

Aflevering van gepersonaliseerde multimediale data in peer-to-peer netwerken

Birger Anckaert

Promotor: prof. dr. ir. Rik Van de Walle

Begeleiders: dr. Davy Van Deursen, Erik Mannens

Masterproef ingediend tot het behalen van de academische graad van
Master in de ingenieurswetenschappen: computerwetenschappen

Vakgroep Elektronica en informatiesystemen
Voorzitter: prof. dr. ir. Jan Van Campenhout
Faculteit Ingenieurswetenschappen
Academiejaar 2009-2010



DANKWOORD

Alvorens aan het echte werk begonnen wordt, wil ik iedereen die ondersteuning heeft geboden bij de opmaak van de masterproef en het afleveren ervan van harte bedanken. In het bijzonder denk ik aan mijn begeleider, dr. Davy Van Deursen, die mij gedurende het jaar voortdurend gemotiveerd heeft en voor belangrijke feedback bij mijn nota's en ideeën zorgde. Daarnaast had ik ook graag mijn vriend Bart Coppens vermeld, die dit werk heeft overlezen en wiens commentaren hebben bijgedragen tot de leesbaarheid van mijn uiteenzettingen.

Graag had ik prof. Rik Van de Walle bedankt, voor het uitschrijven van een masterproef met een onderwerp dat mij meteen aansprak en mij zo de mogelijk te bieden mij een jaar in dit thema te verdiepen. Verder zijn er alle leden van de JXTA gemeenschap die met hun snelle en uitgebreide antwoorden op mijn forumberichten mij veelvuldig uit een impasse hebben gehaald tijdens het programmeren.

Tot slot denk ik aan mijn ouders die de afgelopen jaren in mij zijn blijven geloven en van wie ik de kansen en de mogelijkheden heb gekregen om deze studies af te werken.

TOELATING TOT BRUIKLEEN

De auteur geeft de toelating deze masterproef voor consultatie beschikbaar te stellen en delen van de masterproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze masterproef.

Datum

Handtekening

Aflevering van gepersonaliseerde multimediale data in peer-to-peer netwerken

door

Birger Anckaert

Afstudeerwerk ingediend tot het behalen van de graad van Burgerlijk ingenieur in de computerwetenschappen

Academiejaar 2009-2010

Universiteit Gent

Faculteit Ingenieurswetenschappen

Vakgroep Elektronica en Informatiesystemen

Voorzitter: prof. dr. ir. J. Van Campenhout

Promotor: prof. dr. ir. R. Van de Walle

Masterproefbegeleiders: dr. Davy Van Deursen, Erik Mannens

Samenvatting

Deze masterproef onderzoekt mogelijke oplossingen voor het afleveren van gepersonaliseerde multimediasamenstellingen over een peer-to-peer architectuur. In de eerste plaats wordt een prototype peer-to-peer protocol samengesteld uit een combinatie van elementen uit bestaande protocollen, aangevuld met optimalisaties voor het stromen van multimedia in het bijzonder. Naast een beschrijvende specificatie van het protocol worden ook suggesties en codevoorbeelden gegeven voor een basisimplementatie in het JXTA framework. Ten tweede is het de bedoeling de multimedia afgeleverd via dit protocol aan te passen aan de noden en eisen van de gebruiker. Hiervoor gaan we uit van een reeds aanwezige multimedia-annotatie in RDF die zowel de inhoud als de technische kenmerken van de gedeelde multimedia omvatten. De specificaties van de gebruiker en het weergavemedium worden aan deze metadata getoetst. Daarna is het de bedoeling de geselecteerde multimedia met verschillende coderingen en van uiteenlopende types zonder transcoding op een efficiënte manier over dit netwerk te stromen en geaggregeerd weer te geven.

Trefwoorden: P2P multimedia streaming, peer-to-peer multimedia streaming, gepersonaliseerde multimedia, metadata verwerking bij multimedia, RDF-geannoteerde multimedia, JXTA

Delivery of personalized multimedia content on a peer-to-peer network

Birger Anckaert

Supervisor(s): dr. Davy Van Deursen, Erik Mannens, prof. dr. ir. Rik Van de Walle

Abstract - This article will describe the design of a peer-to-peer multimedia streaming protocol optimized for personalized multimedia content distribution. The first phase of the development entails a specification of an optimal peer-to-peer architecture for general multimedia streaming purposes. In a second phase an automated metadata driven framework for multimedia editing is introduced in order to obtain personalized multimedia presentations. The gained multimedia view will then be transported through the streaming infrastructure specified in the first phase of development.

Keywords - peer-to-peer multimedia streaming, personalized multimedia streaming, automated multimedia editing, metadata driven personalization of multimedia.

I. Introduction

Broadband technology advances have allowed the successful deployment of multimedia streaming for video-conferencing and surveillance applications over private networks for quite some time now. However streaming high-quality multimedia over the Internet to address the constantly growing quality demands of the end user still remains a challenging issue. In recent years, peer-to-peer (P2P) technology has captured the interest of the research community as well as the industry. By allowing peers to serve each other, P2P solutions overcome many limitations of traditional client-server architectures. They can handle flash crowds as well as achieve bandwidth scalability. In addition, P2P solutions do not require any special support from the underlying network topology. As a result, there has been an increasing interest in the use of P2P architecture for large-scale, high-quality multimedia streaming.

With the development of the Semantic Web metadata has become of growing interest to software developers. Many tasks can be automated and integrated through the use of a diligent metadata annotation. By introducing a metadata annotation to multimedia it might be possible to create an automated multimedia editing platform which supports the creation of multimedia presentations solely based on the use of meta information.

When both technologies are combined, a peer-to-peer protocol with streaming capabilities for personalized multimedia content is formed.

The remainder of this paper is structured as follows: Section 2 provides a quick overview of the proposed P2P architecture. Section 3 introduces the inner workings of the personalization mechanism. Finally section 4 has some conclusions about the specification en suggestions for future work.

II. P2P streaming protocol

The proposed P2P streaming architecture is a combination of several strong points of existing architectures. The protocol is closely modeled to the BitTorrent (1) (2) approach, with the main exception of the tracker system. Instead of tracker servers, super users much like the Super Nodes in the FastTrack protocol specification are used. This way the system has no need for a centralized bootstrapping mechanism.

Because of the inherent sequential encoding and playback of multimedia, and the generally larger file sizes of the media types some modifications to the standard P2P transport protocols had to be made in order to accommodate multimedia streaming.

Firstly, due to the parallel transport of multiple multimedia chunks over a best effort protocol, the order of arrival of the pieces could be disturbed. The introduction of a simple reorder buffer between the arrival of the multimedia fragment and its playback should fix this problem. Secondly, in order to secure some progress during the playback of the streaming multimedia a playback buffer is added to anticipate for bandwidth fluctuations. One and the same buffering mechanism can actually be used for both buffer services. The output of the buffer mechanism is fed to a generic multimedia playback framework in order to view the transferred multimedia presentation.

III. Personalized multimedia

In order to have an idea of the network multimedia content, all shared media already should be annotated with a usable metadata vocabulary. A practical multimedia annotations environment will have to fulfill certain requirements (2). RDF is most able to implement all of these concerns in a suitable multimedia annotation environment, due to its adaptive nature.

Once all annotation needs are tended to, an automated multimedia environment could use a powerful query language (e.g. SPARQL) to isolate multimedia fragments which peak a users' interest. These fragments can be aggregated in such a way its composition is tailored specifically to the needs of one user.

IV. Conclusions

When the specifications proposed in this paper are implemented to the letter, a very rudimentary nevertheless working protocol for P2P streaming of personalized multimedia content is gained.

The implementation of the proposed ideas was never meant to be an applicable protocol API. It was intended as a mere proof-of-concept in order to appraise the practical possibility to create such a protocol. The application has never been deployed on more than one machine and was always tested on a locally emulated network. Suffice to say the application has some shortcomings.

Some future optimizations and additions to the protocol may include but are not limited to: the optimization of the configuration of peer groups, the implementation of an adaptive reorder buffer mechanism, the specification of a practical multimedia annotations environment, and the integration of a full-fledged automated multimedia editing environment.

References

1. **Shah, Purvi en Pâris, Jehan-François.** *Peer-to-Peer Multimedia Streaming Using BitTorrent.* [document] Houston : University of Houston, 2007.
2. **Xiaojun, Hei, Lui, Yong en Ross, Keith W.** *IPTV over P2P Streaming Networks: the Mesh-pull Approach.* [document] Brooklyn : Polytechnic University, 2008.
3. **Geurts, Joost, van Ossenbruggen, Jacco en Hardman, Lynda.** *Requirements for practical multimedia annotation.* [document] Amsterdam : CWI, 2005.

