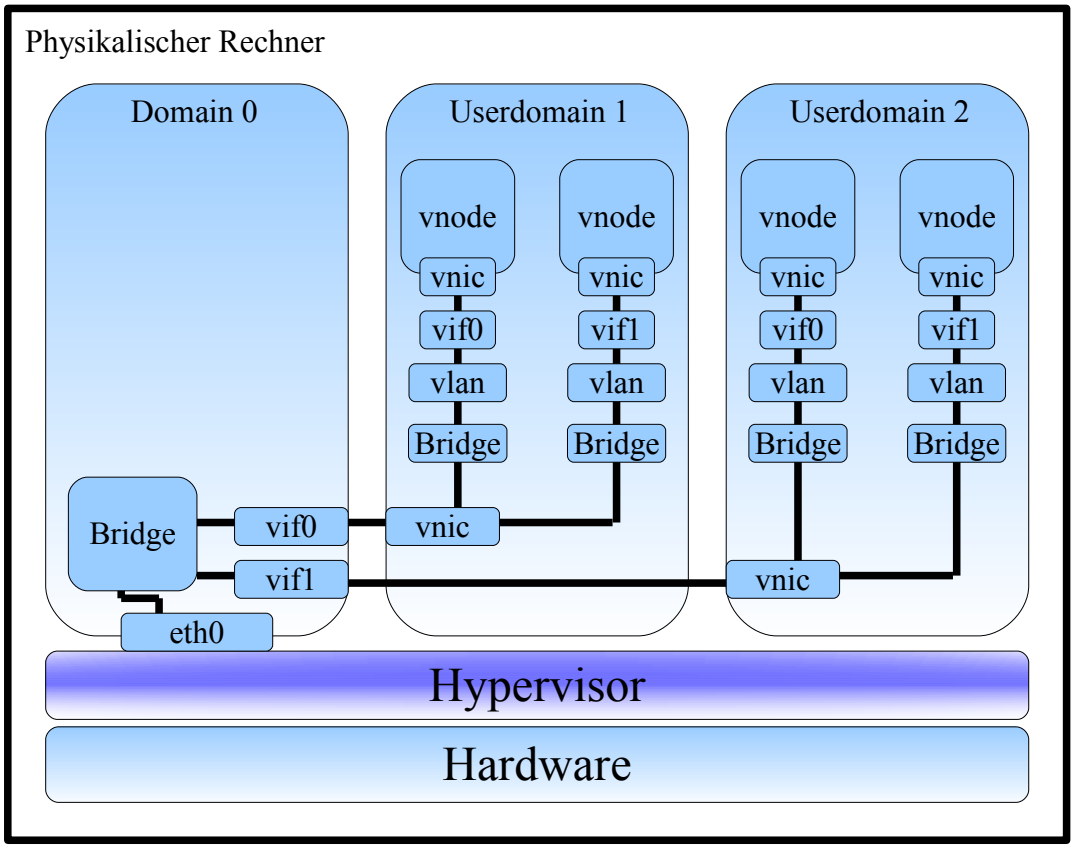
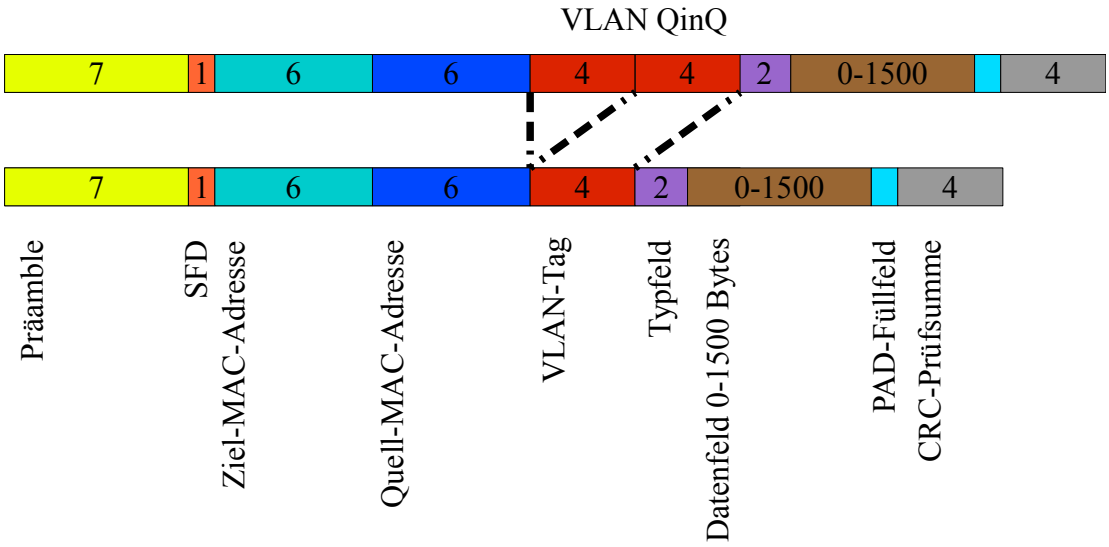


Abbildung 3.2: Ethernetheader mit Vlan und Vlan QinQ



3.4 Verschachteltes Vlan

Eine Einschränkung von „tagged“ Vlan, also Vlan unter Verwendung der Vlan-ID im Ethernetheader, ist die Größenbeschränkung des Vlan-ID Feldes im Ethernetheader auf zwölf Bit. Dies ermöglicht die Verwendung von 4096 IDs. Da drei IDs reserviert sind (die IDs 0,1,4095)[IEEa] bleiben 4093 IDs zur Verwendung übrig. Mit der stetig anwachsenden Netzwerkinfrastruktur ist eine Erweiterung dieses Standards erwünscht.

Ein Ansatz ist das verschachtelte Vlan auch 802.1QinQ genannt. Bei Vlan wird zwischen MAC Quelladresse und Ethertype Feld im Ethernetheader ein weiteres Feld eingefügt. Dieses besteht aus einem 16Bit Protokoll Identifier Feld (8100 für 802.1Q) und dem 16 Bit langen Tag Control Identifier(TCI). Der Tag Control Identifier besteht aus einem drei Bit Feld für eine Prioritätsangabe (PCP Priority Code Point), einem ein Bit Feld zur Identifizierung eines möglichen 802.5 (Token Ring) Rahmens und der zwölf Bit großen Vlan-ID(siehe Abbildung 3.2).
Bei 802.1QinQ wird zwischen der MAC Quelladresse und dem durch Vlan eingefügten Protocoll Identifier ein weiterer Block aus Protocoll Identifier und Tag Control Identifier eingefügt. Der Protokoll Identifier muss hier auf einen anderen Wert als 8100 gesetzt werden, Cisco verwendet 9100 oder 9200[CIS]. Der Tag Control Identifier kann nun Infor-

mationen für eine neue Vlan-ID enthalten. Durch die Verschachtelung ist theoretisch eine Anzahl von 4093*4093=16.752.649 Vlan-IDs möglich.

Ein weiterer Vorteil der Verschachtelung von Vlans im Vergleich zur Vergrößerung des Vlan-ID Feldes ist die Kapselung von Vlangruppen. Wenn ein Internetprovider Kunden hat, die mehrere Vlan-IDs verwenden, kann er durch Hinzufügen eines verschachtelten Vlan Headerfeldes die Vlan-ID des Kunden erhalten und seine eigene hinzufügen, um die Datenpakete im eigenen Netz zum richtigen Empfänger zu leiten. Bevor die Pakete an den Kunden übergeben werden, wird einfach das zusätzliche Vlan Headerfeld wieder entfernt. Dadurch kommt es zu keinen Problemen, falls zwei Kunden die selben Vlan-IDs in ihren lokalen Netzen verwenden sollten - die Netze der Kunden bleiben voneinander getrennt. Die Verschachtelung ermöglicht also eine Datenkapselung nur durch Hinzufügen eines Vlanblocks in den Ethernetheader.
Switches interpretieren immer den äußeren Tag, der sich näher an der MAC Quelladresse befindet. Ist der äußere Tag nicht mehr von Nöten, kann er entfernt werden und der innere Tag wird zum äußeren Tag. Der äußere Tag wird auch als service oder metro Tag bezeichnet und der innere Tag als customer Tag.
Theoretisch ist es auch möglich, beliebig viele Vlan Tags hintereinander zu schachteln und so eine beliebig große Anzahl an Gruppierungen zu erstellen.

Kapitel 4

Entwurfskriterien

Es gibt unterschiedliche Möglichkeiten um Netzwerkverbindungen zu gruppieren. Einige kann man im Kapitel „Related Work“ sehen. Jede Variante hat ihre Vorteile bringt aber auch eigene Probleme und Beschränkungen mit sich. Diese Vor- und Nachteile werden in diesem Kapitel näher betrachtet. Das Ziel ist es, die Beschränkung auf 4096 Gruppen, die durch die Verwendung von Vlan vorgegeben sind, aufzuheben, oder zumindest einen Weg zu finden mehr als 4096 Gruppen zu verwenden.

4.1 Mögliche Vorgehensweisen

In diesem Kapitel wird die Architektur von TVEE als Grundlage genommen, da die Implementierung auch unter TVEE erfolgen soll. Wie im Kapitel zwei detailliert beschrieben, ist TVEE eine verteilte Emulations Architektur, die auf einem Cluster zum Einsatz kommt. TVEE hat zwei Ebenen der Virtualisierung, auf der Hardware mittels eines Hypervisors aufsetzend XEN und in den einzelnen Domänen von XEN unterteilt OpenVZ das XEN Gastbetriebssystem in einzelne Container, welche die virtuellen Knoten der Testumgebung darstellen.
Die einzelnen virtuellen Knoten besitzen eine oder mehr virtuelle Netzwerkkarten. Sie können innerhalb des XEN Gastbetriebssystems mittels virtueller Brücken direkt verbunden werden. Das bedeutet die virtuellen Knoten können direkt mittels virtuellem Routing [MGWR07] innerhalb des XEN Gastbetriebssystems miteinander kommunizieren.
Die virtuellen Knoten können auch mittels virtueller Brücken mit virtuellen Netzwerkkarten der XEN Userdomäne verbunden werden, in der sie sich befinden. Diese virtuellen Netzwerkkarten der XEN Userdomäne sind direkt mit virtuellen Interfaces der administrativen Domain0 verbunden.
Innerhalb der Domain0 ist es nun möglich mittels virtueller Brücken beliebige virtuelle Interfaces zusammenzuschalten. Dadurch kann innerhalb von XEN jeder virtuelle Knoten mit allen anderen virtuellen Knoten eines Hardwareknotens verbunden werden.

Da der Cluster mehrere Hardwareknoten beheimatet, müssen auch Netzwerkverbindungen zwischen den verschiedenen Hardwareknoten ermöglicht werden. Dafür können in der Domain0 die virtuellen Brücken virtuelle Interfaces nicht nur untereinander verbinden sondern auch über den Hypervisor mit der physikalischen Netzwerkkarte des Hardwareknotens.
Die physikalischen Netzwerkkarten der Hardwareknoten sind mittels eines hoch performanten Switches miteinander verbunden. Alle virtuellen Knoten sind über virtuelle Brücken und den Hardwareswitch verbunden, befinden sich also alle im selben LAN.

Ziel ist es nun, mehrere virtuelle Knoten zu Gruppen zusammenzufassen. Den virtuellen Knoten innerhalb einer Gruppe soll es ermöglicht werden mittels Broadcast nur die anderen Knoten innerhalb ihrer Gruppe zu erreichen. Eine mögliche Lösung wäre den Hardwareswitch zu überarbeiten, so dass er eine größere Anzahl an Vlan-IDs verarbeiten kann. Dazu müsste die Hardware des Switches verändert werden, was einen großen finanziellen Aufwand bedeuten würde.
Eine weitere Möglichkeit ist, mittels MAC Adressmodifikation innerhalb der virtuellen XEN Brücken bei Broad- und Multicastsendungen die Adresse durch die Adressen der virtuellen Zielknoten zu ersetzen. Dies bedeutet einen Mehraufwand innerhalb der virtuellen Brücken, da nicht der Hardwareswitch die Vervielfältigung der Nachrichten übernimmt sondern die virtuelle Brücke dies übernehmen muss.
Als nächstes gibt es die Möglichkeit, alle Broadcastsendungen als Multicastsendungen zu verschicken. Dazu müssen bei allen Broadcastsendungen von den jeweiligen virtuellen Brücken mittels MAC Adressmodifikation die Zieladressen in MAC Multicastadressen umgeschrieben werden. Hier müssen die Multicastgruppen verwaltet und die Mitglieder erkannt werden.
Optional kann auch die MAC Adressmodifikation verbunden werden mit IP Multicast um die Anzahl der Mehrfachsendungen pro virtueller Brücke zu reduzieren.
Auch Tunnel können nützlich sein. Sie können mit Vlan kombiniert werden und Pakete zwischen Hardwareknoten transportieren.
Auch die Erweiterung von Vlan, das Geschachtelte Vlan 802.1QinQ ist eine Option. Hier werden die inneren Vlan-IDs innerhalb der virtuellen Maschinen verwendet und die äußeren Vlan-IDs ziwschen den Hardwareknoten.
Nun folgend werden die einzelnen Varianten detailierter vorgestellt.

4.2 Hardwareimplementierung

Ein Weg die Gruppenanzahl zu erhöhen ist die Eigenschaften des verwendeten Hardwareswitches zu verändern oder einen neuen Switch zu kreieren. Der Switch müsste ein größeres Datenfeld für Vlan-IDs akzeptieren um auf diese Weise mehr Vlan-IDs verarbeiten zu können. Außerdem müssten dann alle eingesetzten Netzwerkstacks angepasst werden.

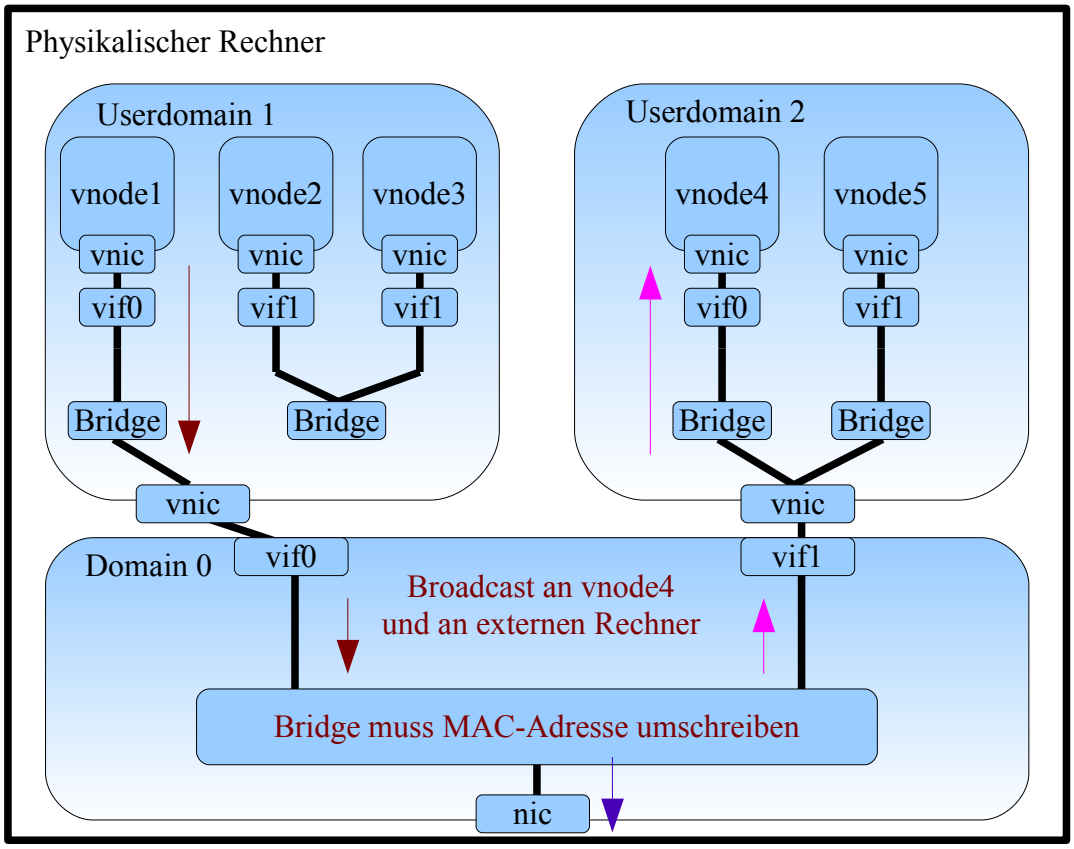
Dadurch wäre eine beliebig große Anzahl an Vlangruppen möglich. Es würde kein Overhead entstehen und alle Gruppen wären direkt adressierbar.

Die Kosten und der Zeitaufwand um ein derartiges Vorhaben realisieren zu können, ist allerdings sehr groß und wohl nicht erstrebenswert.

4.3 MAC Adressen Modifikation

Das Problem bei Gruppierungen ist der Multicast und der Broadcast. Ein Ansatz ist, in den virtuellen Brücken der XEN Domain0 die MAC Zieladressen von Nachrichten der virtuellen Knoten umzuschreiben, wenn ein Broadcast oder Multicast geschickt wurde. Dazu wird in den virtuellen Brücken eine Tabelle vorgehalten, welche virtuellen Rechner sich in welcher Gruppe befinden. Empfängt die virtuelle Brücke nun einen Multicast oder Broadcast ersetzt sie die Multi- bzw. Broadcastadresse durch die eigentlichen Zieladressen der virtuellen Rechner, die erreicht werden sollen. Außerdem übernimmt die virtuelle Brücke auch das notwendige Mehrfachsenden der Nachricht um mehrere Rechner erreichen zu können(siehe Abbildung 4.1).
Die virtuelle Brücke, die das Umschreiben und Mehrfachsenden übernehmen muss wird dadurch natürlich belastet was die Gesamtbelastung der verteilten Emulationsarchitektur negativ beeinflusst. Auch sind die Tabellen in den virtuellen Knoten relativ groß, da alle MAC Adressen aller virtuellen Rechner gespeichert sein müssen, die in einem Bereich liegen, in dem potentiell Adressen umgeschrieben werden müssen.
Wichtig für die Frage der zusätzlichen Belastung der virtuellen Brücke ist Betrachtung welche virtuellen Brücken überhaupt Broadcastnachrichten anpassen müssen. Eine Gruppe virtueller Knoten innerhalb der selben XEN Domain könnte bei diesem Ansatz mittels einer virtuellen Brücke innerhalb des XEN Gastbetriebssystems verbunden werden und es müssten keine Broad- oder Multicastadressen umgeschrieben werden.
Eine Gruppe Rechner, die sich innerhalb des selben Hardwareknotens befindet, also unter Umständen in zwei verschiedenen Userdomänen von XEN können über virtuelle Brücken in der XEN Domain0 verbunden werden und es würde auch kein Umschreiben von Adressen notwendig. Problem hier ist nur die stark begrenzte Anzahl an virutellen Netdevices in XEN. Deshalb ist dieses Vorhaben für mittel bis große Szenarien nicht anwendbar. Folglich müssen die Multi- und Broadcastadressen in diesem Fall von den virtuellen Brücken umgeschrieben und einzeln versandt werden.
Eine Gruppe, die sich auf mehreren Hardwareknoten befindet muss über den Hardwareswitch kommunizieren und es muss deshalb ein Umschreiben der Broadcast- und Multicastadressen erfolgen.
Eine virtuelle Brücke muss also dann umschreiben, falls die Rechner einer Gruppe nicht ausschließlich innerhalb einer XEN Domäne angesiedelt sind. Die Anzahl der Gruppen wird hier nur durch die Größe der Tabellen innerhalb der virtuellen Brücken begrenzt.

Abbildung 4.1: Mehrfachsenden durch MAC Adressenmodifikation

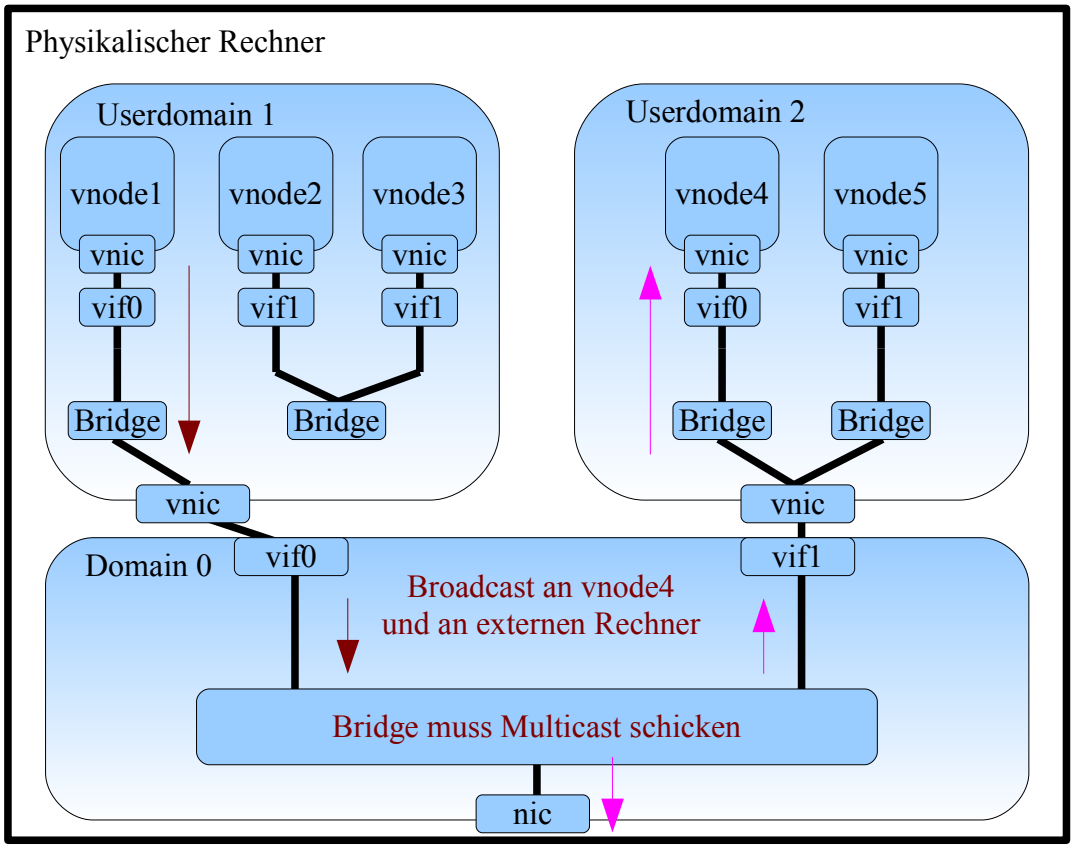


4.4 MAC Multicast

Ein alternativer Ansatz ist, anstatt alle virtuellen Knoten direkt und einzeln anzusprechen, kann auch MAC Multicast verwendet werden. Hier müssen sich alle Mitglieder einer Gruppe in eine Multicastgruppe eintragen. Die virtuellen Brücken müssen dann die Broadcast Nachrichten abändern. In jeder virtuellen Brücke muss eine Tabelle mit den Mitgliedern der einzelnen Gruppen vorgehalten werden. Empfängt eine virtuelle Brücke eine Broadcastnachricht, muss sie Anhand dieser Tabelle die Broadcastadresse durch eine entsprechende MAC Multicastadresse ersetzen. Diese Multicastnachrichten erreichen alle notwendigen Zieladressen(siehe Abblidung 4.2).
Die virtuellen Brücken und der Hardwareswitch erlernen die Gruppenzugehörigkeit der virtuellen Knoten mittels IGMP [IETa] snooping [IETc]. Das bedeutet, tritt ein virtueller Knoten einer Gruppe bei oder ändert er seine Zugehörigkeit, muss er dass mittels IGMP Nachrichten bekannt geben.
Die Tabellen, die in den virtuellen Brücken vorgehalten werden müssen, sind relativ groß, da alle MAC Adressen aller virtuellen Rechner, die in einem Bereich liegen, in dem potentiell Adressen umgeschrieben werden müssen, gespeichert sind.