INHOUDSOPGAVE

Hoofdstuk 1 Inleiding.....	1
1.1 Probleemstelling.....	1
1.1.1 Multimedia overdracht.....	1
1.1.2 Metadata en multimedia.....	4
1.2 Overzicht.....	5
1.2.1 Doel.....	5
1.2.2 Methode.....	6
Hoofdstuk 2 P2P multimedia streaming protocol: state of the art.....	8
2.1 Waarom P2P?.....	8
2.2 Evolutie.....	9
2.2.1 1 ^{ste} Generatie P2P-applicaties.....	9
2.2.2 2 ^{de} Generatie P2P-applicaties.....	10
2.2.3 3de generatie P2P-applicaties.....	12
2.4 Overlays.....	14
2.4.1 Enkelvoudige boom.....	16
2.4.2 Meervoudige boom.....	16
2.4.3 Net.....	17
2.5 Conclusie.....	18
Hoofdstuk 3 P2P multimedia streaming protocol: Specificatie.....	20
3.1 Vooropgestelde vereisten.....	20
3.1.1 P2P-architectuur.....	20
3.1.2 Multimedia kenmerken.....	21
3.1.3 Personalisatie mogelijkheden.....	21
3.2 Traditionele P2P-actoren in een hybride systeem.....	22
3.2.1 Peergroep.....	22
3.2.2 Superknoop.....	22
3.2.3 Normale peer.....	23
3.3 Specificatie.....	24
3.3.1 Aanmelden.....	24
3.3.2 Delen.....	25
3.3.3 Zoeken.....	26
3.3.4 Bestandsoverdracht.....	27
3.3.5 Weergave.....	29
3.3.6 Afmelden.....	29

Hoofdstuk 4 P2P multimedia streaming protocol: Implementatie.....	30
4.1 Waarom JXTA?.....	30
4.1.1 Interoperabiliteit.....	30
4.1.2 Platformafhankelijkheid.....	31
4.1.3 Alomtegenwoordigheid.....	31
4.1.4 Motivatie.....	31
4.2 JXTA werking.....	32
4.2.1 JXTA onderdelen.....	33
4.3 Implementatie.....	35
4.3.1 Aanmelden.....	36
4.3.2 Delen.....	39
4.3.3 Zoeken.....	39
4.3.4 Bestandsoverdracht.....	39
4.3.5 Weergave.....	40
Hoofdstuk 5 Personalisering van multimedia: state of the art.....	42
5.1 Metadata-annotaties voor multimedia.....	42
5.1.1 Overzicht van bestaande standaarden.....	42
5.1.2 Problemen met bestaande standaarden.....	43
5.2 Geautomatiseerde multimedia productie.....	44
Hoofdstuk 6 Personalisering van multimedia: specificatie.....	46
6.1 Vooropgestelde vereisten.....	46
6.1.1 Vereisten voor de metadata-annotatie.....	46
6.1.2 Vereisten voor de personalisatie van de multimedia.....	47
6.2 Specificatie.....	48
6.2.1 Specificatie van de metadata-annotatie.....	48
6.2.2 Specificatie van de personalisatie van de multimedia.....	48
Hoofdstuk 7 Personalisering van multimedia: implementatie.....	50
7.1 Waarom RDF?.....	50
7.1.1 SPARQL.....	50
7.1.2 JENA.....	51
7.2 Implementatie van de metadata in het P2P-streaming protocol.....	52
7.2.1 Integratie van de metadatacatalogoog in het P2P-systeem.....	52
7.2.2 Zoeken in de metadatacatalogoog van het P2P-systeem.....	52
7.3 Implementatie van de geautomatiseerde multimediapublicatie in het P2P-streaming protocol.....	52

Hoofdstuk 8 Conclusie	54
8.1 Geïmplementeerde functionaliteit van het protocol.....	54
8.2 Uitbreidingen van het protocol naar de toekomst toe	54
8.2.1 P2P-overdracht.....	54
8.2.2 Personalisatie van de multimedia.....	55

LIJST VAN FIGUREN

Fig. 1.1: Demonstratie van het unicast principe.....	1
Fig. 1.2: Demonstratie van het multicast principe.....	2
Fig. 1.3: Vereenvoudigde figuur over de samenstelling van een gepersonaliseerde multimediapresentatie a.d.h.v. metadata.....	4
Fig. 1.4: Relatie tussen multimedia en annotaties.....	5
Fig. 2.1: Algemeen schema voor 1ste generatie P2P-applicaties.....	10
Fig. 2.2: Verbinden van een knoop met een Gnutella-netwerk.....	11
Fig. 2.3: Zoekopdracht en bestandsoverdracht binnen een FastTrack-netwerk.....	12
Fig. 2.4: Interactie binnen een BitTorrent opstelling.....	14
Fig. 2.5: IP- vs. Overlay-routing.....	15
Fig. 2.6: Boom-overlay voor media distributie in een P2P-netwerk.....	17
Fig. 2.7: Net-overlay voor media distributie in een P2P-netwerk.....	18
Fig. 2.8: Schets van een mogelijke combinatie van verschillende P2P-technieken.....	19
Fig. 3.1: Schets van interactie van traditionele bouwstenen in een hybride P2P-configuratie.....	23
Fig. 3.2: Aanmelden van een peer bij het P2P-netwerk door lid te worden van een peergroep....	25
Fig. 3.3: Metadata-advertentie van een peergroep.....	25
Fig. 3.4: Intern en extern zoeken in peergroepen.....	27
Fig. 3.5: Werking van een Herorderingsbuffer.....	28
Fig. 4.1: P2P-softwarearchitectuur.....	32

LIJST VAN TABELLEN

Tab. 2.1: Vergelijking van de P2P-generaties.....	14
---	----

LIJST VAN CODEFRAGMENTEN

Code 4.1: Implementatie van een zoekmechanisme voor peergroepen.....	37
Code 4.2: Implementatie van een functie voor het creëren van een nieuwe peergroep.....	38
Code 4.3: Implementatie van een functie voor het toetreden tot een peergroep.....	38
Code 4.4: Functie voor het delen van data in een peergroep.....	39
Code 7.1: SPARQL query naar de namen en e-mailadressen van alle personen in de wereld.....	51
Code 7.2: SPARQL query naar de hoofdsteden van alle landen in Afrika.....	51
Code 7.3: Voorbeeld RDF gebaseerde metadata-annotatie bij een videosegment.....	53

HOOFDSTUK 1

INLEIDING

1.1 PROBLEEMSTELLING

1.1.1 MULTIMEDIA OVERDRACHT

Sinds de globale penetratie van breedband internet is het realistischer geworden om grote bestanden van de ene kant van de wereld naar de andere te versluizen. Bijgevolg is het percentageel aandeel van multimediate bestanden in webinhoud exponentieel gestegen en nemen deze bestanden, ook door hun inherent omvangrijkere bestandsgrootte, een substantiële brok van de netwerkverkeer in beslag.

Unicast

De meest gangbare methode voor het stromen van multimedia op het huidige internet is, dit allermeest om historische redenen, nog steeds het gecentraliseerde client-server model.

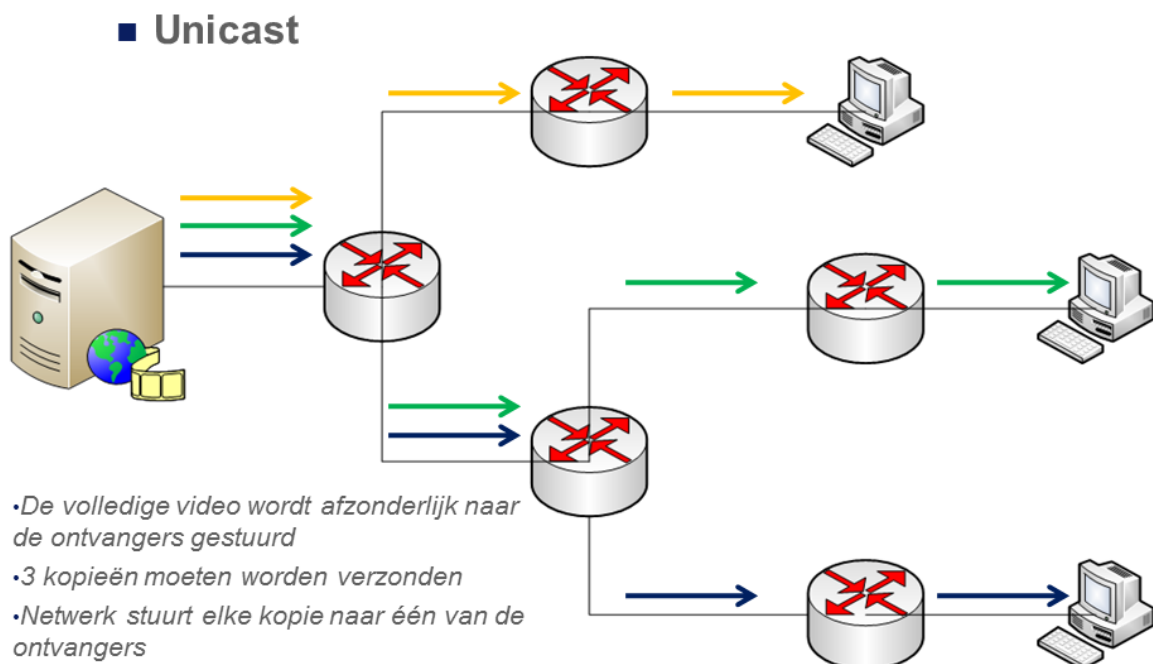


Fig. 1.1: Demonstratie van het unicast principe

Bij een client-server opstelling wordt een IP-unicast techniek gebruikt. De server zal met andere woorden elk van de verbonden clients afzonderlijk bedienen. Zoals voorgesteld in Fig. 1.1 zullen voor het voorzien van de x clients x kopieën van de multimedia moeten verzonden worden, wat een duidelijke redundantie aan netwerkverkeer met zich meebrengt.