Eine Gruppe virtueller Knoten innerhalb der selben XEN Domain könnte auch bei diesem Ansatz mittels einer virtuellen Brücke innerhalb des XEN Gastbetriebssystems verbunden werden und es müssten auch hier keine Broadcastadressen umgeschrieben werden.
Eine Gruppe Rechner, die sich innerhalb des selben Hardwareknotens befindet, also innerhalb der selben XEN Maschine, möglicherweise auch in zwei verschiedenen Userdomänen von XEN, können über virtuelle Brücken in der XEN Domain0 verbunden werden und es würde auch kein Umschreiben von Adressen notwendig. Aber wie bei dem vorherigen Ansatz müssen aufgrund der zahlenmäßig stark begrenzten virtuellen Netdevices in XEN die Broadcastnachrichten doch umgeschrieben werden.
Bei einer Gruppe, die sich über mehrere Hardwareknoten erstreckt und über den Hardwareswitch kommunizieren muss, muss ein Umschreiben der Broadcastadresse erfolgen.
Die virtuelle Brücke muss allen an ihr angeschlossenen Gruppenmitgliedern eine Nachricht schicken und eine Multicastnachricht, um die restlichen virtuellen Rechner zu erreichen, die an eine andere virtuelle Brücke angeschlossen sind oder sich auf einem anderen Hardwareknoten befinden.
Die möglichen IP-Multicastgruppen sind im Adress-Bereich 224.0.0.0 bis 239.255.255.255 (270 Millionen) angesiedelt. Es werden nun die unteren 23 Bit der IP-Adresse in die MAC Multicastadresse 01-00-5e-00-00-00 übernommen. Das ergibt Adressen von 01-00-5e-00-00-00 bis 01-00-5e-7f-ff-ff.

Abbildung 4.2: Mehrfachsenden durch Multicast



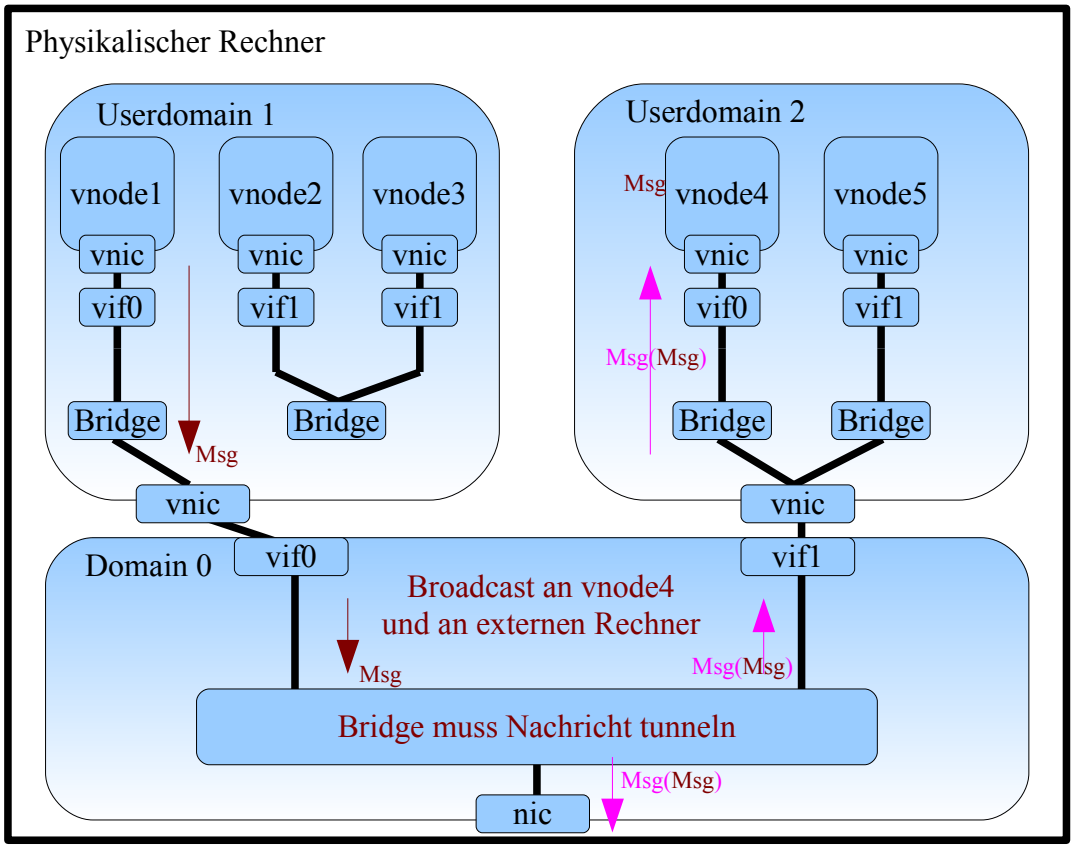
4.5 Tunnel

Bei diesem Ansatz kann man Tunnel auf MAC Adressebene oder auch IP Tunnel einsetzen. Sollte eine virtuelle Brücke eine Broadcastnachricht oder eine Multicastnachricht von einem virtuellen Knoten empfangen, dann muss sie die Nachricht in eine andere Nachricht einpacken und über das Netz direkt an den Empfänger schicken, der sie dann wieder auspacken muss (siehe Abbildung 4.3). Hierfür gibt es unterschiedliche Protokolle wie z.Bsp. L2TP(Layer Two Tunneling Protocol) [RFC99], die Version 3 von L2TP [IETb] oder auf IP Ebene auch GRE (Generic Routing Encapsulation) [rfc]. Den Empfänger muss die virtuelle Brücke anhand einer in der virtuellen Brücke vorgehaltenen Tabelle auswählen. Die Tabelle ist relativ groß, da alle virtuellen Rechner gespeichert sein müssen, die potentiell einen Tunnel benötigen.
Alle innerhalb der selben virtuellen XEN Domäne befindlichen Rechner können mittels einer virtuellen Brücke innerhalb des XEN Gastbertriebssystems verbunden werden und die Broadcasts mittels virutellem Routing austauschen, brauchen also keine Tunnel.
Die virtuellen Rechner, die Mitglied einer Gruppe sind die innerhalb eines Hardwarerechners angesiedelt ist, können über virtuelle Brücken in der XEN Domain Broadcasts austauschen. Die Anzahl der verfügbaren virtuellen Netdevices unter XEN ist leider sehr begrenzt so dass im Realfall die Broadcastadressen getunnelt werden müssen.
Virtuelle Rechner, die in Gruppen Mitglied sind, die über mehrere Hardwareknoten verteilt sind müssen die Tunnel in Anspruch nehmen.
Die einzelnen Nachrichten durch die Tunnel senden muss die virtuelle Brücke, an die der Sender angeschlossen ist.

4.6 Vlan

Bei der Verwendung von Vlans werden allen Gruppen Vlan-IDs zugeteilt. Dies geschieht am einfachsten bei der Verwendung von „tagged“ Vlan, bei dem eine Vlan-ID in den Ethernetheader geschrieben wird. Die einzelnen virtuellen Knoten schreiben die Vlan-ID ihrer Gruppe in den Header der Nachrichten, die sie verschicken. Anhand dieser IDs können die virtuellen Brücken in XEN entscheiden wohin eine Nachricht gehen soll, indem sie aus ankommenden Nachrichten lernen, welche Vlan-IDs auf welchem Port angekommen ist.
Wie in Kapitel zwei beschrieben, sind die Vlan-IDs auf 4096 begrenzt. Das liegt an der Größe des Feldes im Ethernetheader, das die Vlan-ID enthält. Zwischen MAC Quelladresse und Etype des Ethernetrahmens wird die Protokoll-Idenftifier und der Tag Control Identifier, bestehend aus einem drei Bit Feld für eine Prioritätsangabe (PCP Priority Code Point), einem ein Bit Feld zur Identifizierung eines möglichen 802.5 (Token Ring) Rahmens und der zwölf Bit großen Vlan-ID, eingefügt.

Abbildung 4.3: Mehrfachsenden realisiert mit Tunnel



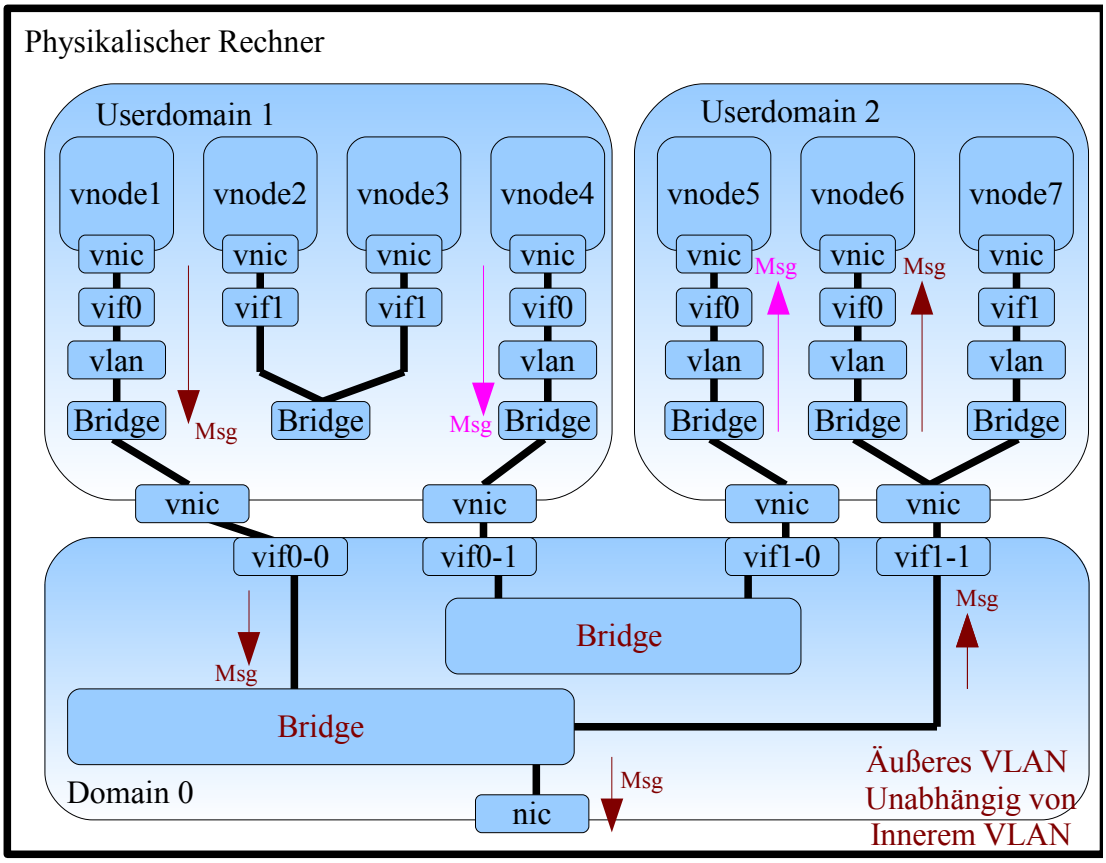
Soll die Kapazität über die 4096 Vlan-IDs hinaus erhöht werden, so kann mittels Vlan Q-in-Q (Verschachteltes Vlan) eine weitere Vlaninstanz in den Ethernetheader eingebracht werden zwischen der schon vorhandenen und der MAC Quelladresse.

Virtuelle Knoten, die sich innerhalb einer XEN Domäne befinden, können auch bei Vlan mittels virtuellem Routing innerhalb des XEN Gastbetriebssystems kommunizieren und bräuchten eigentlich keine Vlan-ID.
Die virtuellen Knoten, die Mitglied in einer Gruppe sind, die sich auf einem Hardwareknoten über mehrere XEN Domänen verteilt, benötigen eine Vlan-ID, damit Broadcasts ihre Ziele erreichen können.
Mitglieder von Gruppen, die über mehrere Hardwareknoten verteilt sind, benötigen ebenfalls eine Vlan-ID.
Das ermöglicht 4096 Vlan-IDs pro Hardwareknoten. Das sind zusammen 65536 auf allen 16 Hardwareknoten. Hinzu kommen die 4096 Vlan-IDs zwischen den Hardwareknoten, zusammen 69632(siehe Abbildung 4.4).
Zur Realisierung dieses Ansatzes, werden zwei unterschiedliche virtuelle Brücken innerhalb jeder Domain0 von XEN benötigt. Alle virtuellen Interfaces, die zu virtuellen Rechnern führen, deren Gruppen sich innerhalb des Hardwareknotens befinden, werden mit einer virtuellen Brücke verbunden.
Alle virtuellen Interfaces, die zu virtuellen Rechnern in den XEN Userdomains führen, die zu Gruppen gehören, die über mehrere Hardwareknoten aufgeteilt sind, werden mit einer zweiten virtuellen Brücke in der Domain0 verbunden.

Beim Einsatz von 802.1 Q-in-Q gibt es mehrere Möglichkeiten. Man kann theoretisch beliebig viele Verschachtelungen von Vlan-IDs vornehmen.
Bei der Verwendung von zwei Vlan-IDs in einem Ethernetheader können mittels der „äußeren“ Vlan-ID 4096 unterschiedliche Gruppen aus verschiedenen Kombinationen der Hardwareknoten erzeugt werden. Z.Bsp. Zehn virtuelle Rechner einer Gruppe liegen auf Hardwareknoten eins, fünf virtuelle Rechner der Gruppe liegen auf Hardwareknoten zwei und sechs virtuelle Rechner der Gruppe liegen auf Hardwareknoten vier. Hier wird eine äußere Vlan-ID benötigt, z.Bsp. ID fünf. In Vlangruppe fünf werden dann Hardwarerechner eins, zwei und vier Mitglied.
Nun gibt es viele unterschiedliche Kombinationsmöglichkeiten, zuerst gibt es 16 Hardwareknoten, dann gibt es 120 mögliche zweier Kombinationen aus Hardwareknoten, dann 560 mögliche 3er Kombinationen und so weiter. Die Zahlen lassen sich mittels des Binomialkoeffizienten berechnen.
Insgesamt gibt es für 16 Hardwareknoten 65535 mögliche Gruppenkombinationen aus Hardwareknoten.

Zu jeder aus den 65535 ausgewählten 4096 Gruppen ist es nun mittels der inneren

Abbildung 4.4: Eine Ebene Vlan IDs



Vlan-ID möglich weitere 4096 Gruppen zu bilden. Das bedeutet zum Beispiel es können 4096 Gruppen aus virtuellen Rechnern gebildet werden innerhalb der oben erwähnten Hardwarerechnerkombination eins,zwei und vier.

Für diesen Ansatz, werden innerhalb der XEN Domain0 zwei Ebenen von hintereinander geschalteten virtuellen Brücken benötigt. Die erste virtuelle Brücke reicht die Nachricht anhand der äußeren Vlan-ID an die entsprechende virtuelle Brücke der zweiten Ebene weiter. Diese entfernt die äußere Vlan-ID aus dem Ethernetheader. Dadurch wird die innere Vlan-ID zur äußeren Vlan-ID. Die virtuelle Brücke reicht nun anhand dieser Vlan-ID die Nachricht entsprechend an die virtuellen Rechner weiter, die Mitglied dieser Vlangruppe sind.

Mit diesem Ansatz sind maximal 4096*4096=16.777.216 Gruppen möglich, sofern immer 4096 innere Gruppen auf 4096 äußere Gruppen kommen.

Beim Einsatz von drei oder mehr Vlan-ID Verschachtelungen steigt die Zahl der möglichen Gruppen weiter an. Es werden innerhalb der XEN Domain0 dementsprechend mehr Ebenen an virtuellen Brücken benötigt, wobei jede weitere jeweils eine Vlan-ID Instanz auswertet und entfernt.
Das einzige Problem, dass mit dem Hinzufügen weiterer Vlan-IDs bestehen bleibt, ist die Fähigkeit des Hardwareswitches nur 4096 Vlan-IDs verarbeiten zu können. Das bedeutet von den 65535 möglichen Hardwareknotenkombinationen sind immer nur 4096 abbildbar. Aufgrund dieses Problems ist es notwendig die Auswahl der 4096 Hardwareknotenkombinationen möglichst effizient zu treffen. Ab Kapitel 6 werden hierfür mehrere Ansätze vorgestellt.

4.7 Einschränkungen durch Hardware

Viele der auf dem Markt erhältlichen Switches haben begrenzte Möglichkeiten. Die Unterstützung für mögliche Vlan-IDs und Multicastgruppen entspricht häufig nicht der nach Definition maximal möglichen Anzahl. Das selbe gilt für die Anzahl an MAC Adressen, die ein Switch intern pro Port speichern kann. Dies führt dazu, dass obwohl die maximal mögliche Anzahl an Multicastgruppen sehr groß ist im Vergleich mit verfügbaren Vlan-IDs, viele Switches gleich viele oder sogar mehr Vlan-IDs ermöglichen als Multicastgruppen. Hier einige aktuelle Beispiele:

Beispielauswahl an handelsüblichen Switches

Switch	Vlangruppen	Multicastgruppen	MAC Adressen	QinQ
Arista 7148S	4094	4500	16000	ja
D-link DGS-6600	4000	1000	32000	ja
Netgear GSM7352S-200	4000	1000	8000	ja
Netgear GSM7252PS	1000	1000	8000	ja
Cisco Catalyst 4900M Switch	4096	56000	55000	ja
Cisco ME 6524	4096	64	96000	ja

Kapitel 5

Konzept für Umsetzung der Vlanerweiterung

In diesem Kapitel wird die Umsetzung von Vlan QinQ genauer beschrieben. Welche Elemente notwendig sind und welche Probleme entstehen können. Da in der realen Umsetzung in XEN die virtuellen Brücken nicht direkt an dem Prozess der Vlan-ID Vergabe beteiligt sind, ist auch hier ein leicht angepasstes Vorgehen von Nöten.

5.1 Komponenten

Die Komponenten, die für die Vlanerweiterung benötigt werden, sind erst einmal die, die auch für normalen Vlanbetrieb notwendig sind. In OpenVZ, werden für die virtuellen Knoten virtuelle Netzwerkkarten zur Verfügung gestellt. Diese werden dann mittels virtueller Brücken im XEN Gastbetriebssystem mit virtuellen Vlandevices verbunden. Diese Vlandevices hängen an den virtuellen Netzwerkkarten der XEN Userdomäne.
Um eine Verbindung zu der administrativen Domain0 von XEN zu erhalten, werden in der Domain0 virtuelle Interfaces benötigt. Diese verbinden die virtuellen Netzwerkkarten der XEN Userdomains mit den virtuellen Brücken in der XEN Domain0.
Die virtuellen Brücken sind nun noch mit der physikalischen Netzwerkkarte über eine virtuelle Netzwerkkarte verbunden.

Zusätzlich zu dem normalen Vlanbetrieb wird nun eine weitere Ebene benötigt, die mit den „äußeren“ Vlan-IDs arbeiten kann. Dazu kann aber keine virtuelle Brücke verwendet werden, da diese unter XEN nicht die Bearbeitung der Vlan-ID übernimmt. Deshalb sind zusätzliche Vlandevices notwendig, die an der virtuellen Netzwerkkarte, die mit der physikalischen Netzwerkkarte verbunden ist, angebracht werden.

Das sind alle zusätzlichen Komponenten, die für einen Q-in-Q Betrieb mit zwei ver-

schachtelten Vlan-IDs notwendig sind.

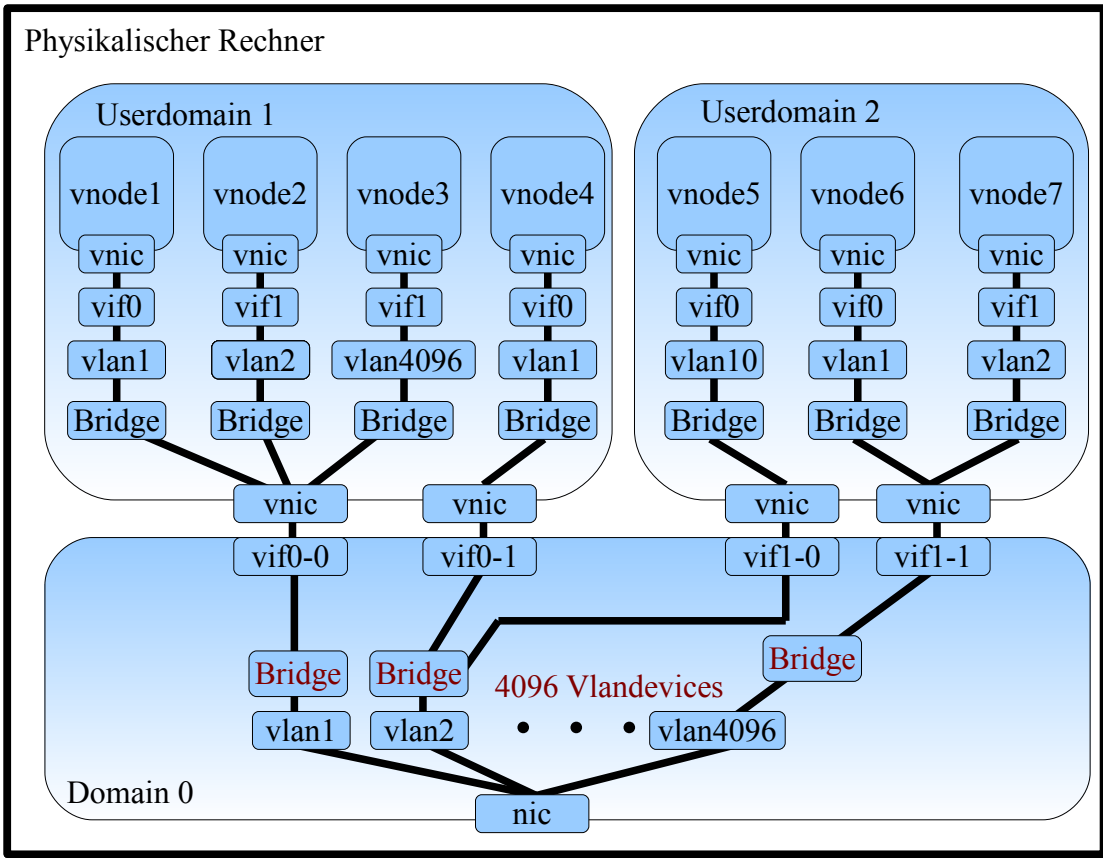
5.2 Design

Das Szenario des Clusters auf dem TVEE zum Einsatz kommt, besteht aus 16 Hardwareknoten mit mehreren virtuellen XEN Userdomänen. In jeder Userdomäne sind mehrere OpenVZ Container, die die virtuellen Knoten enthalten.
Die virtuellen Knoten werden nun zu Gruppen zusammengefasst. Der Idealfall ist, wenn allen Gruppen eine Vlan-ID zugewiesen werden kann. Bei mehr als den möglichen 4096 Gruppen, wird ein mehrstufiger Entwurf notwendig.

5.2.1 Zwei Ebenen Baumstruktur

Bei einem Mehrstufigem Entwurf werden zwischen den Hardwarerechnern 4096 Vlan-IDs vergeben. Diese werden mittels des Hardwareswitches verwendet um Nachrichten an die richtigen Hardwarerechner zu übertragen.
Sobald die Nachrichten an einem Hardwarerechner ankommen, werden sie mittels des Hypervisors von XEN von der physikalischen Netzwerkkarte an die virtuelle Netzwerkkarte der Domain0 übergeben. Nun erfolgt der erste Filtervorgang anhand der äußeren Vlan-ID. Dazu werden an die virtuelle Netzwerkkarte virtuelle Vlandevices angeschlossen. Jedes dieser virtuellen Vlandevices entspricht einer Vlangruppe zwischen den Hardwareknoten. Die Nachricht wird also nur über das entsprechende Vlandevice übertragen, dass die selbe Vlan-ID besitzt wie die äußere Vlan-ID im Ethernetheader der Nachricht. Nun wird noch die äußere Vlan-ID mit Hilfe des virtuellen Vlandevices entfernt, bevor die Nachricht weitergereicht wird.
Zur weiteren Verarbeitung der Nachricht, müssen in XEN an alle Vlandevices virtuelle Brücken angeschlossen werden. Diese virtuellen Brücken eröffnen die Möglichkeit die Nachricht wieder an mehrere mögliche Empfänger weiterzuverteilen.
An die virtuellen Brücken werden nun Netzwerkinterfaces angeschlossen, welche mit den XEN Userdomänen verbunden werden, in denen Rechner Mitglied sind, die zu der in der ersten Ebene entfernten Vlan-ID gehören.
Innerhalb der Userdomänen werden die virtuellen Knoten, also die virtuellen Container von OpenVZ, über virtuelle Netzwerkdevices in OpenVZ mit virtuellen Brücken innerhalb der XEN Userdomäne verbunden. An diese virtuellen Brücken sind virtuelle Vlandevices angebracht, die der inneren Vlan-ID, also der Gruppe entsprechen, in der der angeschlossene virtuelle Knoten Mitglied ist. Nun können die Vlandevices an eine virtuelle Netzwerkkarte innerhalb der XEN Userdomäne angeschlossen werden. Diese Netzwerkkarte wird mit einem virtuellen Interface innerhalb der XEN Domain0 verbunden, dass mit der Brücke

Abbildung 5.1: Zwei Ebenen Vlan-Baum



verknüpft ist, die der richtigen äußeren Vlan-ID entspricht(siehe Abbildung 5.1).
Auf diese Weise ist es möglich zwei Ebenen mittels 802.1 Q-in-Q umzusetzen.

Zu berücksichtigen ist, dass innerhalb der XEN Userdomänen nur eine begrenzte Anzahl an virtuellen Netzwerkkarten definiert werden kann. Da für alle virtuellen Rechner, die unterschiedliche äußere Vlan-IDs haben, eine virtuelle Netzwerkkarte in der jeweiligen XEN Userdomäne notwendig ist, muss ab einer gewissen Anzahl äußerer Vlan-IDs eine Migration und Zusammenfassung von virtuellen Knoten der gleichen äußeren Vlan-ID auf die gleiche XEN Userdomäne stattfinden.
Um drei Ebenen von Q-in-Q zu ermöglichen, müsste zwischen den virtuellen Netzwerkkarten von OpenVZ und der virtuellen Bridge in der XEN Userdomain eine weitere Ebene virtuelle Vlandevices eingefügt werden, die dann diese dritte Ebenen der Vlan-IDs interpretieren würde.

Des weiteren muss die Begrenzung der äußeren Vlan-IDs auf 4096 aufgrund des Hardwareswitches beachtet werden. Diese Begrenzung erfordert ein effizientes Auswählen der verwendeten äußeren Vlan-IDs.

5.2.2 Zwei Ebenen Flat

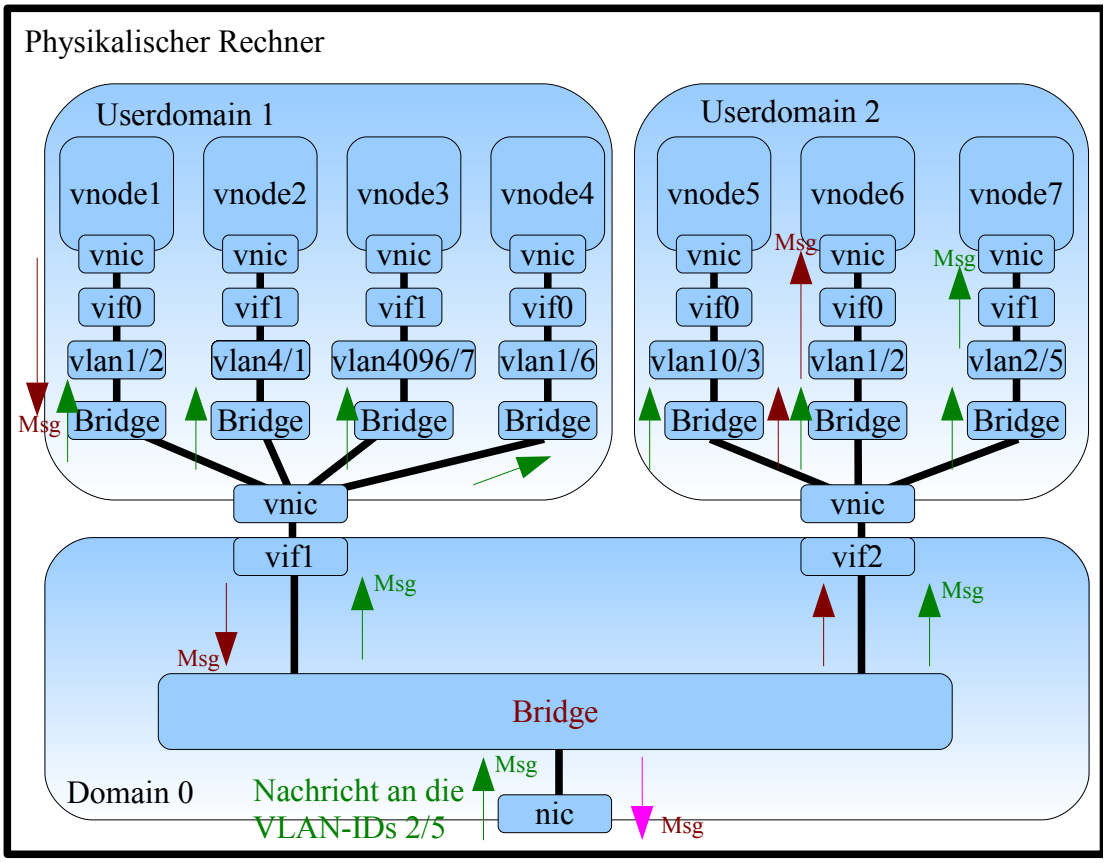
Für den Fall, dass mehr als 4096 Vlan-IDs zwischen den Hardwareknoten eingesetzt werden sollen, muss das Design leicht angepasst werden. Da nun mehr als 4096 Vlan-IDs verwendet werden sollen, der Hardwareswitch aber nur 4096 verarbeiten kann, muss mittels geeigneter Verfahren der Rest der Gruppen durch die 4096 Vlan-IDs ausgedrückt werden. Deshalb ist es nun nicht mehr möglich in der ersten Ebene nach der äußeren Vlan-ID im Ethernetheader vorzusortieren.
Stattdessen wird nicht nur ein zwei stufiges Q-in-Q verwendet, sondern ein drei-stufiges. Der Hardwareswitch muss die äußere Vlan-ID entfernen bevor er die Nachricht an die entsprechenden Hardwareknoten weitergibt. Sobald die Nachricht in den Hardwareknoten angekommen ist, wird sie von der virtuellen Netzwerkkarte in der XEN Domain0 an die erste virtuelle Brücke weitergereicht. Die virtuelle Brücke verteilt die Nachricht nun an alle XEN Userdomänen. Dazu wird sie mittels virtueller Interfaces mit virtuellen Netzwerkkarten in den XEN Userdomänen verbunden(siehe Abbildung 5.2).
Innerhalb der XEN Userdomänen befinden sich die virtuellen Rechner, die virtuellen Container von OpenVZ. Für jeden dieser virtuellen Rechner innerhalb der XEN Userdomäne wird ein virtuelles Vlandevice an die virtuelle Netzwerkkarte der XEN Userdomäne angeschlossen. Diese virtuellen Vlandevices verbindet jeweils eine virtuelle Brücke mit den virtuellen Netzwerkkarten der OpenVZ Container.

Die virtuellen Vlandevices müssen nun die zweite und dritte Vlan-ID verarbeiten. Die erste wurde ja schon vom Hardwareswitch entfernt. Das bedeutet, die virtuellen Vlandevices müssen so überarbeitet werden, dass sie in der Lage sind, zwei Vlan-IDs aus einem Ethernetheader auszulesen. Anhand des Protocol Identifier Feldes kann das Vlandevice erkennen, ob es sich um die letzte Vlan-ID handelt oder ob noch ein weiterer vorhanden ist. Dies ist hilfreich, falls je nach anfallenden Gruppenzahlen die Anzahl der Vlan-IDs vergrößert werden soll.
Die Protocol Identifier für die letzte Vlan-ID ist 8100, die vorhergehenden können 9100 oder 9200 verwenden.

Die Verarbeitung von Nachrichten im Header laufen innerhalb des Kernels so, dass ein Hardwaretreiber díe Nachricht empfängt und in einer „sk_buff structure“ [1] ablegt

[1] definiert in include/linux/sk_buff.h

Abbildung 5.2: Zwei Ebenen Flat Design



und „skb->dev“ auf das eigene Interface setzt. Dann wird die Nachricht an „netif_rx“ [2] weitergegeben und dann von „netif_receive_sbk“ [3] weiterverarbeitet.
Bei Nachrichten mit Vlanfeld wird die Nachricht nicht direkt an „netif_rx“ übergeben, sondern es wird zuerst an „vlan_hwaccel_rx“ [4] weitergereicht. Hier wird „skb->dev“ mit dem korrekten Vlansubinterface der Netzwerkkarte überschrieben. Danach wird die Nachricht an „netif_rx“ übergeben.
Hier kann angesetzt werden um mehr als eine Vlan-ID auszulesen.

Die Vlandevices können nun, da sie zwei Vlan-IDs ausgelesen haben, diese beiden Vlan-IDs zusammensetzen und somit Vlan-IDs mit 24Bit Länge statt mit nur 12Bit Länge verarbeiten. Das ermöglicht es 4096*4096=16.777.216 unterschiedliche Vlan-IDs zu verwalten.
Dies stellt einen Unterschied zum vorhergehenden Ansatz dar, bei dem 4096 Gruppen mit jeweils 4096 Untergruppen erstellt worden sind. Beim letzteren Ansatz sind die 16.777.216 Vlan-IDs über alle Knoten gleich gültig. Die äußere Vlan-ID wird nur noch dafür benötigt, eine Nachricht mit Hilfe des Hardwareswitches auf den richtigen Hardwareknoten zu übertragen.

5.2.3 Sendevorgang

Falls beim „zwei Ebenen Flat“ - Ansatz eine Gruppe zwischen den Hardwarerechnern nicht vollständig ausgedrückt werden kann, muss eine Nachricht an diese Gruppe durch mehrfaches Senden an andere Gruppen die eine Vlan-ID besitzen oder auch durch direktes Senden wenn allen Hardwarerechnern eine Vlan-ID zugeteilt worden ist, übertragen werden.
Um diese Funktionalität zu erhalten müssen an der virtuellen Brücke in der XEN Domäne Dom0 einige Modifikationen vorgenommen werden. Die virtuelle Brücke muss die Vlan-IDs von Nachrichten, die die virtuellen Knoten der XEN Userdomänen versenden, lesen können. Nachdem sie die Vlan-IDs aus dem Ethernetheader extrahiert hat, muss sie diese zu einer Vlan-ID zusammensetzen. Anhand der zusammengesetzten Vlan-ID muss die virtuelle Brücke nun entscheiden an welche Hardwareknoten diese Nachricht zu senden ist. Dazu muss in der virtuellen Brücke eine Tabelle vorgehalten werden.
Außerdem muss die virtuelle Brücke anhand der Tabelle entscheiden, ob es möglich ist die erforderlichen Hardwareknoten mittels einer Vlan-ID zu erreichen, oder ob mehrere Nachrichten notwendig werden um die Nachricht an alle Hardwareknoten zu übertragen. Sollte es mittels einer Vlan-ID möglich sein die Nachricht zu übertragen muss die virtuelle Brücke nur die korrekte äußere Vlan-ID dem Ethernetheader der Nachricht hinzufügen und die Nachricht weiterzureichen. Sollten aber mehrere Nachrichten notwendig werden,

[2] definiert in net/core/dev.c
[3] definiert in net/core/dev.c
[4] definiert in include/linux/if_vlan.h

muss die virtuelle Brücke die Nachricht mehrfach versenden, jedes mal mit einer anderen notwendigen Vlan-ID, solange bis alle Hardwareknoten erreicht werden konnten.

5.3 Zusammenfassung

Sollten 4096 oder weniger Vlan-IDs benötigt werden, so reicht der normale Vlanansatz aus.
Für größere Experimente, bei denen mehr als 4096 Vlan-IDs notwendig sind, kann man noch unterscheiden ob der Ansatz mit zwei Vlan-IDs ausreichend ist. Dieser ermöglicht 4096 Vlan-IDs zwischen den Hardwareknoten und dann für jede dieser Vlan-IDs noch 4096 Unterteilungen mittels den zweiten 4096 Vlan-IDs. Bei vollständiger Ausnutzung aller Unterteilungen ergibt sich hier eine Anzahl von 4096*4096=16.777.216 mögliche Vlan-IDs. Dann ergibt sich aber auch das Problem, dass unter Umständen virtuelle Knoten auf einem Hardwareknoten innerhalb der XEN Userdomänen migriert werden müssen, da die Anzahl an virtuellen Netzwerkkarten innerhalb einer XEN Userdomäne begrenzt ist. Die virtuellen Knoten in der XEN Userdomäne, die zu einer anderen Gruppe der ersten Ebene, der Hardwareknoten, gehört, benötigt eine eigene virtuelle Netzwerkkarte. Somit wird das Migrieren der virtuellen Knoten innerhalb der XEN Userdomänen ab einer gewissen Zahl an Vlan-IDs auf Hardwareknotenebene notwendig. Dies kann aber zu Problemen mit der Lastverteilung im Cluster führen, da je eine XEN Userdomäne pro CPU verteilt wird und bei Migration von virtuellen Rechnern die Last innerhalb des Clusters nicht mehr optimal verteilt ist.
Hier bietet sich die Erweiterung an, die beliebig viele Gruppen aus Hardwareknoten berücksichtigen kann und innerhalb der Hardwareknoten keine Unterteilung von Vlan-IDs verwendet, sondern eine große 24Bit Vlan-ID über alle Hardwareknoten verteilt. Dies hebt das Problem der Vlan-ID bedingten Migration auf.

Kapitel 6

Erzeugen der Vlangruppen

Da zwischen den 16 Hardwareknoten insgesamt 65535 mögliche Kombinationen von Hardwareknoten existieren aber nur 4096 Vlan-IDs auf Ebene des Hardwareswitches verarbeitet werden können, muss ein Weg gefunden werden diese 4096 Vlan-IDs möglichst effizient auszuwählen. Hierfür gibt es unterschiedliche Ansätze. Als Grundlage muss hier der Ansatz „zwei Ebenen flat“ verwendet werden, bei dem mindestens drei Vlan-IDs verwendet und die Vlandevices alle außer der ersten, äußeren ID, zu einer grossen ID zusammenfassen. Außerdem wird der Einfachheit halber nicht berücksichtigt, dass durch die Vlandefinition die Vlan-IDs 0,1 und 4095 nicht verwendet werden dürfen. Es wird im Folgenden von 4096 möglichen Vlan-IDs ausgegangen.

6.1 Mögliche Ansätze

Mögliche Ansätze sind, die Gruppen aus Hardwareknoten entweder in Ihrer Zahl zu reduzieren, sie durch andere Gruppen auszudrücken oder alle Nachrichten direkt an die Teilnehmer der Gruppen auszuliefern.
Um Nachrichten direkt an die einzelnen Teilnehmer einer Gruppe, die Hardwareknoten, auszuliefern ist es notwendig pro Hardwareknoten eine Vlan-ID zu vergeben. Dadurch kann eine Nachricht direkt an diesen Hardwareknoten gesendete werden.
Alternativ könnte die Nachricht auch einfach an alle Hardwareknoten verschickt werden, also ein Broadcast durchgeführt werden. Dazu ist es notwendig eine Vlan-ID zu vergeben, in der alle Hardwareknoten Mitglied sind.
Um die Anzahl der Gruppen zu reduzieren, muss man Gruppen zusammenfassen. Dadurch werden die Gruppen immer umfangreicher. Sollten viel mehr Gruppen vorhanden sein als Vlan-IDs verfügbar sind, kann dies dazu führen, dass alle zusammengefassten Gruppen alle Hardwareknoten enthalten und somit mit der Broadcastgruppe identisch sind.
Bleibt der Ansatz, Gruppen durch andere Gruppen auszudrücken. Falls eine Gruppe z.Bsp. die Hardwareknoten zwei,drei und vier als Mitglieder hat und eine andere Gruppe

nur Hardwareknoten zwei und drei, dann kann die letztere Gruppe gut durch die erste ersetzt werden. Da alle Nachrichten, die an die erstere Gruppe gesendet werden auch auf den Hardwareknoten ankommen, die in der letzteren Gruppe enthalten sind. Es werden nur zusätzlich Nachrichten an Hardwareknoten vier gesendet, der diese nicht erhalten soll. Diese Nachrichten enthalten noch die zweite und dritte Vlan-ID. Falls nun auf Hardwareknoten vier keine virtuellen Knoten sind, die durch die zweite Gruppe aus Hardwareknoten erreicht werden sollten, ist auch kein Vlandevice vorhanden, welches die Kombination aus zweiter und dritter Vlan-ID akzeptieren wird und die Nachricht wird verworfen. Somit muss für diesen Ansatz ein Verfahren gefunden werden, welche Gruppen aus Hardwareknoten am Besten als Vlan-IDs geeignet sind.
Relevante Daten sind die Anzahl der Hardwareknoten, die gleich der maximalen Größe der Gruppen aus Hardwareknoten ist. Dann die Anzahl der vergebbaren Vlan-IDs. Außerdem die Gesamtanzahl der Broadcastdomänen.
Auch erscheint es sinnvoll, je eine Vlan-ID pro Hardwareknoten zu vergeben, für den Fall, dass nicht allen benötigten Gruppen aus Hardwareknoten eine Vlan-ID zugeordnet werden kann, könnte man durch mehrfaches Senden zumindest alle Hardwareknoten erreichen.

6.2 Mehrfachsenden

Gruppen aus Hardwarerechnern, die nicht vollständig durch die gewählten Vlangruppen ausgedrückt werden können, benötigen eine Direktsenden Möglichkeit. Dazu muss an der virtuellen Brücke in der XEN Domain0 eine Modifikation vorgenommen werden. Die virtuelle Brücke muss die Vlan-IDs, einer von einem virtuellen Knoten versandten Nachricht, auslesen. Dann muss die virtuelle Brücke die Vlan-IDs zu einer Vlan-ID zusammensetzen und mit einer intern in der Brücke vorgehaltenen Tabelle vergleichen um zu ermitteln, an welche Hardwareknoten die Nachricht gehen soll und mittels welcher Vlan-ID dies geschehen kann.
Gibt es eine Vlan-ID, die verwendet werden kann, muss die virtuelle Brücke diese Vlan-ID als äußere Vlan-ID in den Ehternetheader der Nachricht einfügen und die Nachricht absenden. Gibt es keine Vlan-ID, mit der die Nachricht an die korrekten Hardwarerechner abgesendet werden kann, dafür aber eine Kombination aus zwei oder mehr Vlan-IDs, dann muss die virtuelle Brücke die Nachricht mehrfach versenden. Die Kombination aus zwei oder mehr Vlan-IDs muss gewährleisten, dass die Nachricht an alle notwendigen Hardwareknoten gesendet wird. Die virtuelle Brücke muss die Nachricht nun mehrfach senden und bei jedem Sendevorgang der Nachricht eine andere Vlan-ID in den Ethernetheader als äußere Vlan-ID einfügen, solange bis die Nachricht mit allen notwendigen Vlan-IDs versendet worden ist. Dadurch erreicht die Nachricht alle notwendigen Hardwareknoten. Alle Hardwareknoten, die aufgrund des mehrfachen Sendens die Nachricht unnötiger Weise erhalten haben, müssen diese verwerfen.
Gibt es auch keine Kombination aus Vlan-IDs die verwendet werden kann um die notwen-

digen Hardwareknoten zu erreichen, kann die Nachricht entweder direkt an die einzelnen Hardwareknoten gesendet werden oder aber als Broadcast an alle Hardwareknoten auf einmal.