In toepassing met grote datastromen blijkt dit dus een zeer inefficiënte aanpak in termen van CPU-gebruik, geheugengebruik en bandbreedte. Multimedia broadcasting over het internet heeft dus nood aan een kostoptimale transmissiemethode voor multimediate distributie.

Multicast

Sinds de jaren '80 zijn verschillende hard- en softwaretechnieken ontworpen voor het efficiënt verspreiden van data over een netwerk. Multicast (1) is een techniek waarbij een zender geen weet hoeft te hebben van het aantal ontvangers en hun locatie. Hierbij wordt de meest efficiënte strategie gehanteerd. Zoals hieronder in *Fig. 1.2* wordt gedemonstreerd, wordt de data slechts eenmaal over elke link in het netwerk gestuurd. Kopieën van de stroom worden slechts gemaakt op het moment dat de netwerkpaden naar verschillende bestemmingen opsplitsen.

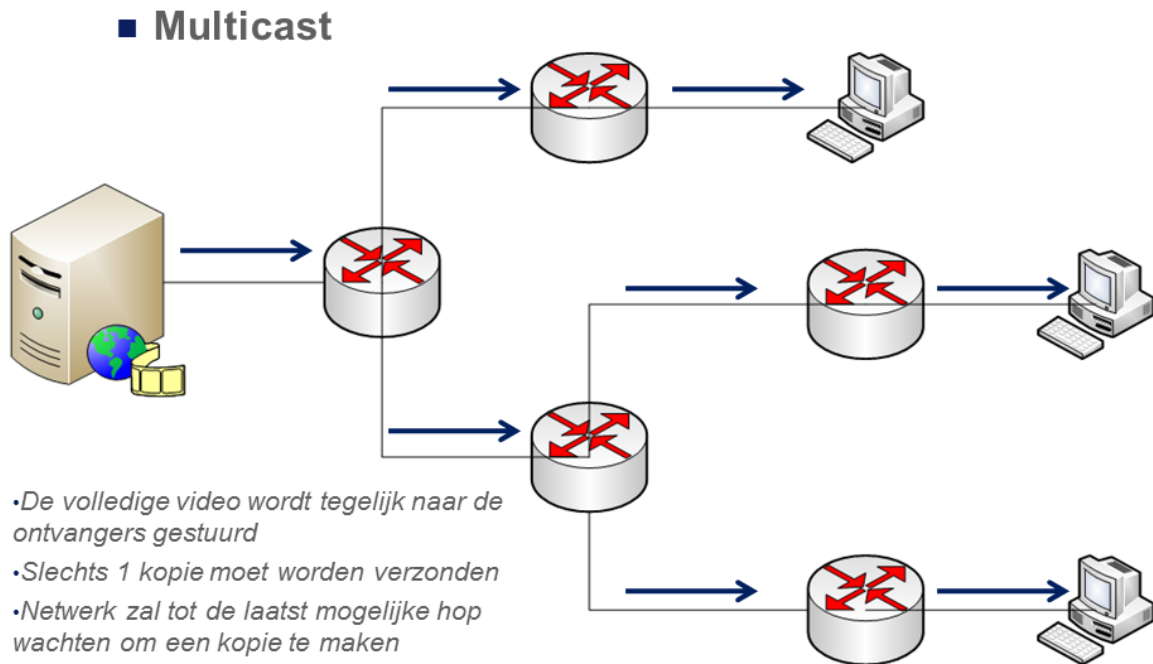


Fig. 1.2: Demonstratie van het multicast principe

De term "multicast" wordt typisch gebruikt in de context van IP-multicasting, een techniek die veelal wordt gebruikt voor het streamen van multimedia in gespecialiseerde toepassingen, internettelevisie en zelfs digitale televisie. Belgacom biedt sinds 2005 Internet Protocol TeleVision (IPTV) diensten aan onder de vorm van hun digitale televisie product: "Belgacom TV"¹. IPTV is een verzamelnaam voor diensten en toepassingen die tv-kijken over het Internet Protocol (IP) mogelijk maken. Hier wordt een breed scala aan verschillende technieken voor gebruikt zoals (de)compressiealgoritmes voor video en audio, multicast routingprotocollen, etc.

Het formaat van IP-multicast pakketten is identiek aan dat van unicast pakketten en onderscheidt zich slechts door het gebruik van een speciale klasse van bestemmingsadres, een klasse D IPv4-adres of in het geval van IPv6 waar een andere voorbehouden multicastklasse adressen is voorzien om een bepaalde multicastgroep aan te duiden. Het speciaal gereserveerde multicastadres, dat zich fysiek binnen een router bevindt, wordt zowel door zender als ontvanger gebruikt om data te verzenden/ontvangen. Multimediaservers stellen dit groepsadres als bestemmingsadres in hun datapakketten in, en zenden de data dus naar een multicastrouter (mrouter).

¹ <http://www.belgacomtv.be>

In tegenstelling tot bij een broadcast, die enkel kan uitgevoerd worden op een lokaal netwerk, zullen multicastclients enkel pakketten ontvangen als ze eerder hiervoor gekozen hebben door toe te treden tot de specifieke groep met een gegeven multicastadres. Lidmaatschap van een groep is dynamisch en wordt gecontroleerd door de ontvangers zelf. Clients kunnen, gebruikmakend van het Internet Group Management Protocol (IGMP) bij IPv4 en Multicast Listener Discovery (MLD) voor IPv6, via het groepsadres aangeven dat ze pakketten die gezonden worden naar deze groep willen ontvangen. Routers in een multicastnetwerk leren welke subnetwerken actieve klanten bedienen voor elke multicastgroep en proberen zo om het verzenden van pakketten te minimaliseren in delen van het netwerk waarin geen actieve clients zijn.

Multicast kan gebruikt worden om een groep van clients te bedienen die een gemeenschappelijke set van gegevens op hetzelfde moment vereisen, of wanneer clients in staat moeten zijn een set gemeenschappelijke data te ontvangen en op te slaan totdat ze nodig zijn (caching). Waar er een gemeenschappelijke behoefte is naar dezelfde data binnen een groep van clients kan multicast zorgen voor een significante besparing van bandbreedte (tot $1/N$ de van de bandbreedte in tegenstelling tot het sturen naar N verschillende unicastclients).

De meerderheid van de huidige Local Area Netwerken (LAN), zoals ethernet, ondersteunen multicast. De eerste LAN netwerkkaarten misten echter een specifieke integratie van het multicast principe en dit introduceerde een groot prestatieverlies. Doordat de adapter alle pakketten ontving (promiscuous mode) en achteraf softwarematig alle ongewenste pakketten uit de ontvangen data filterde, werd onnodig veel extra werk gegenereerd. De meeste moderne netwerkkaarten implementeren hardwarematige multicastfilters, die de host van het uitvoeren van het kostelijk softwarematig filteren weerhouden.

Om de IP-multicast techniek te kunnen toepassen moet het netwerk dus voorzien zijn van speciale hardware. Alle betrokken mroueters moeten de koppels van clientadressen en multicastgroepen kunnen bijhouden en hardwarematig verwerken. Door het extra werk dat het verwerken van de adressering en het ontdebellen van de data met zich meebrengt, is van een globale implementatie echter helemaal geen sprake. Teveel van de systeembronnen van een router zouden moeten besteed worden aan dit proces, systeembronnen die anders zouden kunnen geïnvesteerd worden in het bedienen van klassieke unicast routeringsopdrachten.

Initiatieven zoals MBone (2) proberen een alternatief multicastnetwerk op wereldwijde schaal te implementeren. In het geval van MBone werd al snel voor een virtuele laag bovenop het internet gekozen. Deze laag bestaat uit een combinatie van mroueters die IP-multicast ondersteunen en geen deel uitmaken van het conventionele internet. Tunnels tussen de mroueters zorgen ervoor dat multicastdata kan gestuurd worden over standaard routers in het internet.

Het is dus mogelijk een multicast systeem te emuleren op het bestaande internet Dit neemt niet weg dat IP-multicast nog steeds gebaseerd is op UDP-verkeer, aangezien TCP-verkeer enkel mogelijk is tussen een koppel eindpunten (end systems). Bij een punt-tot-punt protocol zoals TCP wordt de data in volgorde verzonden en pas verdergegaan met de overdracht als de aankomst van een pakket is bevestigd. UDP daarentegen is van nature uit een "best effort" protocol. Berichten worden gestuurd wanneer de verbinding de mogelijkheid hiertoe heeft en een aankomst van een pakket moet niet door de ontvanger bevestigd worden. Bijgevolg kunnen

de pakketten uit volgorde afgeleverd worden of zelfs helemaal verloren gaan. Bovendien wordt, zoals voorgesteld op *Fig. 1.2*, de data via een boomstructuur verzonden. Als gevolg hiervan zal bij het wegvallen van een tussenpunt (intermediate systems) of verbinding de data haar bestemming zeker niet meer zal bereiken en de verzenders hiervan niet op de hoogte worden gebracht.