6.3 Direct

Bei diesem Ansatz werden alle Nachrichten von der virtuellen Brücke einzeln an alle notwendigen Hardwareknoten verschickt. Damit dies möglich ist, wird jedem Hardwareknoten eine Vlan-ID zugeteilt(siehe Abbildung 6.1). Der Cluster des Projektes NET besitzt 16 Hardwareknoten deshalb werden hier 16 Vlan-IDs benötigt.
Wenn z.Bsp. die 16 Hardwareknoten die Vlan-IDs eins bis 16 inne hätten und eine Nachricht versendet wird, die an die Hardwareknoten zwei und vier gehen soll, dann verschickt die virtuelle Brücke die Nachricht einmal mit der äußeren Vlan-ID zwei und einmal mit der äußeren Vlan-ID vier.

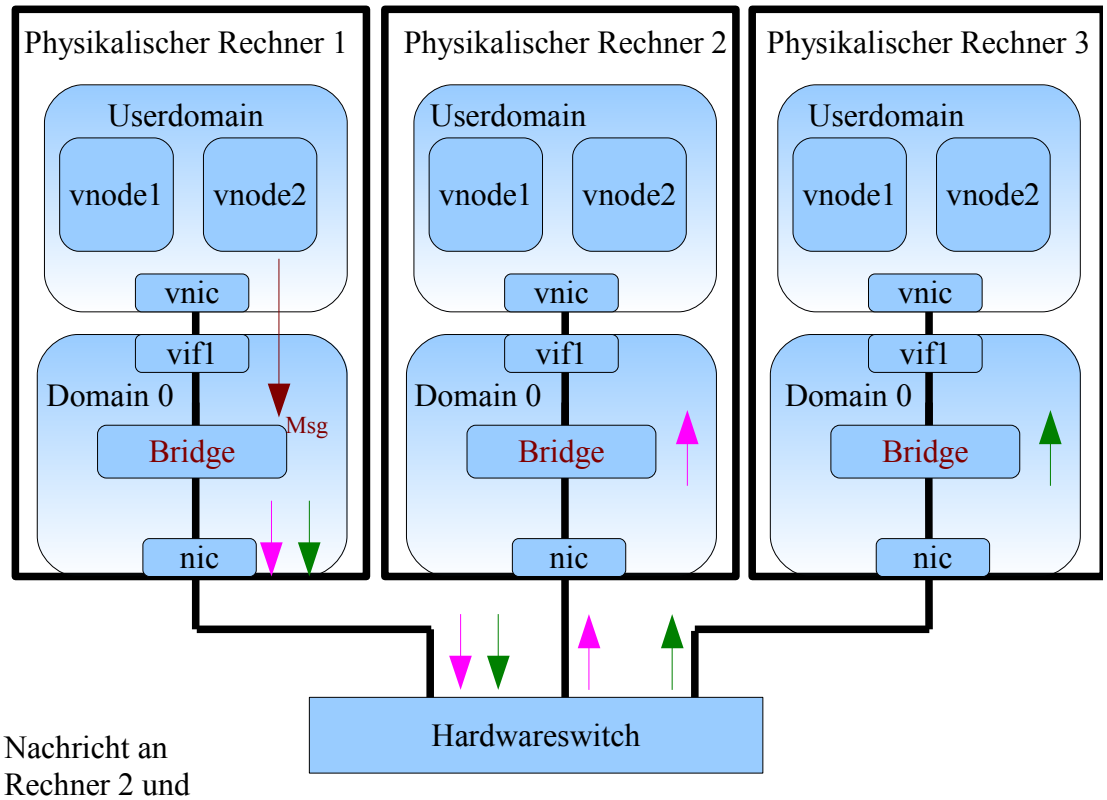
6.4 Broadcast

Bei diesem Ansatz wird eine zu sendende Nachricht mittels eines Broadcastes an alle Hardwareknoten gleichzeitig versendet. Dazu ist eine Vlangruppe notwendig, in der alle Hardwareknoten Mitglied sind. Dadurch kann häufiger der Fall entstehen, dass Hardwareknoten Nachrichten erhalten, die nicht für sie bestimmt sind(siehe Abbildung 6.2). Diese Nachrichten müssen von den jeweiligen Hardwareknoten verworfen werden. Wenn z.Bsp. eine Nachricht an Hardwareknoten zwei und vier gehen soll, müssen die Hardwareknoten 1,3,5 bis 16 die Nachricht verwerfen.

6.5 Gruppen ersetzen

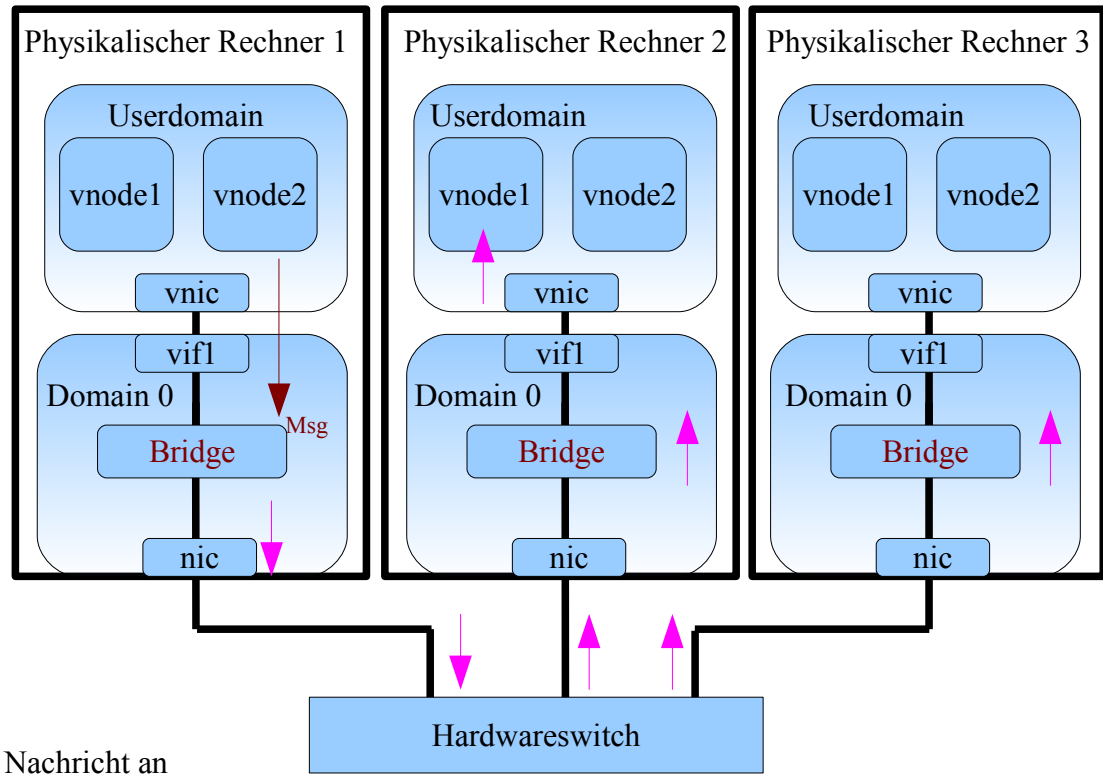
Bei diesem Absatz wird versucht eine Gruppe aus Hardwareknoten, der keine Vlan-ID zugewiesen wurde, durch eine Kombination aus Vlangruppen auszudrücken. Wenn z.Bsp. eine Nachricht an Hardwareknoten zwei, vier und fünf gehen soll, diese Kombination aber in keiner Vlangruppe definiert ist, wird versucht mögliche Ersatzkombinationen zu finden. Falls z.Bsp. eine Vlangruppe definiert ist mit den Hardwareknoten zwei und drei und eine Vlangruppe mit den Hardwareknoten vier, fünf und sechs, dann kann die Nachricht an diese beiden Vlangruppen gesendet werden und würde an den Hardwareknoten zwei,vier und fünf ankommen. Die Hardwareknoten drei und sechs müssten die Nachricht verwerfen. Alle Hardwaregruppen, für die keine geeignete Kombination gefunden werden kann, müssen direkt gesendet werden.

Abbildung 6.1: Verfahren Direct



Nachricht an
Rechner 2 und
Rechner 3 sendet Brücke als 2 Nachrichten

Abbildung 6.2: Verfahren Broadcast



Nachricht an
Rechner 2 sendet
Brücke an alle Rechner

Nun gibt es unterschiedliche Herangehensweisen, die am besten geeigneten Vlangruppen zu ermitteln. Bei diesem Ansatz wird nach der bestmöglichen Kombination aus Gruppen gesucht. Als bestmöglich wird die Gruppenkombination angenommen, die auf dem Cluster die geringste Last durch zusätzliches Senden, Empfangen oder Verwerfen von Nachrichten, verursacht. Die Kosten für zusätzliches Senden, Empfangen und Verwerfen wird für alle Nachrichten als gleich teuer angenommen.
Die verursachte Last auf dem Cluster ist immer abhängig von dem am stärksten belasteten Hardwareknoten da der restliche Cluster sich dem anpasst. Es wird also das minimale Maximum für den durch zusätzliches Senden, Empfangen oder Verwerfen am stärksten belasteten Knoten im Cluster gesucht.

6.5.1 All

Bei dem „All“ Ansatz wird versucht, durch berücksichtigen sämtlicher möglicher Kombinationen die Vlangruppen zu ermitteln, die am besten geeignet sind, alle Gruppen aus Hardwarerechnern auszudrücken. Das bedeutet, die Vlangruppen, die ermittelt werden, müssen nicht zwangsläufig auch einer Gruppe von Hardwarerechnern entsprechen. Es müssen nur die besten Kombinationen von Hardwarerechnern erstellt werden, so dass insgesamt das minimale Maximum des am stärksten belasteten Hardwarerechners erreicht wird.
Dazu müssen zuerst alle möglichen Gruppen aus Hardwarerechnern gebildet werden, mit allen möglichen Längen und Kombinationen aus der gegebenen Anzahl von 16 Hardwareknoten. z.Bsp. 1 - 2 -...- 1,2 - 1,3 -...- 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16.
Die Anzahl der möglichen Kombinationen entspricht der Summe der Binomialkoeffizienten:

$$\sum_{n=1}^{16} \binom{16}{n} = \sum_{n=1}^{16} \frac{16!}{n!(16-n)!} = 65535$$

Danach werden aus diesen Gruppen so viele Gruppen ausgewählt wie Vlan-IDs zur Verfügung stehen, also 4096. Das geschieht nicht nur einmal sondern es werden alle möglichen Kombinationen aus diesen Gruppen gebildet wobei die Kombinationen immer aus 4096 Gruppen bestehen. Die Anzahl an Kombinationen aus 4096 dieser Gruppen:

$$\binom{65535}{4096} = \frac{65535!}{4096!(65535-4096)!}$$

Jede einzelne dieser Kombinationen muss nun mit jeder Hardwarerechnergruppe auf Gemeinsamkeiten verglichen werden. Dies geschieht, indem jede Gruppe aus der Kombination mit den Hardwarerechnergruppen auf gemeinsame Elemente geprüft wird. Ist dies geschehen, kann man die Kombination auswählen, die über alle Hardwareknoten

hinweg gesehen das minimale Maximum auf einem Hardwarerechner erzeugt, wenn man die Hardwareknoten durch diese Kombination ausdrückt. Diese Kombination wird dann als Vlangruppenkombination übernommen.
Anzahl der notwendigen Gruppenvergleiche:

$$\binom{65535}{4096} * 4096 * \textit{Anzahl Hardwarerechnergruppen}$$

6.5.2 Groups

Bei dem Ansatz „Groups“ wird versucht, die Menge der zu überprüfenden Kombinationen aus möglichen Gruppen, die mit den Hardwarerechnergruppen verglichen werden müssen, zu reduzieren. Hier werden die Vlangruppen nur aus den Hardwarerechnergruppen ausgewählt. Es werden keine neuen gebildet. Das bedeutet es werden alle möglichen Kombinationen aus 4096 Hardwarerechnergruppen gebildet und dann wird jeweils versucht, alle Hardwarerechnergruppen durch diese Gruppenkombination auszudrücken. Die Anzahl der möglichen Kombinationen ist abhängig von der Anzahl der Hardwarerechnergruppen:

$$\binom{\textit{Anzahl Hardwarerechnergruppen}}{4096}$$

Die Anzahl der notwendigen Gruppenvergleiche ist:

$$\binom{\textit{Anzahl Hardwarerechnergruppen}}{4096} * 4096 * \textit{Anzahl Hardwarerechnergruppen}$$

6.5.3 Large Groups

Bei diesem Ansatz wird erneut versucht, die Anzahl der zu überprüfenden Kombinationen weiter zu reduzieren. Hier werden die Hardwarerechnergruppen nach ihrer Größe sortiert, also nach der Zahl der Mitglieder die sie haben. Dann werden nur die 4096 größten Hardwarerechnergruppen zu Vlangruppen gemacht. Der Vorteil hier ist, dass große Hardwarerechnergruppen falls sie nicht als Vlangruppe übernommen werden durch mehrere kleine Vlangruppen ausgerückt werden müssen. Dies bedeutet mehrere Sendevorgänge. Werden aber die großen Hardwarerechnergruppen übernommen ist die Wahrscheinlichkeit hoch, dass eine kleine Hardwarerechnergruppe, die keine Vlangruppe ist, in einer der großen Vlangruppen enthalten ist und durch nur einen Sendevorgang an diese Gruppe

ersetzt werden kann.
Die Anzahl der möglichen Kombinationen der Gruppen ist eins da nur die großen Gruppen verwendet werden. Die Anzahl der notwendigen Gruppenvergleiche beträgt:

$$4096 * \textit{Anzahl Hardwarerechnergruppen}$$

6.5.4 Zusammenfassen von Vlangruppen

Bei diesem Ansatz wird versucht, die Anzahl der Hardwarerechnergruppen auf die Anzahl der Vlan-IDs zu verringern. Dabei werden die Hardwarerechnergruppen der Größe nach sortiert, das heißt nach der Anzahl ihrer Mitglieder. Dann werden immer die beiden kleinsten Hardwarerechnergruppen zu einer neuen Gruppe zusammengefasst. Damit wird solange fortgefahren bis nur noch so viele Gruppen wie Vlan-IDs vorhanden sind. Das funktioniert am Besten wenn die Anzahl der Hardwarerechnergruppen nicht viel größer als die Anzahl der Vlan-IDs ist da durch wiederholtes Zusammenfassen von schon einmal zusammengefassten Gruppen diese Gruppen irgendwann alle Hardwarerechner enthalten und somit zum Broadcastverfahren werden. Die Anzahl der möglichen Kombinationen ist eins, da nur 4096 Gruppen übrig bleiben. Die Anzahl der notwendigen Gruppenvergleiche beträgt:

$$4096 * \textit{Anzahl Hardwarerechnergruppen}$$

6.6 Kombinationen

Einige der aufgeführten Verfahren lassen sich auch kombinieren. Dadurch werden die Resultate besser oder die Laufzeit des Auswahlverfahrens verkürzt sich.

6.6.1 Direct - Broadcast

Die Verfahren Direct und Broadcast lassen sich kombinieren. Für jede Hardwarerechnergruppe wird ermittelt, welche Last Direct und Broadcast erzeugen würden. Am Schluss wird dann über alle Hardwarerechnergruppen die Beste Entscheidung für jede Hardwarerechnergruppe getroffen. Anzahl der notwendigen Gruppenvergleiche beträgt:

$$2 * \textit{Anzahl Hardwarerechnergruppen}$$

6.6.2 Large Groups - Direct

Bei der Kombination von Large Groups und Direct werden die Hardwarerechnergruppen nach der Anzahl der Mitglieder sortiert und die 4096 größten Hardwarerechnergruppen

werden als Vlangruppen übernommen. Nun wird nicht versucht die restlichen Hardwarerechnergruppen mittles der gewählten Vlangruppen auszudrücken sondern alle restlichen Hardwarerechnergruppen werden direkt an die 16 Hardwareknoten verschickt. Hierfür werden die 16 Vlan-IDs die den einzelnen Hardwarerechnern zugeteilt worden sind verwendet. Die Kombination dieser Verfahren spart Zeit bei der Suche nach den geeigneten Vlangruppen da die übriggebliebenen Hardwarerechnergruppen nicht durch die Vlangruppen ausgedrückt werden müssen. Anzahl der notwendigen Gruppenvergleiche beträgt:

$$1 * \textit{Anzahl Hardwarerechnergruppen}$$

6.6.3 Large Groups - Direct - Broadcast

Bei der Kombination von Large Groups, Direct und Broadcast werden ebenfalls die Hardwarerechnergruppen nach der Anzahl der Mitglieder sortiert und die 4096 größten Hardwarerechnergruppen werden als Vlangruppen übernommen. Für die restlichen wird nun jeweils Direct und Broadcast berechnet. Am Schluss wird über alle Hardwarerechnergruppen ermittelt für welche Hardwarerechnergruppe Direct und für welche Broadcast angewendet werden soll um das minimale Maximum für den am stärksten belasteten Rechner zu erreichen. Anzahl der notwendigen Gruppenvergleiche beträgt:

$$2 * \textit{Anzahl Hardwarerechnergruppen}$$

6.6.4 Middle Groups - Direct - Broadcast

Bei der Kombination von Middle Groups, Direct und Broadcast werden aus den nach der Anzahl der Mitglieder sortierten Hardwarerechnergruppen nicht die größten Hardwarerechnergruppen als Vlangruppen ausgewählt sondern die mittleren. Für die restlichen Hardwarerechnergruppen werden die Kosten für Direct und Broadcast ermittelt.Am Schluss wird berechnet, für welche Hardwarerechnergruppe Direct und für welche Broadcast verwendet werden muss um das minimale Maximum für den am stärksten belasteten Hardwareknoten zu ermitteln. Anzahl der notwendigen Gruppenvergleiche beträgt:

$$2 * \textit{Anzahl Hardwarerechnergruppen}$$

6.7 Komplexität

Jeder unterschiedliche Ansatz, die optimalen Vlangruppen aus den Hardwarerechnergruppen zu ermitteln, hat unterschiedliche Vor- und Nachteile. Einige Ansätze benötigen eine enorme Anzahl an Vergleichen um die Gruppen zu ermitteln anderen liefern sehr schnell ein Resultat.
Wichtig ist auch, wie stark das Resultat, also die ermittelten Vlangruppen, den Cluster

belasten durch zusätzliches Senden, Empfangen oder Verwerfen von Nachrichten.
Im folgenden wird das Szenario angenommen, dass jeder Hardwarerechner eine Nachricht versendet an alle seine Gruppenmitglieder. Anzahl der Gruppenmitglieder wird mit AGM, Senden mit tx und Empfangen mit rx abgekürzt. Die Kosten für Senden und Empfangen werden als gleich groß angenommen. Hier werden nur die Komplexitäten für die Verfahren Direct und Broadcast betrachtet.

6.7.1 Direct

Bei Direct muss jeder Hardwarerechner eine Nachricht pro Mitglied der Gruppe versenden um alle zu erreichen. Da das alle Mitglieder der Hardwarerechnergruppe so machen, wird dieser Hardwarerechner auch von allen andere Mitgliedern der Gruppe eine Nachricht empfangen.
Aufwand des einzelnen Rechners:

$$(AGM - 1) * (1tx + 1rx)$$

6.7.2 Broadcast

Bei Broadcast muss jeder Hardwarerechner nur eine Nachricht an seine Gruppe senden. Alle Hardwarerechner werden diese Nachricht erhalten. Alle anderen Mitglieder verschicken ihre Nachricht auf die selbe Art und Weise, also wird der Hardwareknoten auch von allen anderen Mitgliedern eine Nachricht empfangen. Bei allen Übermittlungen werden die Nachrichten ausserdem von allen Hardwarerechnern empfangen, die nicht Mitglied der Hardwarerechnergruppe sind. Um den Aufwand des einzelnen Rechners für dieses Verfahren ermitteln zu können, wird für die Abschätzung angenommen, dass alle Hardwarerechnergruppen die selbe Größe haben. Die Anzahl der Hardwarerechner wird mit AHR abgekürzt. Aufwand des einzelnen Rechners für eine Hardwarerechnergruppengröße von zwei:

$$(1tx + 1rx) * 4096 + \frac{4096}{\binom{AHR}{2}} * (AHR - 1) * (AHR - 2) * 2$$

Für eine Hardwarerechnergruppegröße von 16 ergibt sich ein Aufwand pro Rechner von:

$$(1tx + 15rx) * 4096 + 0$$

Daraus kann man ablesen, dass Broadcast für viele kleine Gruppen schlechtere Werte liefert als für große Gruppen.

Kapitel 7

Implementierung der Vlangruppenerstellung

In diesem Kapitel wird die Umsetzung der einzelnen Verfahren zur Erzeugung der Vlangruppen näher beschrieben. Die einzelnen Verfahren können in unterschiedliche Bereiche gegliedert werden. Einige dieser Bereiche kommen in mehreren Verfahren zum Einsatz und werden deshalb getrennt von den Kernbereichen der einzelnen Verfahren beschrieben. Für alle Verfahren wird angenommen, dass Senden, Empfangen und Verwerfen einer Nachricht den selben Aufwand für den Hardwareknoten bedeutet. Nun werden in allen Verfahren die Kosten berechnet, die für die einzelnen Hardwarerechner entstehen, falls jeder Hardwarerechner, in den Gruppen, in denen er Mitglied ist, jeweils eine Nachricht an alle Mitglieder sendet und folglich auch von allen Mitgliedern eine Nachricht erhält. In einigen Verfahren müssen die Hardwareknoten zusätzlich noch Nachrichten verwerfen. Diese Berechnung der Kosten wird in den unterschiedlichen Verfahren benötigt, um die beste Verteilung der Vlangruppen zu errechnen.
Es werden also unterschiedliche Verfahren implementiert, die für bestimmte Rahmenbedingungen möglichst gute Vlangruppenverteilungen zwischen den Hardwarerechnern berechnen.
Im Folgenden werden in Quellcodeausschnitten die Abkürzungen *HRG* für die Hardwarerechnergruppen und *AGM* für Anzahl der Gruppenmitglieder verwendet.

7.1 Szenario

Den einzelnen Verfahren wird ein Szenario mittels einer Configdatei übergeben. In dieser Configdatei werden die einzelnen Hardwarerechnergruppen definiert.

7.2 Direct

Beim Verfahren Direct wird die Configdatei eingelesen und die Hardwarerechnergruppen in Tabellen abgelegt. Bei diesem Verfahren werden alle Hardwarerechnergruppen mittels einer eigenen Vlan-ID direkt angesprochen. Deshalb ist auch keine Vorauswahl bei den Vlangruppen notwendig.
Für die Evaluierung werden für alle Hardwarerechnergruppen direkt die Kosten berechnet. Hierfür wird für die vorhandenen Hardwarerechner, pro Hardwarerechnergruppe eine Tabelle *Kosten* geführt. In dieser Tabelle werden die Kosten, die für die einzelnen Rechner anfallen, gespeichert. Nun werden für jedes Mitglied der Hardwarerechnergruppe die anfallenden Kosten summiert und in der Tabelle abgelegt, siehe Listing 7.1.

Listing 7.1: Kostenberechnung Direct

```
1 HRG = array( array() )
2 for (s=0;s < count(HRG);s++){ //Jede Hardwarerechnergruppe durchlaufen
3   AGM = count(HRG[s])
4   for (i=0;i < AGM;i++){  //Jedes Mitglied der Gruppe durchlaufen
5     Kosten[ HRG [s][i] ] += send_kosten_konst * (AGM - 1)
6     Kosten[ HRG [s][i] ] += receive_kosten_konst * (AGM - 1)
7   }
8 }
```

Nachdem die Kosten für alle Hardwarerechnergruppen berechnet worden sind, lässt sich durch Aufsummieren errechnen, wie stark der am stärksten belastete Hardwarerechner belastet wird. Bei diesem Verfahren muss keine beste Lösung gefunden werden, da einfach alle Hardwarerechnergruppen direkt erreicht werden.

7.3 Broadcast

Beim Verfahren Broadcast werden ebenfalls die Configdatei eingelesen und die Hardwarerechnergruppen in Tabellen abgelegt. Ebenso wie bei Direct ist auch hier keine Vorauswahl der Vlangruppen notwendig. Aber im Gegensatz zu Direct werden hier nicht die einzelnen Vlan-IDs der Hardwareknoten verwendet sondern nur eine Vlan-ID, in der alle Hardwarerechner Mitglied sind. Nun werden alle Nachrichten immer an diese Vlan-ID versendet. Hardwarerechner, die eine Nachricht erhalten, die nicht für sie bestimmt ist, müssen diese Nachricht verwerfen. Die Anzahl der anfallenden zu verwerfenden Nachrichten hängt von der jeweiligen Größe der Hardwarerechnergruppe ab, da jedes Mitglied der Gruppe an alle Gruppenmitglieder sendet und jeder Sendevorgang alle Hardwarerechner erreicht.
Für die Evaluierung können hier ebenfalls direkt die Kosten berechnet werden und für jede Hardwarerechnergruppe in einer Tabelle *Kosten* gespeichert werden, siehe Listing 7.2.

Listing 7.2: Kostenberechnung Broadcast

```
1 HRG = array( array() )
2 for (s=0;s < count(HRG);s++){ //Jede Hardwarerechnergruppe durchlaufen
3   AGM = count(HRG[s])
4   Drop_Kosten = AGM * drop_kosten_konst
5   Diff = array_diff ( Vlangruppe, HRG[s] )
6   for (i=0;i < AGM;i++){  //Jedes Mitglied der Gruppe durchlaufen
7     Kosten[ HRG [s][i] ] += send_kosten_konst * 1
8     Kosten[ HRG [s][i] ] += receive_kosten_konst * (AGM - 1)
9         }
10  for (i=0;i < count(Diff); i++) { //Alle Hardwareknoten durchlaufen, die verwerfen
    müssen
11    Kosten[ Diff [i] ] += Drop_Kosten
12  }
13 }
```

7.4 Gruppen ersetzen

Die Verfahren, die versuchen Hardwarerechnergruppen durch andere Gruppen auszudrücken, müssen im Gegensatz zu Direct und Broadcast zuerst alle dafür in Frage kommenden Gruppen ermitteln. Hierzu lesen alle diese Verfahren zuerst die Configdatei ein und speichern die Hardwarerechnergruppen in Tabellen. Danach wird eine Menge von Gruppen aus Hardwarerechnern erstellt, die je nach Verfahren unterschiedlich ist, aus denen später die Vlangruppen ausgewählt werden, die Vlankandidatengruppen. Dann werden aus dieser Menge von Vlankandidatengruppen so viele Gruppen ausgewählt, wie später Vlan-IDs zur Verfügung stehen, also 4096. Dies wird so oft durchgeführt, bis alle möglichen Kombinationen aus 4096 dieser Vlankandidatengruppen ausgewählt worden sind. Mittels jeder dieser Kombinationen wird nun versucht, die einzelnen Hardwarerechnergruppen, die aus der Configdatei eingelesen worden sind, auszudrücken. Dabei werden alle in einer Kombination befindlichen Vlankandidatengruppen mit den einzelnen Hardwarerechnergruppen aus der Configdatei verglichen. Jede Kombination verursacht dabei unterschiedlich hohe Kosten durch die notwendige Anzahl an Senden, Empfangen und Verwerfen. Diese Kosten werden gespeichert. Nachdem sämtliche Möglichkeiten, die einzelnen Hardwarerechnergruppen aus der Configdatei durch die Kombination aus Vlankandidatengruppen auszudrücken geprüft worden sind, existiert für jede Hardwarerechnergruppe eine oder mehrere mögliche Lösungen. Ziel ist es, aus allen möglichen Kombinationen dieser Lösungen die herauszusuchen, mittels der die geringste Last auf dem Cluster verursacht wird. So wird für alle Kombinationen der Vlankandidatengruppen vorgegangen und nach Beendigung des Verfahrens kann anhand der minimalen Kosten jeder Kombination, die Kombination ausgewählt werden, die insgesamt die geringste Last verursacht. Diese Kombination aus Vlankandidatengruppen wird dann als Vlangruppen verwendet

7.4.1 Gemeinsame Teile aller Verfahren

Hier werden die Teile der einzelnen Verfahren erwähnt, die vom grundsätzlichen Aufbau gleich sind. So müssen z.Bsp. in allen Verfahren, die Gruppen ersetzen, Gruppenkombinationen aus Vlankandidatengruppen erstellt werden.

Gruppenkombinationen

In jedem der Verfahren, die Gruppen durch andere Gruppen auszudrücken versuchen, werden zuerst alle Vlankandidatengruppen erzeugt, die bei dem jeweiligen Verfahren als Vlangruppe in Frage kommen und in einer Tabelle gespeichert. Aufgrund der maximal verfügbaren Vlan-IDs von 4096 muss aus den möglichen Vlankandidatengruppen die am Besten geeignete Kombination von 4096 Gruppen ausgewählt werden. Dazu werden alle möglichen Kombinationen aus 4096 Vlankandidatengruppen geprüft und so die Beste Kombination ermittelt. Um nicht alle möglichen Kombinationen aus Vlankandidatengruppen speichern zu müssen, werden diese zur Laufzeit berechnet und dann an eine Prüffunktion übergeben. Falls z.Bsp. die maximale Anzahl an Vlan-IDs drei wäre und die Anzahl der Vlankandidatengruppen vier dann sähe die Kombination aus möglichen Vlankandidatengruppen so aus, falls bei null zu zählen begonnen wird, Vlankandidatengruppen mit Komma getrennt, Kombinationen aus Vlankandidatengruppen mit Gedankenstrich getrennt: 0,1,2 - 0,1,3 - 0,2,3 - 1,2,3. Die Zahlenkombination wird zur Laufzeit errechnet und dann als Index für die Tabelle verwendet, in der alle Vlankandidatengruppen gespeichert sind, siehe Listing 7.3.

Listing 7.3: Durchlaufen aller Gruppenkombinationen

```
1 id=Anzahl_IDs
2 steps=0
3 check=0
4 Gruppen_nr=array()
5 s = id - 1
6 while (s >= 0){  //Gruppenarray initialisieren
7   array_push( Gruppen_nr,s )
8   s--
9 }
10 s=0
11 Gruppen_nr[0]--
12 while ( Gruppen_nr[ id-1 ] < Gruppen_Anzahl - s ){
13   if ( Gruppen_nr[ 0 ]+1 < Gruppen_Anzahl ){ Gruppen_nr[ 0 ] = Gruppen_nr[ 0 ] + 1}
14   else {uebertrag=1}
15   s=1
16   while ( (s < count(Gruppen_nr)) AND (uebertrag == 1) \
17                                   AND (Gruppen_nr[id-1] < Gruppen_Anzahl - s ) ){
18     if (Gruppen_nr[s] + uebertrag + s == Gruppen_Anzahl){
19       uebertrag=1
```

```
20      }
21      else {
22        if ( Gruppen_nr[ id - 1 ] < Gruppen_Anzahl-s ){
23            Gruppen_nr[s]=Gruppen_nr[s]+uebertrag
24        }
25        uebertrag=0
26        doit=1
27      }
28      steps=s
29      s++
30    }
31
32    if (check==1){ // Falls Übertrag, Gruppenarray neu initialisieren
33      i=steps-1
34      while (i >= 0){
35        Gruppen_nr[i]=Gruppen_nr[i+1] + 1
36        i--
37      }
38      check=0
39    }
40    Teste (Gruppen [Gruppen_nr])
41  }
```

Gruppenvergleiche

In allen Verfahren, die Gruppen durch andere Gruppen auszudrücken versuchen, müssen einzelne Gruppen auf gemeinsame Mitglieder verglichen werden. Dazu werden die einzelnen Gruppen der in Listing 7.3 beschriebenen Gruppenkombinationen mit allen Hardwarerechnergruppen aus der Configdatei verglichen. Siehe Listing 7.4, *Gruppen* ist eine Tabelle die alle möglichen Vlankandidatengruppen enthält, *Gruppen_in_Kombination* ist eine Tabelle, mit den Nummern der ausgewählten Vlankandidatengruppen in der aktuellen Kombination.

Listing 7.4: Gruppenvergleich

```
1  for (HRG_nr=0; HRG_nr < count(HRG); HRG_nr++){
2    for (i=0; i < count(Gruppen_in_Kombination); i++){
3      s=0
4      found=array()
5      while ( (s < count(Gruppen[ Gruppen_in_Kombination [i] ] )&&(count(HRG[HRG_nr] >
6        if (in_array(Gruppen[ Gruppen_in_Kombination [i] ][s],HRG[hwg_nr])){
7          position=array_search(Gruppen[ Gruppen_in_Kombination [i] ][s],HRG[HRG_nr])
8          unset(HRG[HRG_nr][position])
9          array_push(found,Gruppen[ Gruppen_in_Kombination [i] ][s])
10       }
11       s++
```

```
12      }
13    }
14 }
```

Prüffunktion

Die Vlankandidatengruppen aus den jeweiligen Kombinationen müssen mit jeder einzelnen Hardwarerechnergruppe verglichen werden. Ziel ist es herauszufinden, wie die Hardwarerechnergruppe durch eine beliebige Kombination aus Vlankandidatengruppen ausgedrückt werden kann. Hierfür wird eine Prüffunktion definiert, die rekursiv prüft. Dieser Prüffunktion wird die aktuelle Kombination aus Vlankandidatengruppen und eine Hardwarerechnergruppe übergeben. Nun vergleicht die Prüffunktion die erste Vlankandidatengruppe mit der Hardwarerechnergruppe auf gemeinsame Mitglieder. Sollten keine gemeinsamen Mitglieder vorhanden sein wird mit der nächsten Vlankandidatengruppe fortgefahren. Sind jedoch gemeinsame Mitglieder vorhanden wie mittels Listing 7.4 herausgefunden wird, dann ruft die Prüffunktion sich selbst rekursiv auf und übergibt als Hardwarerechnergruppe die Mitglieder der Hardwarerechnergruppe, die nicht als gemeinsame Mitglieder gefunden wurden und fährt mit dem Vergleich bei der nächsten Vlankandidatengruppe fort. Konnten auch nach Durchlauf aller Vlankandidatengruppen nicht alle Mitglieder der Hardwarerechnergruppe gefunden werden, werden diese direkt an ihre zum Hardwarerechner gehörige Vlan-ID verschickt. Sobald alle Mitglieder der Hardwarerechnergruppe gefunden worden sind, kann diese durch die Kombination aus Vlankandidatengruppen, die rekursiv gefunden wurde, ausgedrückt werden und die Kombination wird gespeichert. Zusammen mit der Kombination werden auch Kosten gespeichert, die diese Kombination verursacht. Darauf wird später eingegangen. Danach kehrt die Rekursion wieder eine Ebene zurück und es wird hier mit der nächsten möglichen Vlankandidatengruppe und dem hier vorhandenen Rest der Hardwarerechnergruppe fortgefahren. Alle möglichen Lösungen werden immer gespeichert. Nachdem die Rekursion beendet ist wurde die Hardwarerechnergruppe mit allen möglichen Kombinationen aus Vlankandidatengruppen der aktuellen Kombination verglichen.

Dies wird für alle Hardwarechnergruppen und dieser Kombination aus Vlankandidatengruppen durchgeführt. Als Ergebnis ergeben sich Tabellen mit Lösungen für jede Hardwarerechnergruppe und dieser Kombination aus Vlankandidatengruppen.

Kostenberechnung

Die einzelnen Hardwarerechnergruppen durch die Gruppenkombinationen aus Listing 7.3 auszudrücken, kann mehrere unterschiedliche Lösungen ergeben. Für alle diese Lösungen werden Kosten berechnet, wie stark sie die einzelnen Hardwarerechner belasten. Dazu werden die anfallenden Sende-, Empfangsvorgänge und das Verwerfen von Nachrichten

herangezogen. Es wird angenommen, dass alle drei Fälle gleich teuer sind. Die Gruppenkombinationen enthalten 4096 Vlankandidatengruppen. Diese Kombinationen werden mittels der weiter oben beschriebenen Prüffunktion mit den Hardwarerechnergruppen verglichen. Nach jedem Vergleich innerhalb der rekursiven Prüffunktion werden für die gefundene (Teil-)Lösung die Kosten berechnet, die sie verursacht, siehe Listing 7.5.

Listing 7.5: Kostenberechnung

```
1 found=array(Gefundene_Mitglieder)
2 for (i=0;i < count(HRG);i++){  //Jede Hardwarerechnergruppe durchlaufen
3   for (s=0;count(HRG[i]);s++){  //Jedes Mitglied der Gruppe durchlaufen
4     Kosten[ HRG[i][s] ] += send_kosten_konst
5   }
6   Drop_Kosten = count( HRG[i] ) * drop_kosten_konst
7
8   if(count(found) == 1){  //Falls nur ein Mitglied gefunden wurde, sendet es nicht
  an sich selbst
9     Kosten[ found[0] ] -= send_kosten_konst
10    Drop_Kosten -= drop_kosten_konst
11  }
12
13  for (s=0;s < count(found); s++){
14        Kosten[ found[s] ] += receive_kosten_konst * (count(HRG[i]) - 1)
15  }
16
17  Diff = array_diff ( Gruppe_aus_Kombination, found )
18  for (s=0;s < count(Diff); s++) { //Alle Hardwareknoten durchlaufen, die verwerfen
  müssen
19    Kosten[ Diff [s] ] += Drop_Kosten
20  }
21 }
```

Pfadprüfung

Nach dem Vergleich einer Kombination aus Vlankandidatengruppen mit allen Hardwarerechnergruppen ergeben sich als Ergebnis Tabellen für jede Hardwarerechnergruppe mit möglichen Lösungen zum Ausdrücken der Hardwarerechnergruppe durch die Kombination aus Vlankandidatengruppen. Um die beste Lösung für den Cluster über alle Hardwarerechnergruppen hinweg zu finden reicht es nicht, die kostengünstigste Lösung für jede einzelne Hardwarerechnergruppe auszuwählen. Es ist möglich, dass eine nicht optimale Lösung aus Sicht einer Hardwarerechnergruppe bestimmte Hardwarerechner belastet, die durch die anderen Hardwarerechnergruppen nicht belastet werden und so über alle Hardwarerechnergruppen hinweg eine geringere Gesamtlast für den Cluster ermöglicht. Versucht wird, den durch die Kombination aus Vlankandidatengruppen am stärksten belasteten Hardwarerechner möglichst minimal zu belasten. Dazu werden alle möglichen Kombinationen der Lösungen aller Hardwarerechnergruppen erzeugt. Dann

wird errechnet, was das Verwenden dieser Lösungen jeweils als Maximallast auf einem Hardwarerechner hervorruft. Die Kombination aus Lösungen der Hardwaregruppen, die die minimale Maximallast hervorruft, wird als beste Lösung für diese Kombination aus Vlankandidatengruppen gewählt.

Verbesserungen

Um die Laufzeit zu verkürzen wird in der Prüffunktion ein Cache eingerichtet, damit Vergleiche zwischen Vlankandidatengruppen und Hardwarerechnergruppen nur einmal durchgeführt werden müssen. Sollte der Vergleich zwischen den selben Gruppen später noch einmal notwendig werden können die Ergebnisse direkt aus dem Cache geladen werden.

Ebenfalls wird in der Prüffunktion innerhalb der Rekursion „Treepruning“ durchgeführt. Hierfür wird nach Ermittlung einer Lösung die durch sie verursachte maximale Belastung für einen Hardwarerechner gespeichert. Bei allen folgenden Vergleichen zwischen einer Kombination aus Vlankandidatengruppen und den Hardwarerechnergruppen wird nach jeder Hardwarerechnergruppe mittels Pfadprüfung geprüft, ob die aktuelle Belastung für einen Hardwarerechner bereits höher ist als die bisher maximale Belastung durch eine fertige Lösung. Ist dies der Fall, muss die aktuelle Kombination aus Vlankandidatengruppen nicht weiter geprüft werden.

7.4.2 All

Bei dem Ansatz All wird die optimale Kombination aller möglichen Vlankandidatengruppen ermittelt. Aus diesem Grund ist es auch notwendig, sämtliche Möglichkeiten zu berücksichtigen. Zuerst werden die Hardwarerechnergruppen aus der Configdatei ausgelesen und in Tabellen gespeichert. Dann werden alle Vlankandidatengruppen erzeugt, die mit der verwendeten Anzahl an Hardwarerechnern möglich sind. Mit 16 Hardwarerechnern können die einzelnen Gruppen maximal 16 Mitglieder haben. Die Vlankandidatengruppen werden fortlaufend erzeugt, haben zwischen eins und 16 Mitglieder. Hier als Beispiel die Vlankandidatengruppen mittels Gedankenstrich getrennt und die Mitglieder mittels Komma, begonnen bei null: 0 - 1 - 2 - ... - 15 - 0,1 - 0,2 - ... - 0,15 - 1,2 - 1,3 - ... - 14,15 - 0,1,2 - 0,1,3 - ... - 14,15,16 - 0,1,2,3 - 0,1,2,4 - ... - 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15. Dies geschieht in einer Schleife in der immer die nächste Gruppe aus der vorhergehenden berechnet wird. Jede Gruppe wird als Tabelle gespeichert, siehe Listing 7.5.

Listing 7.6: Erzeugung aller möglichen Gruppen aus einer Anzahl Hardwarerechner

```
1 s=0
2 step=0
3 check=0
```

```
4 Gruppen_Mitglieder=array('-1')
5 while_check=0
6 while( while_check==0 ){
7   if ( Gruppen_Mitglieder[0]+1 <  max_rechner ){
8     Gruppen_Mitglieder[0] = Gruppen_Mitglieder[0] + 1
9     }
10  else {uebertrag=1
11        if (count( Gruppen_Mitglieder ) == 1 ){array_push( Gruppen_Mitglieder,-1 )}
12  }
13  s=1
14  while ( (s < count(Gruppen_Mitglieder)) AND (uebertrag == 1) AND (s <= max_gruppen
15    if (Gruppen_Mitglieder[s] + uebertrag + s == max_rechner){
16      uebertrag=1
17      if ( count(Gruppen_Mitglieder)< s + 2){array_push(Gruppen_Mitglieder,-1)}
18    }
19    else {
20      Gruppen_Mitglieder[s]=Gruppen_Mitglieder[s] + uebertrag
21      uebertrag=0
22      doit=1
23    }
24    step=s
25    s++
26  }
27  if (check==1){ //Bei Übertrag Neuinitialisierung der Gruppenmitglieder
28    i=step-1
29    while (i >= 0){
30      Gruppen_Mitglieder[i]=Gruppen_Mitglieder[i+1] + 1
31      i--;
32    }
33    check=0
34  }
35  array_push( gruppen, Gruppen_Mitglieder );
36  if ( count(Gruppen_Mitglieder) == max_gruppengr ){  // Abbruchbedingung
37    if ( Gruppen_Mitglieder[max_gruppengr-1] >= max_rechner-s ){
38      while_check=1
39    }
40  }
41 } // while
```

Nun wird, wie in Listing 7.3 beschrieben, aus den erzeugten Vlankandidatengruppen jede mögliche Kombination gebildet. Dann muss jede einzelne Kombination, mit allen aus der Configdatei ausgelesenen Hardwarerechnergruppen verglichen werden. Dazu wird eine rekursive Prüffunktion verwendet, siehe oben unter Prüffunktion. Diese Prüffunktion wird innerhalb der Schleife aufgerufen, die die Kombinationen der Vlankandidatengruppen errechnet. Somit werden alle Kombinationen aus Vlankandidatengruppen berücksichtigt, ebenso wie alle Hardwarerechnergruppen. Innerhalb der Prüffunktion werden die Kosten der einzelnen Lösungen errechnet. Nun kann man aus den Tabellen mit den Lösungen

ablesen, welche Kombination für welche Hardwarerechnergruppe die geringste Last erzeugt. Benötigt wird aber ein Ergebnis über alle Hardwarerechnergruppen hinweg. Deshalb wird nach Abschluss der Prüffunktion für eine Kombination von Vlankandidatengruppen mittels der Pfadprüfung die über alle Hardwarerechnergruppen gesehene beste Lösung bestimmt, siehe oben Pfadprüfung. Diese beste Lösung wird für diese Kombination aus Vlankandidatengruppen gespeichert. Nachdem alle besten Lösungen für die jeweiligen Kombinationen aus Vlankandidatengruppen bestimmt worden sind, kann aus diesen Lösungen die beste Lösung bestimmt werden. Die zu dieser Lösung gehörende Kombination aus Vlankandidatengruppen wird als Vlangruppen übernommen.