1.1.2 METADATA EN MULTIMEDIA

Een tweede probleem dat dit proefschrift wil belichten is het verwerken van metadata die de te versturen multimedia beschrijft. De noodzaak voor het verwerken van complexe semantische annotaties bij multimedia kan gemotiveerd worden vanuit een nood aan de productie van gepersonaliseerde multimediapresentaties. Dit veronderstelt dat een groot aantal semantisch geannoteerde multimedia beschikbaar is, waarvan een subset kan gekozen worden uit het archief en de verschillende fragmenten kunnen worden samengevoegd tot één multimediapresentatie afgestemd op een bepaalde gebruiker.

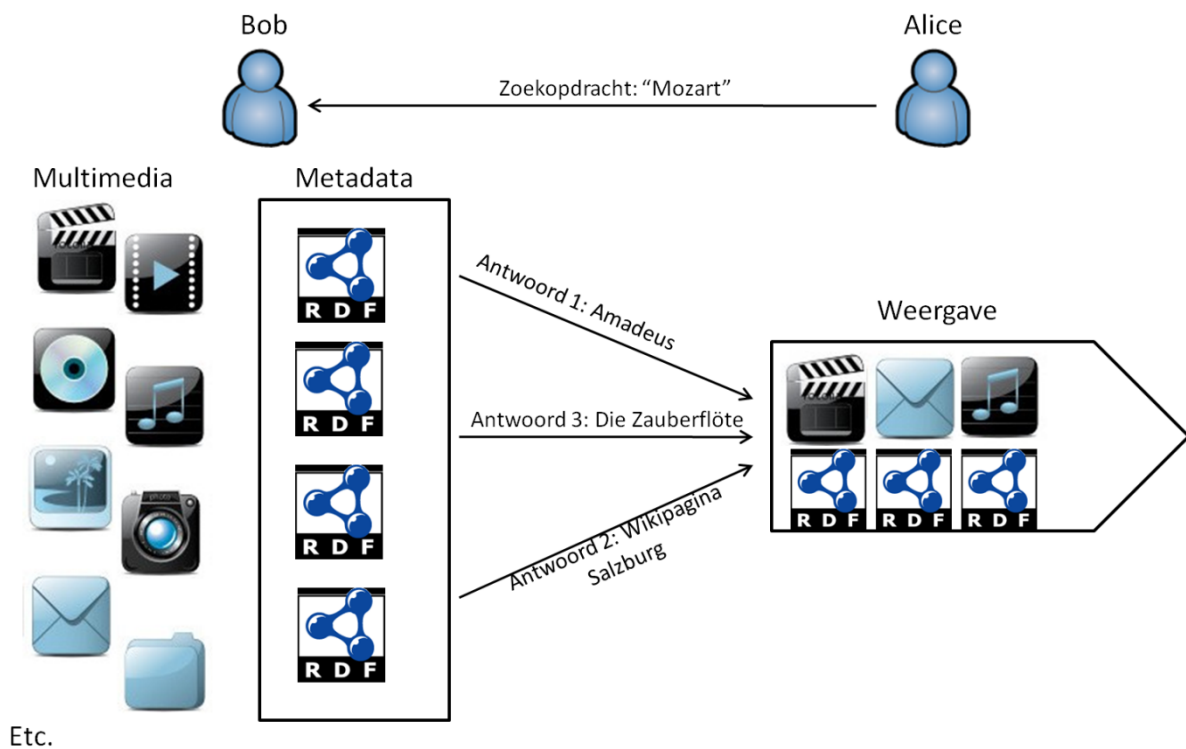


Fig. 1.3: Vereenvoudigde figuur over de samenstelling van een gepersonaliseerde multimediapresentatie a.d.h.v. metadata

Een metadata annotatie voor multimedia brengt over het algemeen enkele fundamentele problemen (3) met zich mee, deze problemen kunnen in drie categorieën ingedeeld worden. Allereerst moeten de relaties tussen de wereldse objecten, de bijhorende digitale multimediarepresentaties, en tussen hun beider metadata-annotaties expliciet worden vastgelegd. De relaties tussen de verschillende instanties worden gevisualiseerd in *Fig. 1.4*. In de tweede plaats vereisen multimediatoepassingen een onderscheid tussen inhoudsannotaties en een technische beschrijvingen van de multimedia. Een derde punt is dat multimediatoepassingen vaak metadata uit een verscheidenheid aan beschrijvende standaarden vereisen. Het combineren van deze in één enkel, coherent en flexibel annotatieschema is een niet-triviaal en vaak onderschat probleem.

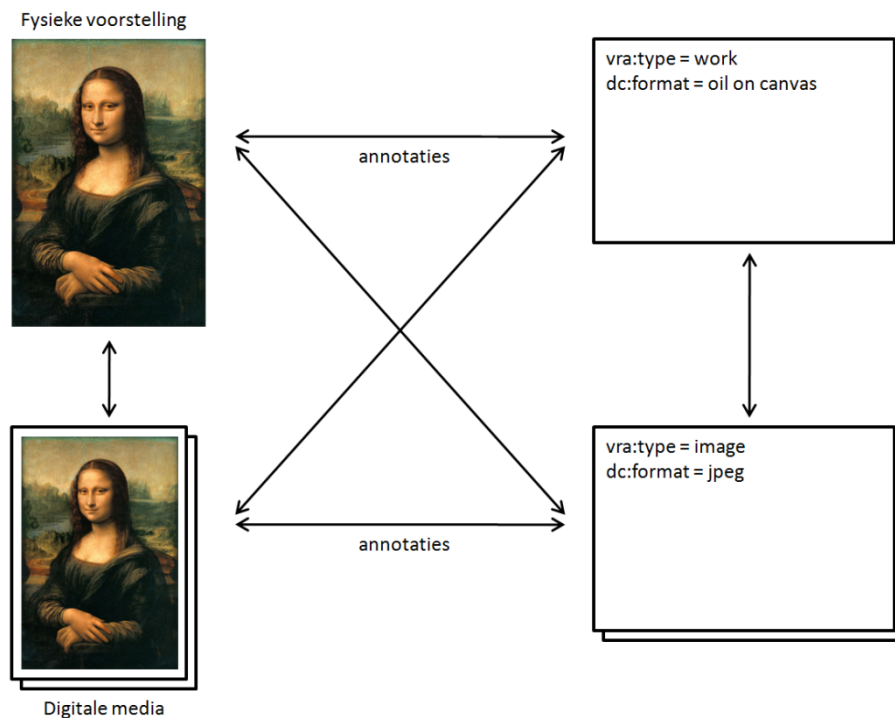


Fig. 1.4: Relatie tussen multimedia en annotaties

1.2 OVERZICHT

Na dit inleidend hoofdstuk volgt een voorstel voor het verhelpen van het eerst vermelde probleem, het bedenken van een kostoptimale multimediaoverdracht uitgaande van een peer-to-peer (P2P) aanpak. De basisfunctionaliteit van de gebruikte API, Sun JXTA², wordt voorgesteld en de implementatie van de voorgestelde oplossingen in een demotoepassing toegelicht.

In het tweede deel van de masterproef is het de bedoeling om de struikelblokken rond metadata bij multimedia aan te pakken. Allereerst wordt een selectie bestaande metadata-annotatiestandaarden voorgesteld waarmee het mogelijk is om tot op een bepaald niveau multimedia te voorzien van metadata. Daarna worden de tekortkomingen van deze metadatabibliotheken bij dit specifieke probleem belicht. Vanuit de problemen worden enkele eigenschappen vooruitgeschoven waaraan een metadataschema, rekening houdend met personalisatie van multimedia, zou moeten voldoen. Tenslotte wordt aan de hand van de bestaande standaarden een combinatie voorgesteld die alle vooruitgeschoven verwachtingen inlost. De keuze voor een metadata standaard, RDF³, wordt toegelicht en de implementatie ervan in de demoapplicatie overlopen.

1.2.1 DOEL

Dit onderzoek heeft tot doel enerzijds enkele struikelblokken bij het distribueren van multimediale data toe te lichten en anderzijds problemen bij het personaliseren van de te verspreiden media te analyseren. Voor de besproken problemen binnen deze twee domeinen

² <https://jxta.dev.java.net>

³ <http://www.w3.org/RDF>

worden enkele mogelijke oplossingen vooruitgeschoven. De meeste van de beschreven principes worden dan in een demoapplicatie geïmplementeerd om ze aan de realiteit te toetsen.

De toepassing is puur educatief en het is in geen geval de bedoeling een volledig operatief framework af te leveren. Op enkele optimalisaties na zullen alle beschreven oplossingen geïmplementeerd worden om de viabiliteit en de efficiëntie van de ideeën te demonstreren.