7.4.3 Groups

Bei dem Verfahren Groups wird versucht, die Anzahl der Überprüfungen gegenüber dem Verfahren All zu senken. Auch hier werden zuerst die Hardwarerechnergruppen aus der Configdatei ausgelesen. Dann werden aber nicht alle möglichen Vlankandidatengruppen erzeugt sondern es werden alle Hardwarerechnergruppen als Vlankandidatengruppen übernommen. Dann werden, wie in Listing 7.3 beschrieben, aus den Vlankandidatengruppen alle möglichen Kombinationen gebildet. Auch hier werden dann, mittels der rekursiven Prüffunktion, alle Kombinationen aus Vlankandidatengruppen mit allen Hardwarerechnergruppen verglichen, siehe oben unter Prüffunktion. Die Prüffunktion wird wie beim Verfahren All innerhalb der Schleife aufgerufen, in der die Kombinationen der Vlankandidatengruppen erzeugt werden, um alle Kombinationen berücksichtigen zu können. Ebenso werden alle Hardwarerechnergruppen berücksichtigt. Dann werden wie beim Verfahren All die Kosten der einzelnen Lösungen, innerhalb der Prüffunktion ermittelt. Nachdem alle Lösungen für alle Hardwarerechnergruppen mit der aktuellen Kombination aus Vlankandidatengruppen erstellt wurden, kann mit der Pfadprüffunktion die für diese Kombination aus Vlankandidatengruppen beste Lösung bestimmt werden. Dies wird für alle Kombinationen aus Vlankandidatengruppen wiederholt. Nach Abschluss des Verfahrens kann anhand der Lösungen der verschiedenen Kombinationen der Vlankandidatengruppen die insgesamt beste Lösung und somit die Vlangruppen bestimmt werden. Es werden aber nur 4080 Vlangruppen ausgewählt, da 16 Vlan-IDs für die direkte Vergabe von Vlangruppen an die Hardwarerechner benötigt wird.

7.4.4 Large Groups

Dieses Verfahren ist identisch mit dem Verfahren Groups bis auf die Auswahl der Vlankandidatengruppen. Bei Groups wurden alle Hardwarerechnergruppen als Vlankandidatengruppen gewählt, bei Large Groups werden die Hardwarerechnergruppen der Größe nach sortiert und dann die 4080 größten Gruppen ausgewählt zuzüglich der 16 Vlan-IDs für die einzelnen Hardwarerechner. Nun müssen nicht alle Kombinationen aus Vlankandidatengruppen gebildet werden, da die Vlangruppen bereits feststehen. Die Prüffunktion (s.o.)

mit der Kostenberechnung(s.o.) muss dennoch verwendet werden, da ermittelt werden muss, welche Hardwarerechnergruppe durch welche Vlangruppe zuzüglich möglichem direktem Senden ausgedrückt werden kann. Auch hier muss am Ende mittels einer Pfadprüfung(s.o.) die über alle Hardwarerechnergruppen beste Lösung ermittelt werden.

7.4.5 Zusammenfassen von Vlangruppen

Bei diesem Verfahren werden auch zuerst die Hardwarerechnergruppen aus der Configdatei eingelesen. Nun werden auch hier die Hardwarerechnergruppen als Vlankandidatengruppen verwendet. Unterschied zu dem Verfahren Large Groups ist, dass nun die Vlankandidatengruppen zusammengefasst werden. Die Vlankandidatengruppen werden zuerst der Größe nach sortiert. Dann werden immer die beiden kleinsten Gruppen zu einer neuen Vlankandidatengruppen zusammengefasst, siehe Listing 7.7. Dies wird solange fortgesetzt, bis nur noch 4080 Vlankandidatengruppen vorhanden sind. Diese werden mit den 16 Vlan-IDs, für die einzelnen Hardwarerechner, ergänzt. Nun wird wie bei dem Verfahren Large Groups auch die Prüffunktion(s.o.) mit der Kostenberechnung(s.o.) verwendet, um zu bestimmen, welche Hardwarerechnergruppe durch welche Vlangruppe zuzüglich möglichem Direktsenden ausgedrückt werden kann. Dann wird mittels der Pfadprüfung(s.o.) die über alle Hardwarerechnergruppen beste Lösung ermittelt.

Listing 7.7: Zusammenfassen von Gruppen

```
1 while ( count(Gruppen) > ID ){
2   Gruppen[1] = array_values(array_unique(array_merge(Gruppen[1],Gruppen[0])))
3   Gruppen[0] = Gruppen[ count(Gruppen) - 1 ]
4   unset( Gruppen[ count(Gruppen) - 1 ] )
5   array_multisort(Gruppen)
6 }
```

7.5 Kombinationen

Hier werden einige Verfahren miteinander kombiniert, um die Vorteile beider Verfahren nutzen zu können.

7.5.1 Direct - Broadcast

Bei diesem Verfahren werden Direct und Broadcast kombiniert. Es werden zuerst die Hardwarerechnergruppen aus der Configdatei ausgelesen. Als Vlan-IDs werden die 16 Vlan-IDs für die einzelnen Hardwarerechner und eine Vlan-ID für die Vlangruppe, in der alle Hardwarerechner Mitglied sind, verwendet. Nun wird die Prüffunktion(s.o.) mit der Kostenberechnung(s.o.) verwendet. Für jede Hardwarerechnergruppe werden

einmal die Kosten für Direct und einmal für Broadcast berechnet. Ist dies geschehen, kann mittels der Pfadprüfung die beste Zuordnung von Direct oder Broadcast zu jeder Hardwarerechnergruppe erfolgen.

7.5.2 Large Groups - Direct

Bei diesem Verfahren werden Large Groups und Direct kombiniert. Es werden ebenfalls die Hardwarerechnergruppen aus der Configdatei ausgelesen. Dann werden die Vlangruppen direkt aus den Hardwarerechnergruppen ausgewählt. Dafür werden die Hardwarerechnergruppen nach der Größe sortiert und dann die 4080 größten Gruppen als Vlangruppen verwendet. Hinzu kommen noch die 16 Vlan-IDs der einzelnen Hardwarerechner. Der Unterschied zu Large Groups ist, dass hier nicht versucht wird die Hardwarerechnergruppen, die keine Vlangruppe sind, durch die Vlangruppen auszudrücken sondern sie werden sofort mittels direktem Senden an die Vlan-IDs der einzelnen Hardwarerechner behandelt. Nun wird die Prüffunktion(s.o.) mit der Kostenberechnung(s.o.) nur zur Evaluierung der Messergebnisse benötigt. Eine Pfadprüfung(s.o.) ist hier nicht notwendig.

7.5.3 Large Groups - Direct - Broadcast

Bei diesem Verfahren werden Large Groups, Direct und Broadcast kombiniert. Zuerst werden die Hardwarerechnergruppen aus der Configdatei eingelesen. Dann werden die Vlangruppen ausgewählt. Wie bei dem Verfahren Large Groups - Direct werden die Hardwarerechnergruppen nach ihrer Größe sortiert und dann die 4079 größten Hardwarerechnergruppen, die 16 einzelnen Hardwarerechner und eine Gruppe, in der alle Hardwarerechner Mitglied sind, als Vlangruppen verwendet. Nun muss die Prüffunktion(s.o.) mit der Kostenberechnung(s.o.) verwendet werden da jede Hardwarerechnergruppe, die nicht als Vlangruppe verwendet wird entweder direkt oder mittels Broadcast versendet werden kann. Nach Durchlaufen der Prüffunktion stehen die Kosten für jede Hardwarerechnergruppe fest und es kann mittels der Pfadprüfung(s.o.) entschieden werden, welche Hardwaregruppe, die nicht als Vlangruppe verwendet wurde, besser direkt oder mittels Broadcast erreicht wird.

7.5.4 Middle Groups - Direct - Broadcast

Bei diesem Verfahren werden Middle Groups, Direct und Broadcast kombiniert. Auch hier werden zuerst die Hardwarerechnergruppen aus der Configdatei ausgelesen. Dann werden die Vlangruppen ausgewählt. Hierfür werden die Hardwarerechnergruppen ihrer Größe nach sortiert. Im Unterschied zu Large Groups - Direct - Broadcast werden hier nicht die größten Gruppen ausgewählt, sondern die mittleren. Nun wird die Prüffunktion(s.o.) mit der Kostenberechnung(s.o.) eingesetzt. Für alle Hardwarerechnergruppen, die nicht als Vlangruppen übernommen worden sind, werden hier die Kosten für direktes Senden und

Senden mittels Broadcast ermittelt. Das anschließende Ausführen der Pfadprüfung ergibt dann, welche Hardwarerechnergruppe, die nicht als Vlangruppe übernommen worden ist, besser mittels direktem Senden oder mittels Broadcast erreicht wird.

Kapitel 8

Evaluierung

Bei der Evaluierung werden die verschiedenen Verfahren zur Erstellung der Vlangruppen unter mehreren Szenarien getestet. Dabei werden möglichst unterschiedliche Szenarien zum Einsatz kommen um die Eigenschaften der unterschiedlichen Verfahren besser ausloten zu können. Die verwendeten Szenarien und die eingesetzten Tools werden hier vorgestellt.
Die unterschiedlichen Verfahren sind implementiert, um aus einem gegebenen, errechneten Szenario, das aus einer Configdatei eingelesen wird, eine möglichst effiziente Verteilung der Vlangruppen zwischen den Hardwarerechnern zu berechnen.
Als grundlegendes Kostenmodel für die Berechnung der Vlangruppen wird angenommen, dass ein Sendevorgang, ein Empfangsvorgang und das Verwerfen einer Nachricht alle eine Kosteneinheit teuer sind.
Mittels dieses Kostenmodels wird nun die Last auf allen Hardwarerechnern errechnet, indem die Anzahl der Sende-, Empfangsvorgänge und Verwerfungen von Nachrichten berechnet werden. Anhand dieser berechneten Last der Hardwarerechner wird die effizienteste Verteilung als die angesehen, die auf dem am stärksten belasteten Hardwarerechner die geringste Last hervorruft.

8.1 Szenarien

Es wird versucht möglichst unterschiedliche Szenarien zu generieren. Als flexible Größen werden die Anzahl der Hardwarerechner, die Anzahl verfügbarer Vlan-IDs und die Anzahl der Hardwarerechnergruppen, die durch die Vlangruppen ausgedrückt werden sollen, verwendet. Als weiterer Punkt wird berücksichtigt, dass in den Experimenten, die auf dem Cluster unter TVEE ausgeführt werden, auch unterschiedliche Bandbreiten für unterschiedliche Hardwarerechnergruppen zum Einsatz kommen. Eine Hardwarerechnergruppe mit einer größeren Bandbreite kann mehr Nachrichten versenden als eine Hardwarerechnergruppe mit geringer Bandbreite. Mehr versendete Nachrichten belasten den Hardwarerechner stärker. Deshalb wird in den Szenarien zu jeder Hardwarerechnergruppe eine zufällige

Bandbreite zwischen 1Mbit/s und 100Mbit/s festgelegt, die in den einzelnen Verfahren bei der Bestimmung der Vlankandidatengruppen berücksichtigt wird. Ebenso wird bei der Kostenberechnung die Bandbreite als Gewicht mit eingerechnet.
Außerdem gibt es zwei unterschiedliche Bereiche von Anwendungen, die berücksichtigt werden müssen. Einerseits Anwendungen, die eine feste Datenmenge übertragen müssen, und es nur darauf ankommt, wie lange die Übertragung der Datenmenge dauert, z.Bsp. Emailempfang oder das Herunterladen einer neuen Linuxdistribution. Andererseits das Szenario, in dem viele Rechner gleichzeitig ein LAN benutzen und zeitnahe Anwendungen ausführen, z.Bsp. zehn Anwender betrachten unterschiedliche Videostreams im selben LAN. Falls das LAN voll ausgelastet ist, werden in diesem Fall einfach Videoframes nicht rechtzeitig ihr Ziel erreichen können. Das bedeutet im zweiten Fall wird das LAN für eine definierte Zeit voll ausgelastet, unabhängig ob alle Daten übertragen werden konnten oder nicht.
Betrachtet wird aber nur der erste Fall. Im ersten Fall werden alle notwendigen Sende-, Empfangsvorgänge und notwendigen Verwerfungen von Nachrichten berücksichtigt für die Berechnung der Belastung der Hardwarerechner. Im zweiten Fall würde das Gewicht der verwendeten Bandbreite durch die Anzahl der beteiligten Rechner dividiert und somit der Einfluss einer größeren Anzahl an Rechnern im LAN berücksichtigt.

Um die Szenarien zu erstellen wird ein Szenariogenerator eingesetzt. Dieser erzeugt zu gegebenen Rahmenbedingungen beliebig viele zufällige Szenarien. Die Hardwarerechnergruppen werden in ihrer Größe und ihren Mitgliedern zufällig zwischen zwei Mitgliedern und der maximalen Anzahl an Hardwarerechnern variiert. Die Bandbreite wird als zufälliges Gewicht zwischen 1 und 100 für jede Hardwarerechnergruppe festgelegt.
Als Rahmenbedingungen wurden verwendet:
Anzahl der Hardwarerechner: 4, 8, 16 und 32.
Anzahl der verfügbaren Vlan-IDs: 256, 500, 1024, 2048, 4096.
Anzahl der vorhandenen Hardwarerechnergruppen: 500, 1000, 2500, 5000, 10000, 25000, 50000.
Für jede mögliche Kombination dieser Rahmenbedingungen wurden zehn zufällige Configdateien erzeugt und somit zehn Testläufe durchgeführt.

8.1.1 Szenariogenerator

Der Szenariogenerator akzeptiert als Parameter unter anderem die Anzahl der Hardwarerechner, die Anzahl der Vlan-IDs und die Anzahl der Hardwarerechnergruppen. Außerdem kann angegeben werden, wie viele zufällige Configdateien für ein Szenario erzeugt werden sollen. Mit Hilfe dieser Parameter erzeugt der Szenariogenerator Configdateien mit Hardwarerechnergruppen, die in Größe und Mitgliedern zufällig gewählt sind, also zwischen zwei Mitgliedern und der maximalen Anzahl Hardwarerechner an Mitgliedern haben. Außerdem werden die Bandbreiten als zufällige Gewichte für jede Hardwarerechnergruppe

erzeugt.

8.2 Ziel

Das Ziel ist es, das Verfahren zu finden, das es ermöglicht, mit möglichst geringer Maximalbelastung für den Cluster mehr Hardwarerechnergruppen zu verarbeiten als Vlan-IDs verfügbar sind. Die unterschiedlichen Verfahren werden auf die Szenarien angewendet und es werden die besonderen Eigenschaften jedes Verfahrens genauer betrachtet. Jeder Sende-, Empfangsvorgang und das Verwerfen von Nachrichten wird bei bei dieser Evaluierung als gleich teuer angesehen.

8.2.1 Overhead

Um die geringstmögliche Maximalbelastung ermitteln zu können, ist es notwendig eine Referenzgröße einzuführen. Hierfür wird ein Verfahren Benchmark eingeführt das später erläutert wird. Der Kostenunterschied zwischen dem Benchmarkverfahren und dem zu testenden Verfahren wird als Overhead angesehen. Je geringer dieser Overhead ist, desto weniger stark belastet die Lösung des zu testenden Verfahrens den Cluster.

8.2.2 Benchmark

Dieses Verfahren wird eingeführt um als Referenz zu dienen. Die anderen Verfahren werden mit den Ergebnissen dieses Verfahrens verglichen. Bei dem Benchmark Verfahren wird davon ausgegangen, dass so viele Vlan-IDs zur Verfügung stehen wie Hardwarerechnergruppen vorhanden sind. Das bedeutet es, wird immer ein optimales Ergebnis erzielt, unabhängig von der vorhandenen Anzahl an Vlan-IDs.

8.2.3 Laufzeit

Für alle Testläufe wird die Laufzeit, die jedes Verfahren benötigt um eine Verteilung der Vlangruppen zu ermitteln, gemessen. Nicht jedes Verfahren ist in der Lage, alle Szenarien in einer annehmbaren Zeit auszuwerten. Die Laufzeit hat also auch einen Einfluss darauf, welches Verfahren für ein bestimmtes Szenario am besten geeignet ist.

8.3 Ergebnisse

Im Folgenden werden die verschiedenen Verfahren mittels der unterschiedlichen Szenarien getestet. Es wird ermittelt, welches Verfahren für welches Szenario am besten geeignet ist und welches Verfahren im Allgemeinen die besten Resultate liefert. Außerdem wird ermittelt, ob alle Verfahren in annehmbarer Zeit die Szenarien abarbeiten können. In

den Abbildungen der Ergebnisse der einzelnen Verfahren werden entweder die Vlan-IDs oder die Anzahl der Hardwarerechnergruppen als feste Größe verwendet. Auf der Y-Achse wird der Overhead im Verhältnis des getesteten Verfahrens zum Verfahren Benchmark in Prozent aufgetragen. Auf der X-Achse ist die Hardwarerechnergruppenanzahl zu sehen, wenn die feste Größe der Abbildung die Anzahl der Vlan-IDs ist. Ist die feste Größe die Hardwarerechnergruppenanzahl, dann ist auf der X-Achse die Anzahl der Vlan-IDs zu sehen. Die für die jeweilige Abbildung feste Größe ist oberhalb des Diagramms abzulesen. Der auf der X-Achse aufgetragene Wert setzt sich zusammen aus jeweils zehn Szenarien da für jede Rahmenbedingung zehn Testläufe absolviert worden sind. Wenn z.Bsp. Hardwarerechnergruppenanzahlen auf der X-Achse aufgetragen sind und die erste Hardwarerechnergruppenanzahl ist 500, dann repräsentieren die ersten zehn Messpunkte im Diagramm die zehn Testläufe mit Hardwaregruppenanzahl 500 und den zehn dazugehörigen zufälligen Szenarien. Mit Hilfe dieser Messpunkte wurde ein Durchschnittswert gebildet, der als Hardwarerechner Durchschitt in den Abbildungen zu sehen ist, und mittels einer Linie mit den Durchschnittswerten der selben Hardwarerechneranzahl und der nächsten Größe auf der X-Achse verbunden ist. Dadruch können die Schwankungen der einzelnen Testläufe im Diagramm abgelesen werden.

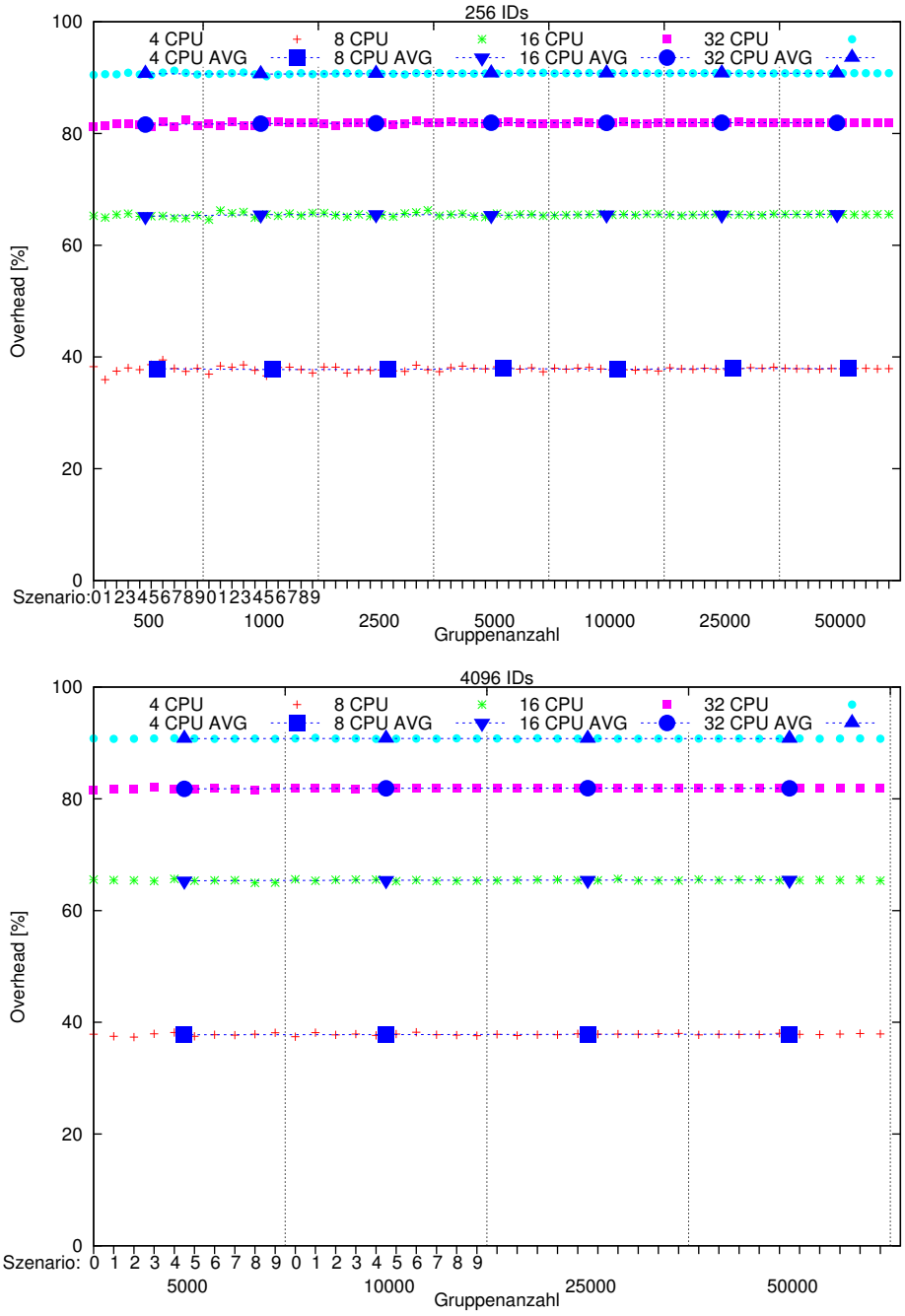
8.3.1 Direct

Das Verfahren Direct ist unabhängig von der verwendeten Anzahl an Vlan-IDs. Auch die Hardwarerechnergruppenanzahl hat keinen Einfluss auf den Overhead der Ergebnisse. Einzig die Anzahl an Hardwarerechnern ist entscheidend. Je mehr Hardwarerechner vorhanden sind, desto größer können die einzelnen Hardwarerechnergruppen werden(siehe Abbildung 8.1). Je größer die einzelnen Hardwarerechnergruppen sind, desto mehr Sendvorgänge und Empfangsvorgänge pro Rechner sind notwendig, um alle anderen Mitglieder einer Hardwarerechnergruppe mittels direktem Senden zu erreichen. Deshalb ist der Overhead in Abbildung 8.1 auch als Gerade zu erkennen. Je mehr Hardwarerechner in einem Szenario vorhanden sind, umso höher liegt die Gerade und umso höher ist der Overhead bei diesem Verfahren. Der Overhead steigt bei jeder Verdoppelung der Hardwarerechnerzahlen immer geringer an und nähert sich bei immer größeren Hardwarerechnerzahlen der 100% Overheadmarke an.

8.3.2 Broadcast

Das Verfahren Broadcast ist, ebenso wie Direct, unabhängig von der Anzahl der verfügbaren Vlan-IDs sowie unabhängig von der verwendeten Anzahl an Hardwarerechnergruppen. Der Overhead steigt auch bei dem Verfahren Broadcast mit wachsender Hardwarerechneranzahl wie in Abbildung 8.2 erkennbar ist. Je größer die Anzahl an Hardwarerechnern, desto größer auch die Hardwarerechnergruppen. Dadurch erfolgen pro Rechner mehr Empfangsvorgänge und auch mehr Nachrichten müssen verworfen werden. Der Anstieg des Overheads nimmt

Abbildung 8.1: Verfahren Direct



bei jeder Verdoppelung der Hardwarerechner immer mehr ab und nähert sich für große Hardwarerechnerzahlen 45% an.

8.3.3 Gruppen ersetzen

Die Verfahren, die versuchen Hardwarerechnergruppen durch andere Hardwarerechnergruppen zu ersetzen, haben das Problem, dass sie aufgrund der großen Anzahl an Vergleichen um die beste Vlangruppenkombination zu ermitteln eine extrem lange Laufzeit aufweisen. Außerdem wird eine sehr große Menge Arbeitsspeicher benötigt, um alle möglichen Lösungen zwischenzuspeichern und danach die Pfadprüfung(s. Kapitel 7) nach der besten Lösung durchzuführen. Keines der Verfahren ist in der Lage, eines der verwendeten Szenarien zu verarbeiten.
Daraufhin ist eine Modifikation an den Verfahren vorgenommen worden. Die Prüffunktion mit der Kostenberechnung (s. Kapitel7) speichert nur noch die aus Sicht der einzelnen Hardwarerechnergruppen beste Lösung. Dadurch ist nach Abschluss der Prüffunktion mit Kostenberechnung keine Pfadprüfung mehr notwendig. Als Folge fallen aber mögliche, bessere Lösungen weg. Zum Beispiel werden die Kosten für das Verwerfen von Nachrichten nicht über alle Hardwarerechnergruppen hinweg berücksichtigt, sodass es möglich sein kann, dass eine lokale Lösung mit vielen zu verwerfenden Nachrichten lokal aus Sicht einer einzelnen Hardwarerechnergruppe gut ist, aber die Kosten für die Verwerfungen sich mit den Kosten der anderen Hardwarerechnergruppen summieren und die lokal gute Lösung stark verschlechtern. Ebenso wird die Cachefunktion(s. Kapitel7) abgeschaltet um Arbeitsspeicher zu sparen.
Aber auch mit Hilfe dieser Modifikationen ist es nur möglich die kleinsten Szenarien mittels Groups, Large Groups und Zusammenfassen von Vlangruppen zu verarbeiten. Dass diese Verfahren extrem viel Laufzeit und Arbeitsspeicher in Anspruch nehmen war zu erwarten, dennoch war der Ressourcenbedarf höher als geplant. Aufgrund dessen können diese Verfahren leider nicht in der Evaluierung berücksichtigt werden.

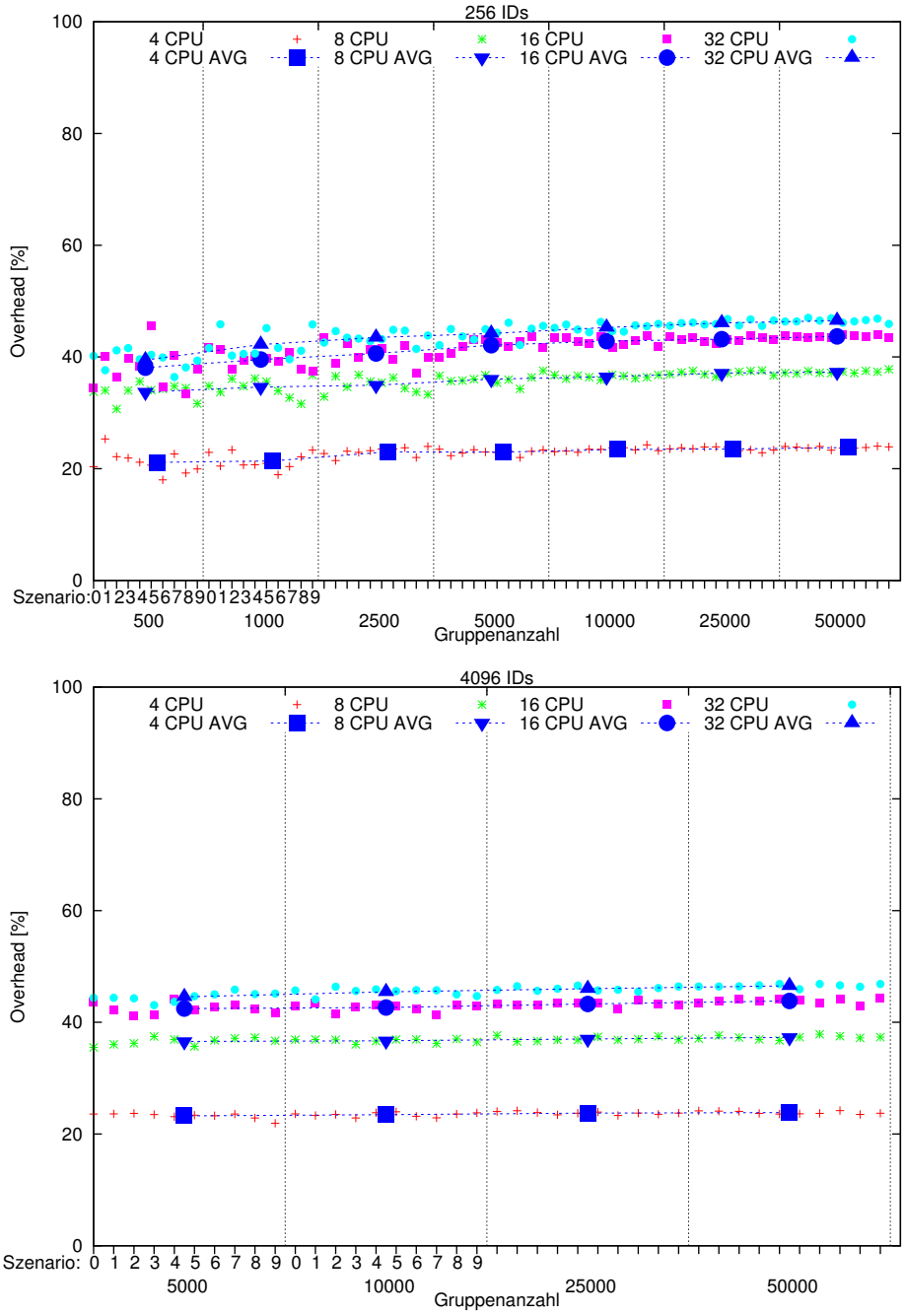
8.3.4 Kombinationen

Hier werden die Verfahren evaluiert, die durch die Kombination der oben erwähnten Verfahren entstehen. Es wird erwartet, dass die Vorteile beider Verfahren genutzt werden können, um ein besseres Resultat zu erzielen.

8.3.5 Direct - Broadcast

Die Kombination der beiden Verfahren Direct und Broadcast wird hier näher betrachtet. Wie aus den Testläufen der beiden Verfahren zu erkennen war, sind beide unabhängig von der Anzahl der Vlan-IDs und unabhängig von der Anzahl der vorhandenen

Abbildung 8.2: Verfahren Broadcast



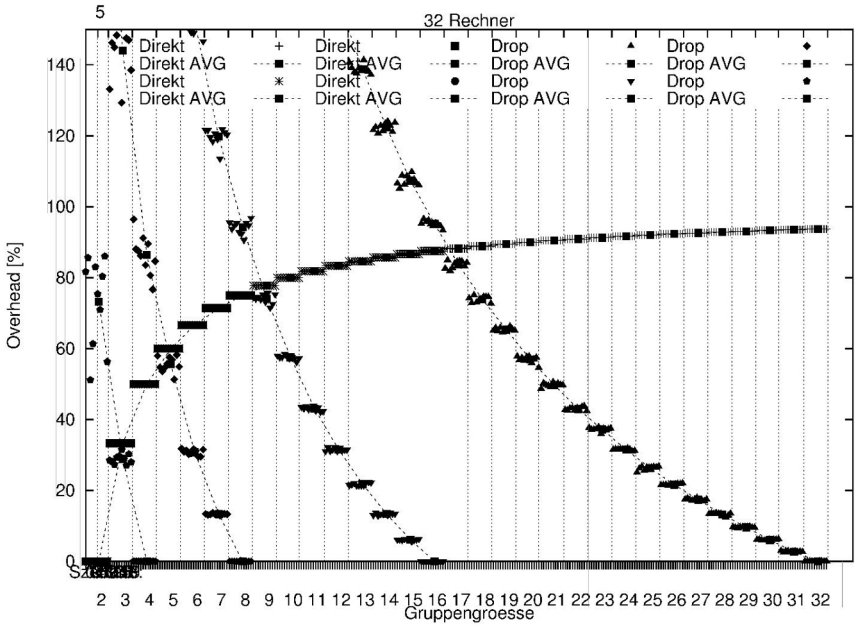
Hardwarerechnergruppen. Nur die Anzahl der Hardwarerechner und somit die Hardwarerechnergruppengröße haben einen Einfluss auf die Ergebnisse der Verfahren. Auch bei der Kombination der beiden Verfahren bleibt dieses Verhalten erhalten. Direct ist nach der Betrachtung des Verfahrens besser bei kleinen Hardwarerechnergruppengrößen als bei großen, da der Aufwand sowohl für Sendevorgänge und Empfangsvorgänge mit der Hardwarerechnergruppengröße zunimmt. Bei Broadcast ist das Verhalten für kleine Hardwarerechnergruppengrößen auf das Gesamtergebnis eher schlecht, da zwar die kleinen Hardwarerechnergruppengrößen nur wenig Sende- und Empfangsvorgänge benötigen, aber alle anderen Hardwarerechner, die nicht in den kleinen Gruppen Mitglied sind, erhalten eine Nachricht die sie verwerfen müssen. Bei großen Hardwarerechnergruppengrößen ist Broadcast dafür besser, da immer nur ein Sendevorgang notwendig ist und die Rechner, die Nachrichten verwerfen müssen, mit zunehmender Größe der Hardwarerechnergruppen weniger werden.

Um dieses Verhalten der Gruppen genau zu bestimmen, wird hier ein spezieller zusätzlicher Testlauf für beide Verfahren durchgeführt. Die Rahmenbedingungen für den Testlauf sind 4, 8, 16 und 32 Hardwarerechner, die Anzahl der Vlan-IDs ist nicht relevant, die Anzahl an Hardwarerechnergruppen ist 10000. Die Hardwarerechnergruppengrößen werden diesmal nicht zufällig gewählt. Es soll in dem Testlauf herausgefunden werden, wie sich die beiden Verfahren bei unterschiedlichen Hardwarerechnergruppengrößen verhalten. Es werden nun als Hardwarerechnergruppengrößen zwei Hardwarerechner bis hin zur maximalen Anzahl an Hardwarerechnern verwendet. Für die Rahmenbedingung mit vier Hardwarerechnern werden also die Hardwarerechnergruppengrößen zwei, drei und vier verwendet. Nun werden alle 10000 Hardwarerechnergruppen mit der gleichen Größe erstellt. Hardwarerechnergruppenmitglieder und Gewichte werden zufällig erzeugt. Nun wird das Verfahren Benchmark (s. Kapitel 7) auf die Szenarien angewendet, um einen Referenzwert zu erhalten für die beiden Verfahren Direct und Broadcast.
In Abbildung 8.4 kann man nun erkennen, dass Direct für kleine Hardwarerechnergruppengrößen gute Ergebnisse erreicht, aber für größere Hardwarerechnergruppen gegen die 100% Overhead strebt. Broadcast kann für Szenarien mit nur kleinen Gruppen keine guten Resultate liefern, wird aber immer besser, je größer die Hardwarerechnergruppen werden und strebt gegen 0% Overhead.

Eine Kombination der beiden Verfahren verspricht also bessere Ergebnisse als die Einzelverfahren zu liefern, da sich beide Verfahren gegenseitig ergänzen. Der Schnittpunkt der beiden Verfahren liegt genau bei der Hälfte der maximalen Hardwarerechnergruppengröße.

Da auch diese Kombination aus Direct und Broadcast die Prüffunktion mit Kostenberechnung mit anschließender Pfadprüfung benötigt um die besten Kombinationen zu ermitteln, ist eine Anwendung auf die Szenarien nicht möglich aufgrund von zu hohem

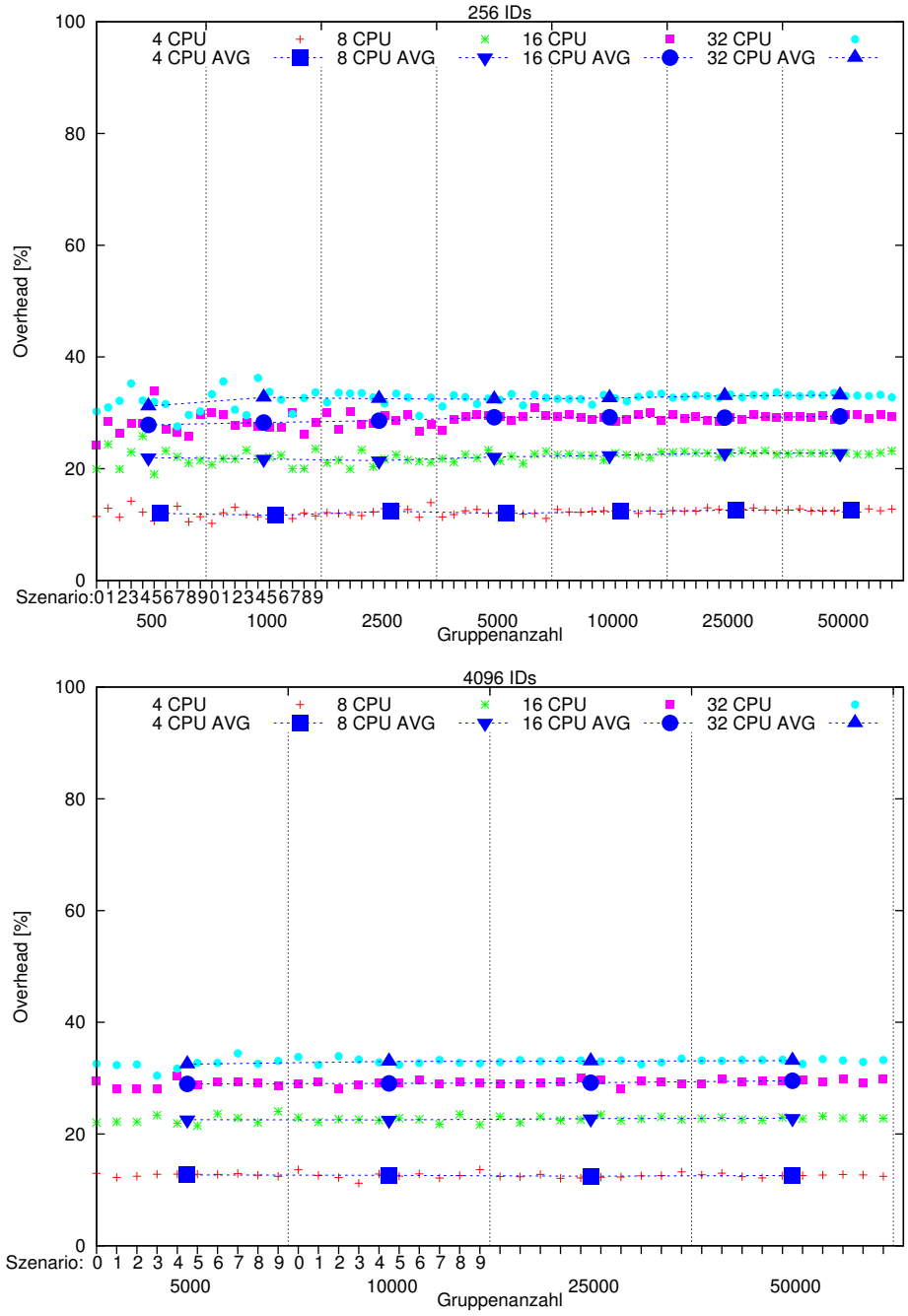
Abbildung 8.3: Direct und Broadcast Gruppengrößentestlauf



Arbeitsspeicherbedarf. Die unter Gruppen ersetzen(s.o.) angesprochenen Methoden zur Verringerung des Speicherbedarfs führen hier zu sehr schlechten Ergebnissen, da eine Einschränkung auf ein lokales bestes Ergebnis, die zu verwerfenden Nachrichten des Broadcast Verfahrens nicht für alle Hardwarerechnergruppen berücksichtigt.

Ein anderer Ansatz ist, eine Aufteilung der Hardwarerechnergruppen in zwei Bereiche für Direct und Broadcast vorzunehmen. Wie oben angeführt und in Abbildung 8.3 zu erkennen ist, liegt der Schnittpunkt der Kosten der beiden Verfahren genau bei der halben maximalen Hardwarerechnergruppengröße. Die Kombination der beiden Verfahren wird nun so modifiziert, dass alle Nachrichten an Hardwarerechnergruppen, die kleiner sind als die Hälfte der maximalen Hardwarerechnergruppengröße, mit dem Verfahren Direct gesendet werden. Alle Nachrichten an Hardwarerechnergruppen, die größer oder gleich groß sind wie die Hälfte der maximalen Hardwarerechnergruppengröße, werden mit dem Verfahren Broadcast versendet. Wie in Abbildung 8.4 zu erkennen ist, sind die Ergebnisse auch besser als die der Einzelverfahren. Besonders die Testläufe mit vier Hardwarerechnern, also die kleinen Hardwarerechnergruppengrößen, liefern bessere Ergebnisse.

Abbildung 8.4: Verfahren Direct - Broadcast



8.3.6 Large Groups - Direct

Bei der Kombination der Verfahren Large Groups und Direct werden die großen Hardwarerechnergruppen als Vlan-IDs übernommen und die restlichen Hardwarerechnergruppen werden mittels Direct versendet. Deshalb ist hier ein Einsatz der Prüffunktion mit Kostenberechnung(s. Kapitel 7) nicht notwendig, ebensowenig die Verwendung der Pfadprüfung (s. Kapitel 7). Der Einsatz von Large Groups ohne den Versuch die restlichen Hardwarerechnergruppen durch die Vlan-IDs auszudrücken, scheitert nicht an zu großem Arbeitsspeicherbedarf und liefert somit ein Ergebnis. Das Verfahren Direct ist für große Hardwarerechnergruppen nicht sehr gut geeignet, da die großen Hardwarerechnergruppen nun als Vlangruppen direkt ansprechbar sind, ist das Ergebnis der Kombination der beiden Verfahren besser als die einzelnen Verfahren. Wie in Abbildung 8.5 zu sehen ist, sind die Ergebnisse um so besser, je näher sich die Anzahl der Vlan-IDs und die Anzahl der Hardwarerechnergruppen sind, da in diesem Fall die meisten Hardwarerechnergruppen als Vlangruppe übernommen worden sind und direkt erreicht werden können. Je größer der Unterschied zwischen der Anzahl der Vlan-IDs und der Anzahl der Hardwarerechnergruppen ist, desto mehr größere Gruppen müssen dann durch das Verfahren Direct erreicht werden und die Kombination nimmt immer mehr die Charakteristik des Verfahrens Direct an, was zu einer Verschlechterung der Ergebnisse führt.

8.3.7 Large Groups - Direct - Broadcast

Bei der Kombination der Verfahren Large Groups und Direct und Broadcast werden die Vlan-IDs aus den großen Hardwarerechnergruppen gebildet und die restlichen nicht als Vlan-ID übernommenen Hardwarerechnergruppen werden dann entweder mit Direct oder mit Broadcast erreicht. Hier tritt wieder das Problem der Prüffunktion(s. Kapitel 7) auf, das oben unter Direct - Broadcast erwähnt ist. Hier wird deshalb auch der Ansatz verwendet, alle nicht durch Vlangruppen erreichbaren Hardwarerechnergruppen mit Direct zu erreichen, falls die Größe der Hardwarerechnergruppen kleiner als die Hälfte der maximalen Hardwarerechnergruppengröße ist und mittels Broadcast zu erreichen, wenn die Größe der Hardwarerechnergruppen kleiner als die Hälfte der maximalen Hardwarerechnergruppengröße ist. Dies führt dazu, wie bei der Kombination von Large Groups und Direct, wenn die Anzahl der Vlan-IDs nahe an der Anzahl der Hardwarerechnergruppen liegt, dass die Ergebnisse gut sind, da die meisten Hardwarerechnergruppen als Vlangruppen übernommen worden sind. Für den Fall, dass die Anzahl der Hardwarerechnergruppen viel größer ist als die Anzahl der Vlan-IDs, wird die Mehrheit der Nachrichten mittels Direct - Broadcast versendet was eine Verbesserung gegenüber Large Groups und Direct darstellt.