1.2.2 METHODE

Een literatuurstudie moet ervoor zorgen dat op de verschillende probleemdomeneinen een idee van de huidige ontwikkelingen gevormd kan worden. Door de bestaande technieken te evalueren en combineren, met eventueel nieuwe elementen, kan een nieuwer, vollediger antwoord op de probleemstelling geformuleerd worden. Zoals reeds vermeld is het de bedoeling de ideeën gepresenteerd in deze masterproef gedemonstreerd worden in een applicatie.

DEEL I

P2P-OVERDRACHT VOOR EEN MULTIMEDIA STREAMING PROTOCOL

HOOFDSTUK 2

P2P MULTIMEDIA STREAMING PROTOCOL: STATE OF THE ART

Het eerste probleem dat wordt besproken in dit werk is het optimaliseren van de gegevensoverdracht voor de multimediatistributie. In het inleidend hoofdstuk werden al enkele problemen met de veelgebruikte uni-/multicast technieken naar voor geschoven. Historisch gezien blijkt het overschakelen van de overdrachtstechniek van een centrale server oplossing naar een gedecentraliseerde peer-to-peer (P2P) architectuur een eerste logische stap.

In dit hoofdstuk bespreken we P2P-systemen meer in detail. Deze netwerken worden gekenmerkt door de symmetrische aard van de interactie tussen processen, dit in tegenstelling tot bij de client-server systemen. Bij de laatstgenoemde configuratie is er een duidelijk onderscheid tussen de relaties: een bepaald proces is ofwel strikt verantwoordelijk voor de afgifte van verzoeken ofwel het afhandelen van antwoorden op deze verzoeken. Een standaard voorbeeld van een client-server toepassing is de interactie tussen webserver (server) en een webbrowser (client).

In P2P-netwerken daarentegen zorgen dezelfde componenten voor zowel de aflevering van verzoeken als voor het verwerken ervan. Een typisch voorbeeld is natuurlijk P2P-bestandsdeling, waarbij een proces verzoekt een digitaal object op te halen bij zijn peers, maar op hetzelfde moment leverancier kan zijn van dergelijke objecten voor andere knopen.

2.1 WAAROM P2P?

Een belangrijke reden voor het verkiezen van P2P-systemen boven een zuivere client-server architectuur, is de beperkte schaalbaarheid en de daarbij horende oplopende kosten van de servergeoriënteerde architecturen. Naarmate het aantal clients groeit, naargelang de dienstverlening dus populairder wordt, moet de infrastructuur sterk worden uitgebreid. Dit kan de hardware- en operationele kosten sterk laten oplopen. Bovendien verloopt de kosttoename niet volgens een lineaire schaal met het aantal gebruikers. Elke technologie en systeem is slechts optimaal voor een welbepaald aantal gebruikers. Als deze grens wordt overschreden, moet worden gemigreerd naar andere hard- en software, wat leidt tot hogere kosten.

Volgens de P2P-filosofie schenkt elke gebruiker een deel van zijn systeemmiddelen aan het netwerk. Hierdoor groeien de voorzieningen van het netwerk evenredig met het aantal gebruikers en wordt met dit principe het belangrijkste probleem van client-server systemen omzeild. Desalniettemin introduceert deze benadering op haar beurt nieuwe complexiteit. Software systemen moeten in dit geval immers kunnen werken zonder gecentraliseerde controle, wat de invoering van intelligente algoritmes en complexere softwarearchitecturen vereist.

Het leveren van een schaalbare high-end infrastructuur aan een lage kost heeft dus wel degelijk een prijs. Allereerst blijven knopen onder controle van de eindgebruikers, ze kunnen met andere woorden worden uitgeschakeld zonder de voorafgaande kennisgeving aan het systeem. Het P2P-netwerk zal dus bestaan uit knopen die in principe instabiel zijn, en dat niet volledig onder controle van het netwerk zelf. Bovendien staat er geen limiet op het aantal participerende knopen in een netwerk, wat opnieuw bijdraagt tot de complexiteit van het systeem. Ten tweede is het netwerk dat wordt gebruikt om twee randknopen, knopen die geen deel uitmaken van eenzelfde lokaal netwerk (Local Area Network of LAN), via het internet te verbinden in handen

van een derde partij. Deze globale netwerken (Wide Area Network of WAN) zijn vaak traag en onbetrouwbaar. Ze hebben meer last van uitvallende verbindingen en bandbreedteschommelingen dan lokale netwerken. Een derde struikelblok is dat door een gebrek aan een gecentraliseerde infrastructuur ook gedistribueerde algoritmes nodig zijn voor alle management functies. Deze algoritmes zijn niet alleen inherent complexer dan hun gecentraliseerde tegenhangers maar er bestaat ook een reëel gevaar dat de infrastructuur waarop de kritische managementfuncties worden uitgevoerd faalt.

We kunnen dus stellen dat P2P-netwerken worden gekenmerkt door gedecentraliseerde controle. Typisch zullen ze een bepaalde vorm van zelforganisatie nodig hebben om het falen van knopen op te vangen en dynamisch de werklast te verdelen afhankelijk van de verzoekpatronen. Alle knopen in het netwerk hebben in principe identieke verantwoordelijkheden, toch zijn P2P-systemen doorgaans gevoelig voor misbruik van bepaalde gebruikers die het systeem gebruiken zonder zelf iets bij te dragen.

2.2 EVOLUTIE

2.2.1 1^{STE} GENERATIE P2P-APPLICATIES

P2P-systemen werden in de eerste plaats populair in de context van bestandsuitwisseling. Om het zoeken en indexeren eenvoudiger te kunnen implementeren, werd bij de eerste generatie systemen de controle gecentraliseerd. Het downloaden van digitale inhoud gebeurde zonder inmenging van deze centrale controle-infrastructuur, eens de gewenste objecten waren gelokaliseerd. Hoewel dit een aantrekkelijke aanpak is door de beperkte complexiteit, een gecentraliseerde controle is altijd makkelijker te realiseren dan gedistribueerde alternatieven, vertonen deze systemen aanzienlijke nadelen. De schaalbaarheid blijkt ondermaats en het systeem gaat neer als een enkele verbinding of de centrale controleknoop faalt. Voorbeelden van P2P-applicaties die tot deze categorie behoren zijn ICQ en Napster.

ICQ

ICQ (4) is een van de eerste P2P applicaties voor het wijde publiek en de moeder van alle instant messaging (IM) programma's. Het werd oorspronkelijk ontworpen om tekstberichten naar andere online gebruikers te verzenden.

Lokalisatie van een gebruiker op het internet is niet altijd even eenduidig. Veel mensen verbinden van op verschillende geografische locaties met het internet, en zelfs als je enkel van thuis uit werkt zal je vaak een verschillend IP-adres krijgen.

Het is eenvoudiger voor ICQ geen gebruik te maken van een pure P2P-architectuur. Voor de identificatie van gebruikers wordt gekozen voor een centrale server. Deze zal ook de online status aan de andere gebruikers melden. Verdere communicatie gebeurt via een directe verbinding. De architectuur is dus een combinatie van een klassieke client-server en een P2P-architectuur.

Napster

Napster (5) (6), dat drie jaar na ICQ op het internet verscheen, is een bestandsdelingsapplicatie die een compleet nieuwe aanpak voor mediaopslag introduceert. Voor het eerst is het mogelijk om bestanden op de lokale harde schijf van gebruikers te delen, dit in tegenstelling tot de klassieke client-server architectuur waarbij alle data van de server wordt verzonden.

Bij het aanmelden zal een gebruiker een lijst van zijn te delen bestanden doorsturen naar de centrale server. Wanneer een andere gebruiker een bestand wil zoeken op het netwerk, zal hij een zoekopdracht naar deze server sturen. De server antwoordt op zijn beurt met een lijst van IP-adressen waarop een bestand kan worden gevonden die aan de parameters voldoet. Op dit moment kan een rechtstreekse verbinding tussen de peers zorgen voor de informatieoverdracht. Hier ligt net het succes van Napster, moesten alle bestanden op één centrale server zijn opgeslaan zouden noch deze server, noch het netwerk alle transacties kunnen verwerkt hebben.

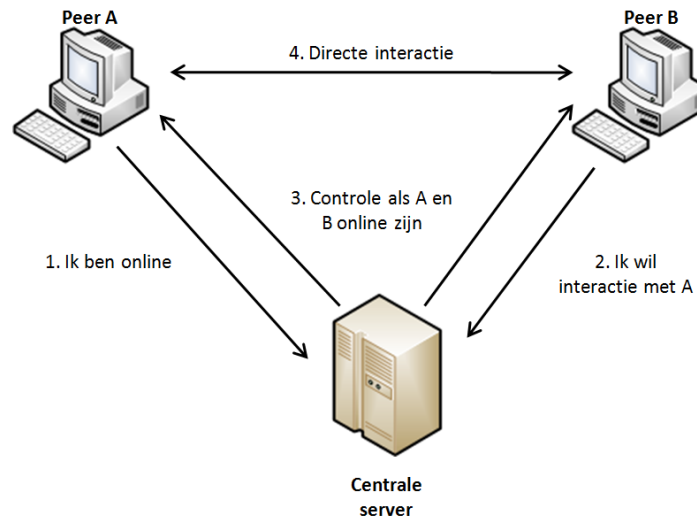


Fig. 2.1: Algemeen schema voor 1^{ste} generatie P2P-applicaties.

2.2.2 2^{DE} GENERATIE P2P-APPLICATIES

De tweede generatie P2P-systemen zijn nog steeds gericht op het delen van bestanden, maar streven naar een volledig gedecentraliseerde controle. Een overlay-netwerk⁴ moet de connectiviteit garanderen binnen het netwerk, zelfs in aanwezigheid van wegvallende verbindingen en knopen.

Twee varianten van deze gedecentraliseerde aanpak bestaan. Enerzijds is er een zuivere P2P-architectuur waarbij alle knopen identiek zijn en dezelfde verantwoordelijkheden hebben. Ten tweede zijn er hybride systemen, waarbij een aantal bevoorrechte knopen meer logica bevatten dan anderen. Hybride P2P-systemen werden ontwikkeld om de schaalbaarheidsproblemen bij zuivere P2P-systemen te adresseren. Wanneer er media wordt gezocht in een netwerk gemodelleerd als complete graaf (flat network), zullen de zoekopdrachten worden verstuurd naar alle andere knopen in het netwerk. Deze stroom van verzoeken veroorzaakt een snelle toename van het controleverkeer. De verzoeken stijgen exponentieel bij toename van het aantal burens. In hybride systemen neemt een “superknoop” (super node) de verantwoordelijkheid voor een groep van knopen op zich. Onder de verantwoordelijkheden valt het bijhouden van een centrale index van de inhoud van alle knopen die behoren tot de groep. Zoekopdrachten worden eerst door een dergelijke superknoop verwerkt. Slechts als deze de zoekopdracht niet kan beantwoorden, zal de opdracht worden doorgestuurd naar andere superknopen.

⁴ Uiteenzetting volgt in 2.4 Overlays

Gnutella

De Gnutella-toepassing (7) (8) probeert een connectie te maken met een voorgedefinieerde set van knopen om een lijst van IP-adressen van verbonden peers te bekomen. Daarna zal het programma proberen te verbinden met de peers in de lijst om op deze wijze meer andere knopen te ontdekken. Dit proces zal zich herhalen tot de applicatie met voldoende knopen verbonden is. Door op deze wijze te verbinden zal de applicatie volledig gedecentraliseerd en onafhankelijk van de netwerk topografie blijven. Zolang er initiële opstartknopen (seed nodes) voorhanden zijn, kunnen peers tot het netwerk toetreden of het verlaten.

Wanneer de applicatie met het netwerk is verbonden, kan een zoekopdracht worden gestuurd naar een aantal verbonden knopen. Deze zullen op hun beurt de zoekopdracht doorsturen naar hun burens, peers waarmee de knoop is verbonden, en zo verder. Via dit domino-effect kan een bestand gelokaliseerd worden. Eens een bepaalde peer een bestand heeft dat aan de zoekopdracht voldoet zal hij de originele afzender contacteren om de bestandsoverdracht tussen hen op te zetten.

Om te voorkomen dat een zoekopdracht in een eindeloze lus door het netwerk propageert, krijgt elk bericht een time to live (TTL). Deze wordt verminderd elke keer een zoekopdracht een knoop doorkruist. Wanneer de TTL 0 aanduidt, zal het bericht niet meer worden verzonden. Naast een TTL houdt elke kopie van de gepropageerde zoekopdracht een lijst van bezochte peers bij, zodat het bericht niet tweemaal dezelfde knoop passeert. Dit kan echter niet voorkomen dat eenzelfde knoop langs verschillende paden wordt bevraagd, wat hoogstwaarschijnlijk is als de knoop een buur is van meerdere andere knopen.

Zoals reeds vermeld is het probleem met deze architectuur dat naar mate het aantal gebruikers toeneemt de hoeveelheid controleverkeer tussen de knopen het netwerk verstopt. Bijgevolg is de netwerkcongestie die in een client-server architectuur zich rond de servers concentreert op een gelijkaardige wijze nu aanwezig rond de peers. Een groot voordeel tegenover zijn voorganger, Napster, is wel dat door de decentralisatie het netwerk minder gevoelig wordt aan aanvallen en minder gemakkelijk kan worden neergehaald.

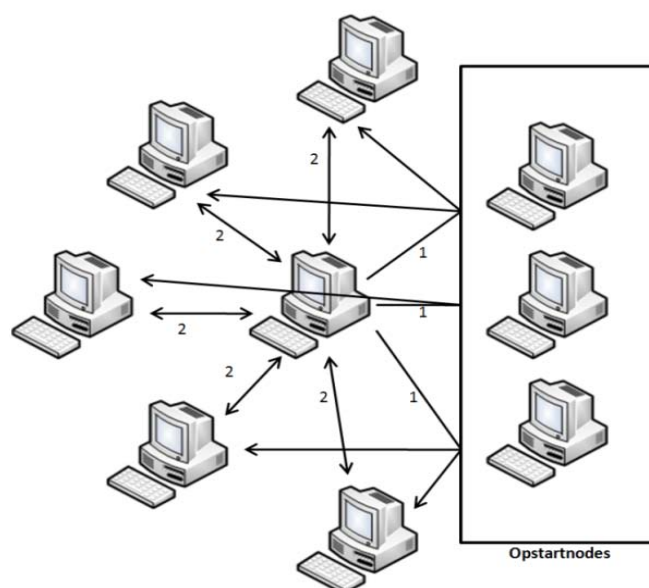


Fig. 2.2: Verbinden van een knoop met een Gnutella-netwerk.

Kazaa

Kazaa (9) (10) introduceerde in 2000 het concept van “Super Nodes” in zijn FastTrack protocol. Volgens deze specificatie was de peer zelf niet langer verantwoordelijk voor het onderhouden van een eigen bestandscatalogoog. Elke knoop zou nu periodiek een update van deze lijst uploaden naar zijn superknoop.

Zoekopdrachten worden bijgevolg enkel op deze superknoepen uitgevoerd, waarna deze antwoorden met een lijst van peers die het gezochte bestand aanbieden. De zoekopdrachten worden zo niet langer in een sterpatroon gepropageerd. Nadat het bestand is gevonden kan een gebruiker opnieuw via een rechtstreekse verbinding downloaden van de knoop die het bestand aanbiedt. Het FastTrack protocol introduceert zo beduidend minder netwerkverkeer dan de Gnutella gebaseerde toepassingen.

Superknoepen kunnen automatisch worden gekozen door de applicatie, dit op basis van hun beschikbare systeembronnen (opslag, bandbreedte, etc.) of een gebruiker kan ervoor kiezen een superknoop te worden. De collectie van superknoepen kan beschouwd worden als de implementatie van een gedistribueerde server.

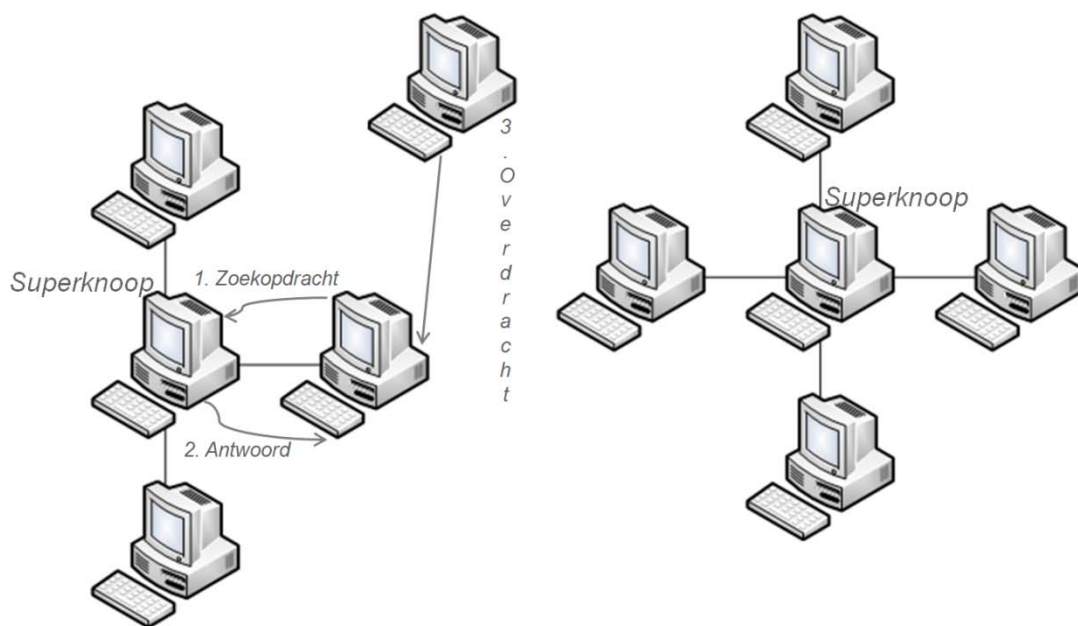


Fig. 2.3: Zoekopdracht en bestandsoverdracht binnen een FastTrack-netwerk.