Abbildung 8.5: Verfahren Large Groups - Direct

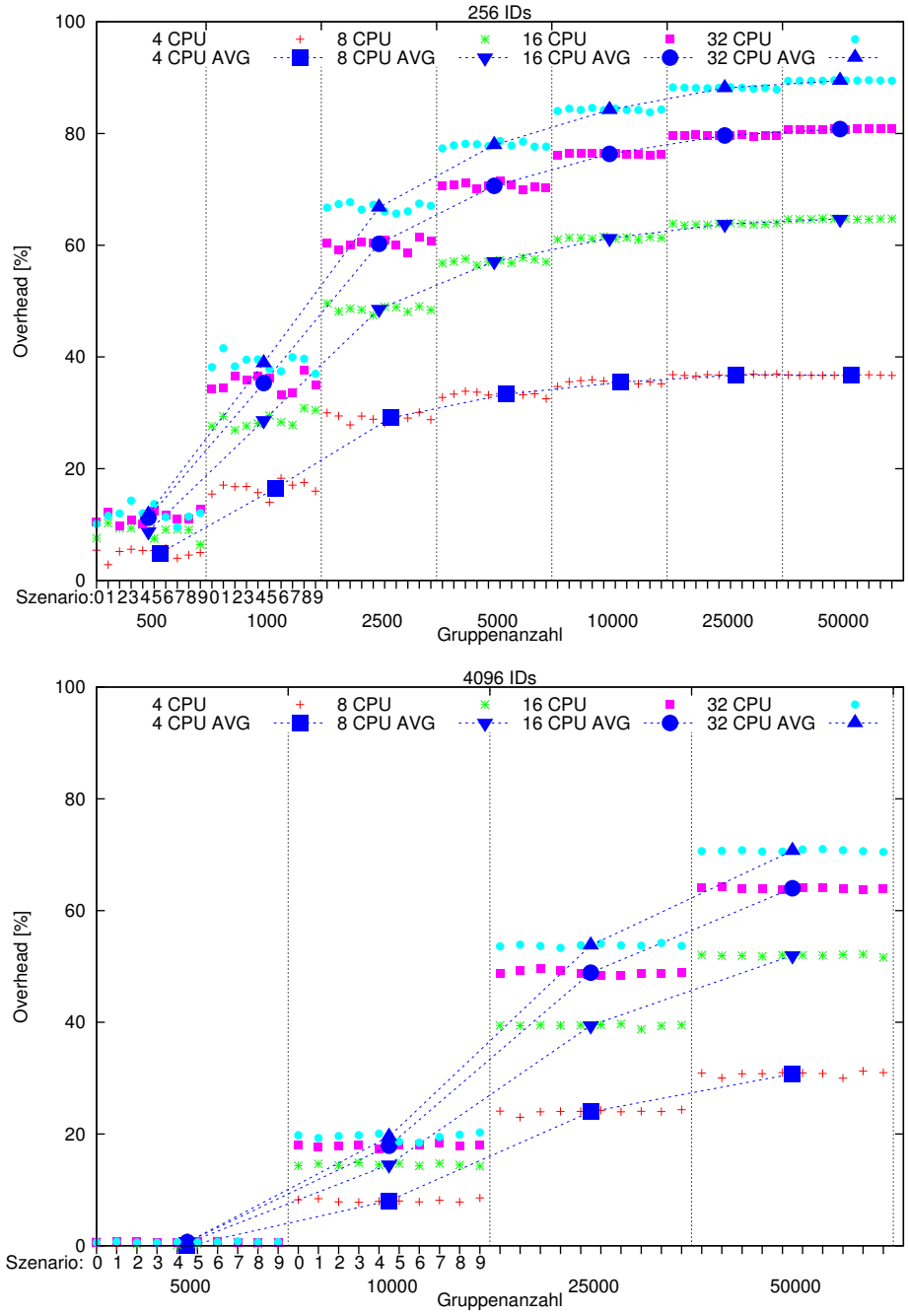
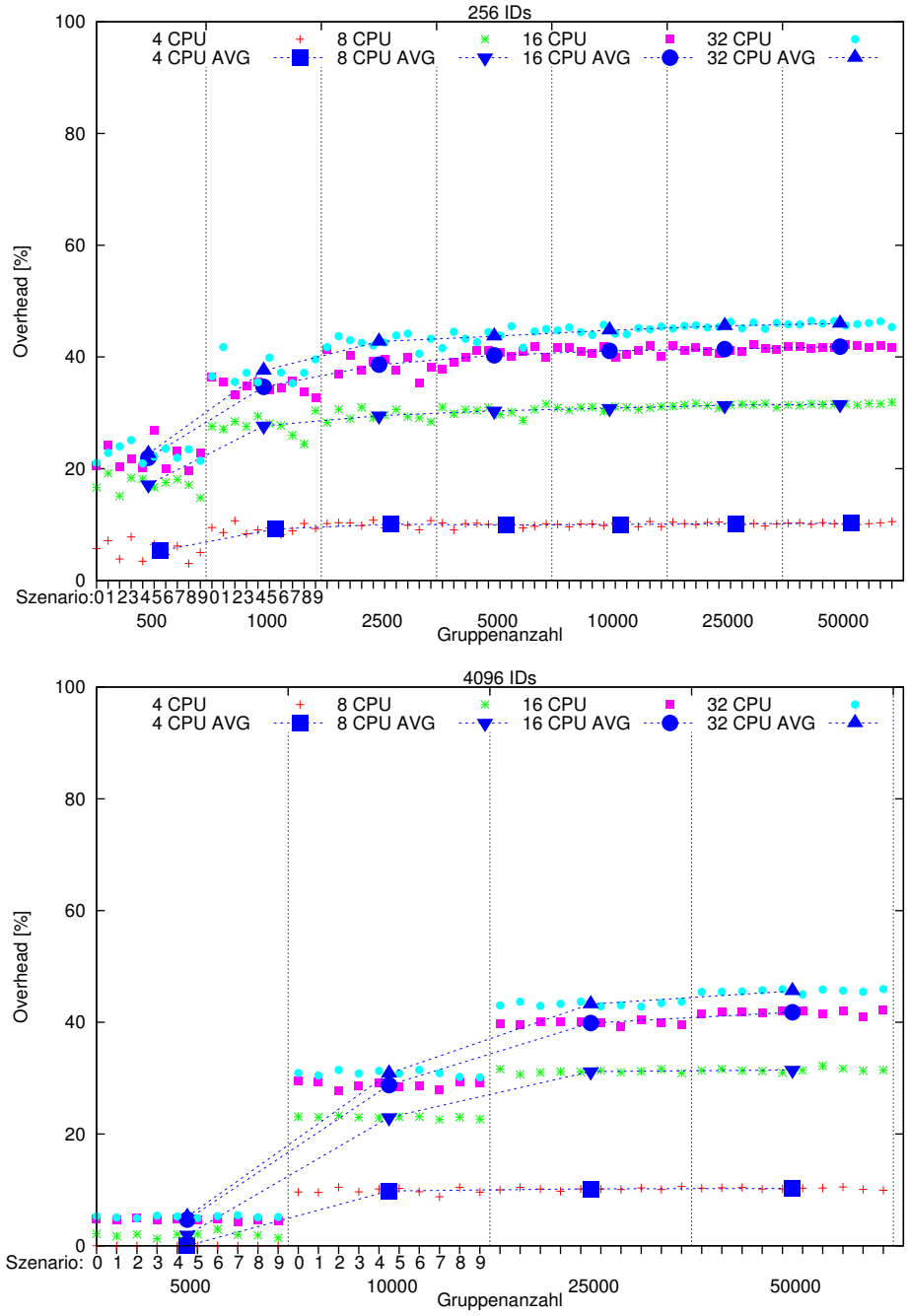


Abbildung 8.6: Verfahren Large Groups - Direct - Broadcast



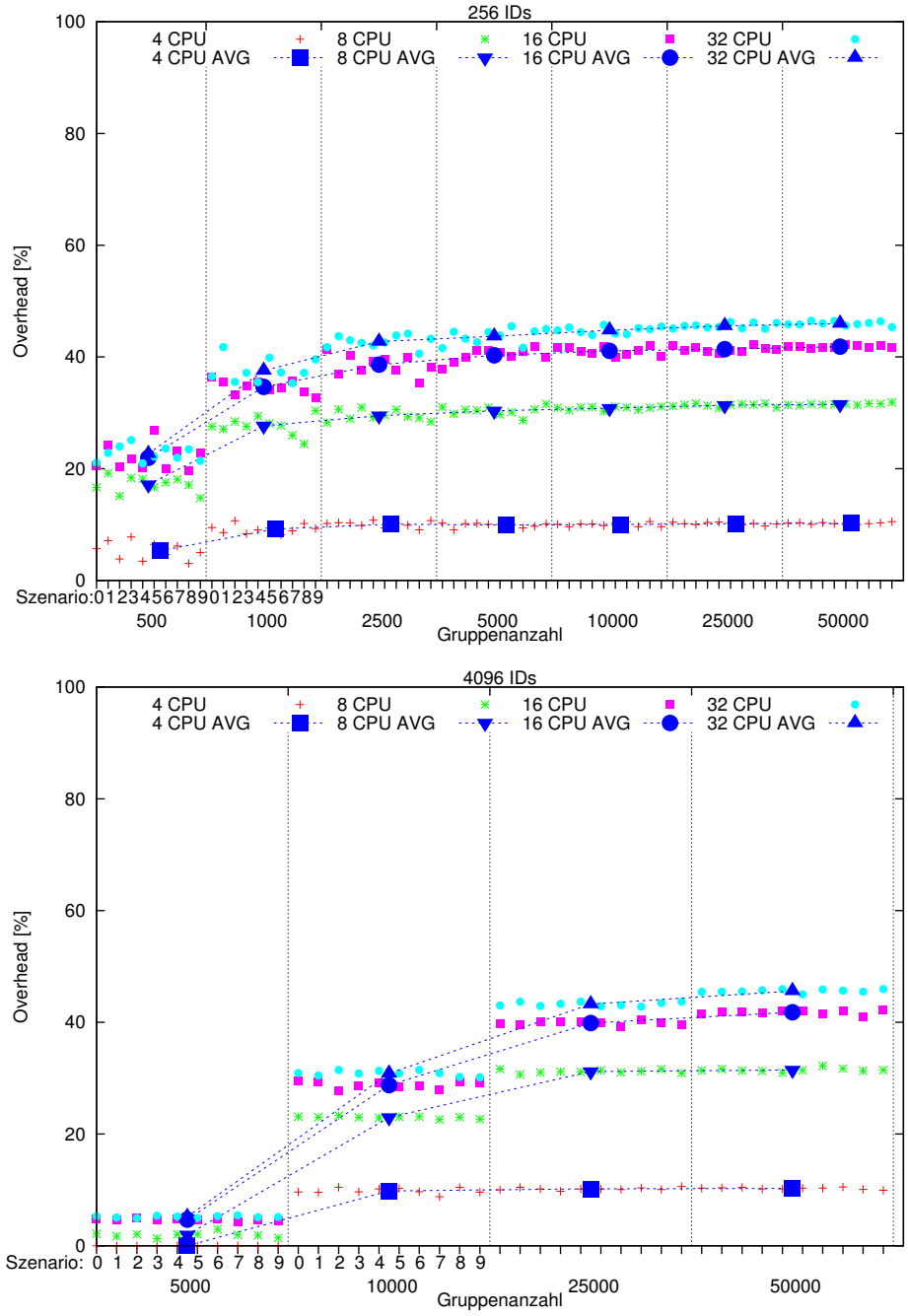
8.3.8 Middle Groups - Direct - Broadcast

Bei der Kombination von Middle Groups und Direct und Broadcast werden wieder die Hardwarerechnergruppen als Vlan-IDs verwendet. Hier werden jetzt nur im Unterschied zu Large Groups, in Anlehnung an Abbildung 8.7, die mittleren Hardwarerechnergruppengrößen verwendet. In Abbildung 8.3 ist zu erkennen, dass hauptsächlich die mittleren Gruppen, also der Bereich in dem sich Direct und Broadcast schneiden, der Bereich ist, in dem am meisten gespart werden kann.

8.4 Bewertung

Wie in den obigen Abbildungen erkennbar ist, kann mittels Kombinationen der einzelnen Verfahren unter Ausnutzung der Stärken der Verfahren eine Verbesserung des Ergebnisses erreicht werden. Das Verfahren Direct eignet sich gut für kleine Hardwarerechnergruppengrößen und ist ebenso wie Broadcast unabhängig von der Anzahl der Vlan-IDs und der Anzahl der verwendeten Hardwarerechnergruppen. Es eignet sich am Besten für Hardwarerechnergruppen mit zwei oder drei Mitgliedern.
Broadcast ist, wie in Abbildung 8.3 erkennbar, gut geeignet für große Hardwarerechnergruppengrößen. Die Kombination der beiden Verfahren ist besser als die einzelnen Verfahren, wie aus Abbildung 8.4 abzulesen ist. Die Hinzunahme von Vlangruppen zu den beiden Verfahren ergibt eine weitere Verbesserung. Wie Abbildung 8.3 zeigt, ist der Bereich der mittleren Hardwarerechnergruppengrößen der Bereich, in dem das Verfahren Direct und Broadcast am schwächsten ist.
Werden nun diese mittleren Hardwarerechnergruppen als Vlangruppen übernommen, dann ergibt sich das beste Ergebnis. Die Kombination aus Mittlere Gruppen, Direct und Broadcast liegt auch für sehr große Szenarien um 40% Overhead. Nur mit den Vlan-IDs, die für das Direktsenden benötigt werden, übersteigt es, wie man an Direct und Broadcast sieht, die 45% Overhead nicht.

Abbildung 8.7: Verfahren Middle Groups - Direct - Broadcast



Kapitel 9

Zusammenfassung und Ausblick

In diesem Kapitel werden die Resultate der Diplomarbeit, Skalierbare Emulation von Netzwerkverbindungen zwischen virtuellen Knoten, noch einmal zusammengefasst. Anschließend wird ein Ausblick auf weiterführende Arbeiten zu diesem Thema gegeben.

9.1 Zusammenfassung

In dieser Diplomarbeit wird eine Möglichkeit gesucht, die Einschränkungen der aktuellen Verwendung von Vlans auf dem Cluster des Projektes NET [NET11] unter der Virtualisierungsarchitektur TVEE aufzuheben oder zu erweitern.
Dazu wird in Kapitel 2 eine grundlegende Einführung in die notwendigen und verwendeten Komponenten gegeben, die für diese Diplomarbeit notwendig sind, wie z.Bsp. die Vorstellung der TVEE und der netzwerktechnischen Grundlagen.
Kapitel 3 gibt anhand von „Related Work“ einen Einblick in bereits existierende Arbeiten auf diesem Gebiet. Einige grundsätzliche Überlegungen dieser „Related Work“ können in die Entwurfskriterien in Kapitel 4 übernommen werden.
Die Entwurfskriterien in Kapitel 4 enthalten einige verschiedene Überlegungen, wie die Einschränkungen durch die verwendete Vlantechnologie erweitert oder auch durch grundsätzlich andere Ansätze ersetzt werden kann. Nach Abwägung sämtlicher Ansätze und der vorhandenen Einschränkungen, wie z.Bsp. durch die Eigenschaften realer verfügbarer Hardwareswitches, wird die Vlantechnologie beibehalten und erweitert und nicht durch ein neues Verfahren ersetzt.
In Kapitel 5 wird dann der Entwurf zu einem konkreten Konzept verfeinert. Die dabei entstehenden Schwierigkeiten und ihre Lösungsansätze werden beschrieben. Die dadurch nicht behebbare Einschränkung der Anzahl der verfügbaren Vlan-IDs auf der Ebene des Hardwareswitches bedarf einer intensiven Betrachtung. Es werden Verfahren entworfen, um die Verteilung der eingeschränkten Anzahl an Vlan-IDs zu optimieren.
In Kapitel 6 werden mögliche Ansätze für diese Verfahren diskutiert. Es gibt einige unter-

schiedliche Möglichkeiten die Verteilung der Vlan-IDs vorzunehmen.
Die Umsetzung der Verfahren zur Verteilung der Vlan-IDs wird in Kapitel 7 beschrieben. Hier wird auf die notwendigen Komponenten der Verfahren eingegangen.
In Kapitel 8 werden die einzelnen Verfahren evaluiert. Hier werden die Verfahren mittels unterschiedlicher Szenarien getestet. Es wird ermittelt, wie gut die Verteilungen der einzelnen Verfahren in Vergleich zu einen optimalen Benchmark sind. Hier wurde erkannt, dass ein Teil der Ansätze aus den vorhergehenden Kapiteln in realen Umständen nicht einsetzbar sind, da ihre Laufzeit und ihr Arbeitsspeicherbedarf viel zu hoch sind. Die funktionierenden Verfahren haben ergeben, dass für kleine Unterschiede in der Anzahl der verfügbaren Vlan-IDs und der Anzahl der Hardwarerechnergruppen ein sehr geringer Overhead erzielt werden kann, und selbst bei Verwendung der minimal notwendigen Anzahl an Vlan-IDs kann steigt der Overhead nicht über 40%. Dadurch wird mit dieser Diplomarbeit ein Konzept gegeben, mit dem die Einschränkungen durch die Verwendung der Vlantechnologie auf dem Cluster des Projektes NET unter der Virtualisierungsarchitektur TVEE stark erweitert werden können. Mit den Verfahren zur effizienten Verteilung der Vlangruppen auf der Hardwareswitchebene kann auch dieses Hinderniss mit einem akzeptablen Overhead kompensiert werden.

9.2 Ausblick

Das entwickelte Konzept muss noch in die Vlandevices und virtuellen Brücken der XEN Domain0 integriert werden.

Bei den Verfahren zur effizienten Verteilung der Vlangruppen muss überlegt werden, wie sich das Verfahren „Middle Groups - Direct - Broadcast“ verändert, falls die Kosten für Empfangsvorgänge und das Verwerfen von Nachrichten nicht identisch sind und sich somit die Kostenkurve des Verfahrens Broadcast gesehen über die Hardwarerechnergruppengröße verlagert.
Des weiteren kann überlegt werden, wie mobile virtuelle Knoten am besten in das Konzept intergriert werden können, da ein mobiler Knoten innerhalb seiner Kollisionsdomäne unterschiedliche Knoten erreichen kann, und jede Verbindung kann eine unterschiedliche Bandbreite haben. In dieser Diplomarbeit wurden nur Kollisionsdomänen mit der gleichen Bandbreite berücksichtigt. Das hat einen Einfluss auf die Gewichte in der Evaluierung.

Literaturverzeichnis

[AH06a] G. Apostolopoulos, C. Hassapis. V-eM: A Cluster of Virtual Machines for Robust, Detailed, and High-Performance Network Emulation. Technical report, ICS-FORTH, Greece, 2006.

[AH06b] G. Apostolopoulos, C. Hassapis. V-eM: A Cluster of Virtual Machines for Robust, Detailed, and High-Performance Network Emulation. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, pp. 117 – 126. 2006. doi:10.1109/MASCOTS.2006.51.

[BDF+03] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield. Xen and the art of virtualization. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pp. 164–177. ACM, New York, NY, USA, 2003. doi:http://doi.acm.org/10.1145/945445.945462. URL `http://doi.acm.org/10.1145/945445.945462http://www.xen.org`.

[Bel05] F. Bellard. *Qemu, a Fast and Portable Dynamic Translator. In Usenix annual technical conference.*, 2005. URL `http://www.qemu.org`.

[Boc] *BOCHS*. URL `http://bochs.sourceforge.net/`.

[CIS] *Cisco 10000 Series Routers IEEE 802.1Q-in-Q VLAN Tag Termination*. URL `http://www.cisco.com/en/US/docs/routers/10000/10008/feature/guides/qinq.html`.

[Def81] Defense Advanced Research Projects Agency. *RFC791 - INTERNET PROTOCOL*, 1981. URL `http://wiki.tools.ietf.org/pdf/rfc791.pdf`.

[GHR09] A. Grau, K. Herrmann, K. Rothermel. Efficient and Scalable Network Emulation Using Adaptive Virtual Time. In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pp. 1 –6. 2009. doi:10.1109/ICCCN.2009.5235306.

[GHR11] A. Grau, K. Herrmann, K. Rothermel. NETbalance: Reducing the Runtime of Network Emulation using Live Migration. In *Proceedings of the 20th International Conference on Computer Communication Networks (ICCCN'11)*, pp. 1–6. IEEE Computer Society, Maui, HI, USA, 2011. URL `http://www2.informatik.uni-stuttgart.de/cgi-bin/NCSTRL/NCSTRL_view.pl?id=INPROC-2011-22&engl=`.

[GMHR08] A. Grau, S. Maier, K. Herrmann, K. Rothermel. Time Jails: A Hybrid Approach to Scalable Network Emulation. In *Principles of Advanced and Distributed Simulation, 2008. PADS '08. 22nd Workshop on*, pp. 7 –14. 2008. doi:10.1109/PADS.2008.19.

[IEEa] IEEE, New York. *IEEE 802.1Q - Virtual Bridged Local Area Networks*, 2005 edition. URL `http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf`.

[IEEb] IEEE, New York. *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*. URL `http://standards.ieee.org/getieee802/download/802-2001.pdf`.

[IEE08] IEEE, New York. *IEEE Std 802.3 - Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. Section One*, 2008. URL `http://standards.ieee.org/getieee802/download/802.3-2008_section1.pdf`.

[IETa] IETF. *RFC3376 - Internet Goup Management Protocol v3*. URL `http://www.rfc-editor.org/rfc/pdfrfc/rfc3376.txt.pdf`.

[IETb] IETF. *RFC3931 - Layer Two Tunneling Protocol - Version 3 (L2TPv3)*. URL `http://tools.ietf.org/pdf/rfc3931.pdf`.

[IETc] IETF. *RFC4541.pdf - IGMP Snooping switches*. URL `http://wiki.tools.ietf.org/pdf/rfc4541.pdf`.

[IET89] IETF. *RFC1112 - Host Extensions for IP Multicasting*, 1989. URL `http://tools.ietf.org/pdf/rfc1112.pdf`.

[jav] *Java*. URL `http://www.java.com`.

[MGWR07] S. Maier, A. Grau, H. Weinschrott, K. Rothermel. Scalable Network Emulation: A Comparison of Virtual Routing and Virtual Machines. In *Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on*, pp. 395 –402. 2007. doi:10.1109/ISCC.2007.4381529.

[MRBV05] P. Mahadevan, A. Rodriguez, D. Becker, A. Vahdat. MobiNet: a scalable emulation infrastructure for ad hoc and wireless networks. In *Papers presented at the 2005 workshop on Wireless traffic measurements and modeling*, WiTMeMo '05, pp. 7–12. USENIX Association, Berkeley, CA, USA, 2005. URL http://portal.acm.org/citation.cfm?id=1072430.1072432.

[NET11] NET. Network Emulation Testbed, 2011. URL http://net.informatik.uni-stuttgart.de/.

[ope] *OpenVZ*. URL http://www.openvz.org.

[rfc] *RFC2784 - Generic Routing Encapsulation (GRE)*. URL http://tools.ietf.org/pdf/rfc2784.pdf.

[RFC99] IETF. *RFC 2661 - Layer Two Tunneling Protocol L2TP*, 1999. URL http://tools.ietf.org/html/rfc2661.

[SCDB+09] K. Singh, T. Castiglion, L. Dymoke-Bradshaw, P. Hanninen, J. Vincente Ranieri, P. Kappeler. *Security on the IBM Mainframe*. Number SG24-7803-00 in IBM Redbooks. IBM, 2009. URL http://publib-b.boulder.ibm.com/abstracts/sg247803.html?Open.

[Tan03] A. S. Tanenbaum. *Computer Networks , 4th edition*. Pearson Studium, 2003.

[vira] *VirtualBox*. URL http://www.virtualbox.org/.

[virb] *Virtuozzo*. URL http://www.parallels.com/.

[vmw] *VMWARE*. URL http://www.vmware.com/.

[ZN04] P. Zheng, L. Ni. EMPOWER: a cluster architecture supporting network emulation. *Parallel and Distributed Systems, IEEE Transactions on*, 15(7):617 – 629, 2004. doi:10.1109/TPDS.2004.21.

Erklärung

Ich versichere, dass ich diese Arbeit selbständig verfasst und nur die angegebenen Hilfsmittel verwendet habe.

Stuttgart, _____

_____

Markus Schirmer