2.2.3 3DE GENERATIE P2P-APPLICATIES

P2P-systemen zijn geëvolueerd van eerste generatie systemen met een gecentraliseerde controlestructuur (bv. Napster, SETI @ home, ICQ, etc.) tot een tweede generatie waarbij de knopen op basis van complete connectiviteit rechtstreeks met elkaar communiceren (bv. Gnutella, etc.). Sommige werken definiëren de classificatie van een derde generatie als een hybride structuur van zowel de eerste- als tweede generatie netwerken (bv. Kazaa, Grokster, Gnutella2, etc.). Aangezien deze ontwikkeling echter geen fundamentele revolutie op technologisch vlak betekent, volg ik liever Ian Foster (11) en karakteriseer derde generatie P2P-

netwerken als systemen die gebaseerd zijn op gedistribueerde hashtabellen (bv. BitTorrent⁵, Pastry, Freenet, etc.).

Distributed Hash Tables (DHT) zijn een klasse van gedecentraliseerde gedistribueerde systemen die een lookup service vergelijkbaar met een standaard hashtable aanbieden. De (sleutelwaarde)-paren die het systeem bijhoudt zijn in de DHT verspreid opgeslagen op verschillende knopen in het netwerk. Alle deelnemende peers kunnen op een efficiënte manier de waarde verbonden aan een bepaalde sleutel ophalen. De verantwoordelijkheid voor het handhaven van de sleutels-op-waarden bijectie is verdeeld onder de knopen op een zodanige wijze (overlappende redundantie, etc.) dat een verandering in de netwerkconfiguratie slechts een minimale invloed op de werking betekent. Dit maakt dat DHTs schaalbaar zijn voor zeer grote netwerken en zich perfect kunnen aanpassen aan de vluchtige aard van een P2P-configuratie.

BitTorrent

De grootste evolutie die BitTorrent (12) (13) introduceerde, was op het vlak van bestandsoverdracht. Waar vroeger één enkele peer instond voor de overdracht van een bestand, zal de gebruiker nu porties van het bestand simultaan van verschillende knopen kunnen verkrijgen om zo een “stroom” of “torrent” van data te creëren. Aangezien dit een heel efficiënte aanpak bleek, hebben vele protocols zoals FastTrack dit in latere versie van hun protocolbeschrijving opgenomen.

Via de stroom kunnen bestanden sneller gedownload worden dan bij de klassieke P2P-systemen. Bij 1^{ste} en 2^{de} generatie toepassingen kon je het ongeluk hebben om van een peer met een kleine bandbreedte te downloaden. De enige oplossing om een snellere download te bekomen was dan de verbinding te verbreken in de hoop een snellere connectie met een andere knoop te bekomen. BitTorrent kan de afzonderlijke upstreambandbreedtes van een aantal peers combineren om een snellere bestandsoverdracht te bekomen.

Om gelijktijdig van verschillende locaties te kunnen downloaden wordt de te verspreiden media in stukken (chunks) van een bepaalde grootte verdeeld. Elk stukje wordt van de nodige metadata voorzien zodat het op een niet -sequentiële wijze afgehaald kan worden, en achteraf het originele bestand te reconstrueren is. Zo kan een peer in willekeurige volgorde de voorhanden stukjes downloaden, ongeacht waar ze in het oorspronkelijke bestand gealloceerd zijn.

Om de communicatie tussen peers via het BitTorrent-protocol te verzorgen wordt beroep gedaan op een centrale server die een BitTorrent tracker wordt genoemd . Bij afwezigheid van uitbreidingen van het oorspronkelijke protocol hieronder beschreven, is dit het enige grote kritieke punt in de BitTorrent-configuratie. Peers die een download trachten aan te vangen moeten immers met de tracker communiceren. Tijdens het downloaden is er ook sprake van periodieke interactie met de tracker om te onderhandelen met nieuwe peers en statistieken uit te wisselen. Na de eerste ontvangst van peergegevens kan de P2P-communicatie wel strikt gezien verdergaan zonder enige tussenkomst van een tracker.

Later boden enkele torrenttoepassingen ondersteuning aan voor het downloaden van torrents die geen actieve tracker meer hadden. Dit werd mogelijk gemaakt door een DHT implementatie onder de vorm van een gedistribueerde databank onder te brengen in elke knoop, waarin

⁵ Bij gebruik van een gedistribueerde tracker

chunks en hun verspreiding kunnen worden opgezocht. De ontwikkelaars van BitTorrent zagen meteen het potentieel hier van in en introduceerden de ontwikkeling in de officiële specificatie van hun protocol. Sinds op deze wijze het belangrijkste probleem met het trackersysteem omzeild werd, kan BitTorrent dus gezien worden als een 3^{de} generatie P2P-systeem.

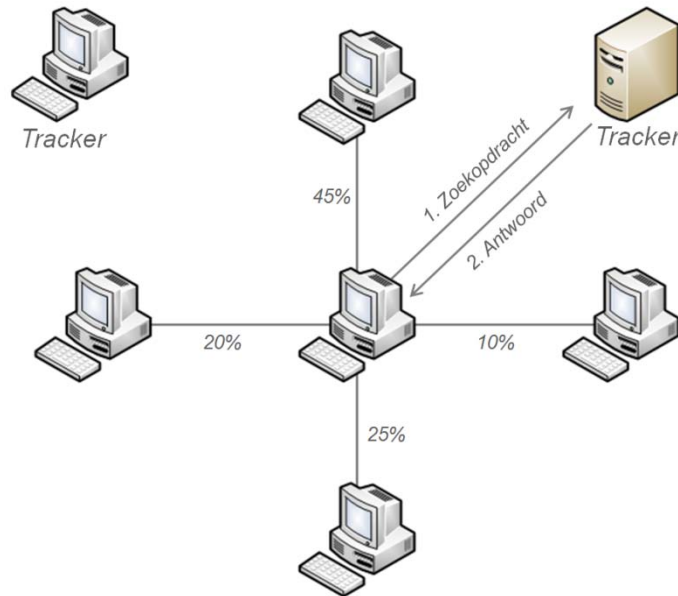


Fig. 2.4: Interactie binnen een BitTorrent opstelling.

	Gemedieerde P2P 1 ^{ste} Generatie P2P	Zuivere P2P	Hybride P2P 2 ^{de} Generatie P2P	DHT-P2P 3 ^{de} Generatie P2P
Dataverkeer	P2P	P2P	P2P	P2P
Controleverkeer	Client-server	P2P	Client-server + P2P	P2P
Efficiëntie	Efficiënt zoeken Efficiënte controle	Inefficiënt zoeken Bandbreedte consumerende controle	Efficiënt zoeken Relatief efficiënte controle	Efficiënt zoeken Efficiënte controle
Schaalbaarheid	Load balancing en redundantie nodig voor indexservers	Nodige bandbreedte groeit exponentieel door controle	OK	OK
Robuustheid	Wegvallen van 1 verbinding of de server haalt het systeem neer	Verbindingen of knopen mogen wegvallen dit vertraagt enkel het systeem	zie zuivere P2P	Wegvallen van een verbinding of knoop heeft geen gevolgen voor het systeem
Verantwoordelijkheid	Eenvoudig	Eenvoudig	Ingewikkeld	Ingewikkeld

Tab. 2.1: Vergelijking van de P2P-generaties.

2.4 OVERLAYS

Het principe van zelforganisatie betekent dat er niet langer een centrale coördinerende node is die alle activiteiten stuurt. Bijgevolg wordt er ook geen centrale database van de globale situatie in het systeem bijgehouden. Het globale gedrag van het netwerk is het rechtstreeks gevolg van alle beslissingen die lokaal in de peers genomen worden. Bij gebrek aan een uitgesproken centrale controle zou elke knoop in een P2P-systeem idealiter elke andere of tenminste zoveel

mogelijk andere knopen moeten kennen. Zoals reeds vermeld blijkt deze aanpak echter niet schaalbaar naar grote systemen omdat elke peer in dit geval een lijst moet onderhouden van alle andere peers in het netwerk, wat resulteert in een overvloed aan controleverkeer. Daarom is de kennis van elke knoop beperkt tot een aantal andere peers, waarnaar vanaf nu om praktische redenen zal gerefereerd worden als burens. Om alle systeemmiddelen in het P2P-netwerk te kunnen bereiken, zullen aanvragen die niet kunnen worden behandeld door dergelijke subgroep worden doorgestuurd naar de burens van de burens. Deze benadering is analoog aan de filosofie van routing in IP-netwerken, waar in feite dezelfde schaalbaarheidsproblemen ontstaan naarmate het aantal IP-knopen toeneemt. Routeringsverzoeken worden echter gedaan op het applicatieniveau, in termen van het OSI netwerk referentiemodel, en het bekomen netwerk wordt een "overlay"-netwerk genoemd. De pakketten mogen dan wel logisch omgeleid worden via de overlay, ze blijven uiteindelijk getransporteerd worden over IP-verbindingen.

Overlay routing (14) (15) wordt mogelijk gemaakt door het toevoegen van een overlayheader aan de inhoud van het IP-pakket. Deze overlayheader bestaat uit een unieke identificatie van de bron- en bestemmingsknopen in de overlay. Het is belangrijk op te merken dat de beide lagen, IP en overlay, specifieke regels voor adressering en eigen routing protocollen hebben. Sinds overlayknopen zich anders gedragen dan IP-knopen, zullen de routingprotocollen die worden gebruikt in de overlay uiteraard niet optimaal zijn volgens de routing op IP-niveau en visa versa. Telkens een IP-pakket aankomt in een overlayknoop, zal deze knoop ofwel het pakket zelf afhandelen als de knoop de eindbestemming van het pakket is, ofwel het pakket doorzenden naar de volgende overlayknoop door middel van een eigen routingstabel op overlay niveau zoals voorgesteld op Fig. 2.5.

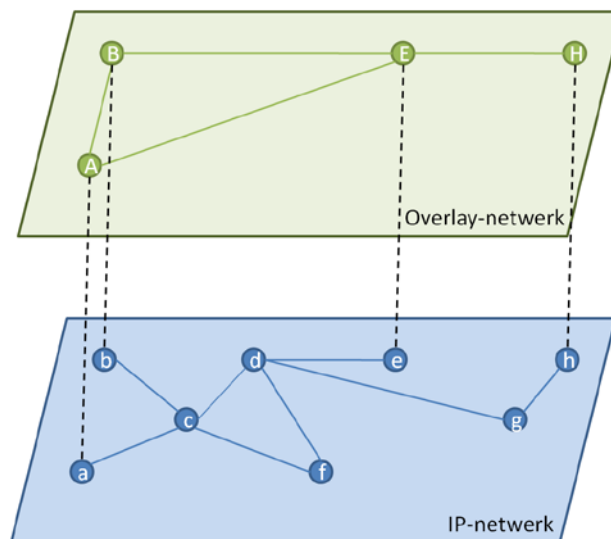


Fig. 2.5: IP- vs. Overlay-routing.

Bij inhoudgebaseerde routing krijgt elk digitaal object een wereldwijd unieke identificatie (Global Unique Identifier of GUID) die integraal wordt gebruikt in het routeringsproces. Deze GUIDs bestaan meestal uit een hashwaarde van het digitale object. Dit kan een hashvoorstelling zijn van de inhoud zelf (bv. de eigenlijke tekst van een tekstuele digitale object) of van de bijhorende metadata van het object (bv. titel, auteur, versie, ...). Vanwege deze hashingbenadering, zullen overlays die gebruik maken hashwaarden om verzoeken te sturen vaak aangeduid als "Distributed Hash Tables" (DHTs).

Twee types overlay-architecturen voor het structureren van een P2P-netwerk worden over het algemeen vooruitgeschoven voor het verdelen van multimedia. Aan de ene kant zijn er de boomgebaseerde structuren die downloadsessies van mediabronnen naar een aantal clientpeers voorzien. Aan de andere kant heb je netwerkstructuren die een uitgebreide parallelle gegevensdistributie onder verschillende peers onderling ondersteunt.

2.4.1 ENKELVOUDIGE BOOM

Bij een enkelvoudige boomstructuur (Single Tree Overlay) worden de peers in een boomstructuur geconfigureerd die de bron met de media gegevens en de clients verbindt. Clients zijn bladeren van de distributieboom, terwijl tussenknoten de digitale multimedia van de bron richting de bladeren duwen. Een peer kan tegelijk een tussenknoop zijn in de ene distributieboom en een blad in de andere. Enkelvoudige boomarchitecturen zijn gemakkelijk te implementeren en onderhouden. Er zijn echter ook limiterende factoren. Allereerst is dit een zeer onstabiele architectuur door de veranderlijke aard van het systeem, het hoge tempo waarop peers tot het netwerk toetreden en eruit vertrekken. Wanneer een tussenknoop het systeem verlaat is de boomstructuur onderbroken en zal de media de eindbestemming niet kunnen bereiken. Verder is de kwaliteit van de mediastroom bij deze architectuur beperkt tot de minimum uploadbandbreedte van een tussenknoop in de tak, aangezien elke client verbonden is via een afzonderlijke tak.

2.4.2 MEERVOUDIGE BOOM

Een meervoudige boomstructuur (Multiple Tree Overlay) is ontworpen om de connectiviteitsproblemen bij een enkelvoudige structuur uit de weg te gaan, dit door redundantie in de netwerkpaden in te bouwen. Een peer zal de media dus niet langer via een enkel pad afhaken. Een meervoudige boomstructuur kan dus gezien worden als een verzameling van enkelvoudige distributiebomen waarvan de bron en de eindpunten gelijk zijn, maar die verschillende tussenknoten gebruiken om de multimedia van bron tot eindbestemming te sturen. Wanneer een van de tussenknoten wegvalt, kan nog altijd via de andere paden gestroomd worden. Bovendien kunnen de bandbreedtes van de verschillende paden geaggregeerd worden, om downloadkwaliteit te bevorderen.

Dergelijke systemen zijn niet alleen moeilijker te ontwerpen, maar ook complexer in onderhoud. De verschillende paden mogen uiteraard in geen geval overlappen, anders zou de ingebouwde redundantie geen effect hebben bij het falen van een gezamenlijke node. Dit probleem introduceert een nieuwe paradox. Het is de bedoeling dat de diepte van de boom geminimaliseerd wordt en tegelijk een uitbreiding van de paddiversiteit bekomen wordt. Ook met de onderliggende fysieke topologie moet rekening gehouden worden om een optimale spreiding van de gegevens te bekomen. Verschillende verbindingen hebben verschillende bandbreedtes en de ene link betrouwbaarder als de andere.

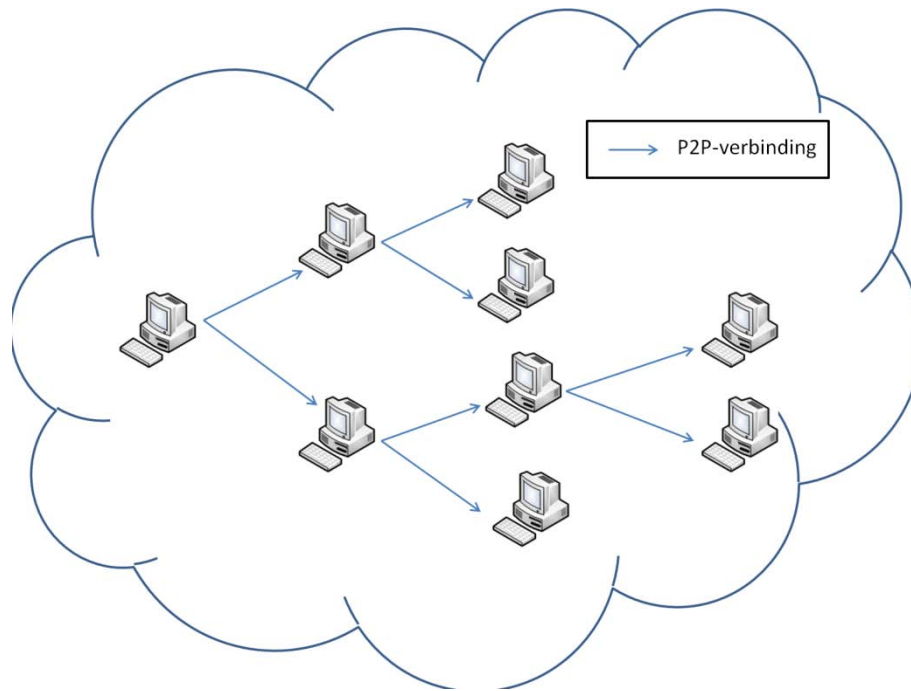


Fig. 2.6: Boom-overlay voor media distributie in een P2P-netwerk.

2.4.3 NET

Een netoverlay is een architectuur waarbij een knoop bereikbaar is langs ten minste twee paden en de peers georganiseerd zijn in een netachtige structuur (mesh overlay). Wanneer elke knoop verbonden is met elke andere in het netwerk spreekt men van een volledige netoverlay (fully meshed).

In het geval waar slechts enkele knopen rechtstreeks met elkaar verbonden zijn is het dus nodig dat sommige peers functioneren als tussenknoop, een switch als het ware, bij de communicatie. Als dergelijke knoop wegvalt, zullen andere communicatiekanalen verbroken worden. Hoe groter de connectiviteit van een peer, des te stabiel het netwerk dus is. Bij een volledige netoverlay kan een willekeurige knoop het netwerk verlaten zonder enig gevolg voor de interknoop verbindingen.

De originele mediastroom die bij de boomoverlay van één bron werd verstuurd, kan nu vertrekken van verschillende bronpeers. Om een optimale overdrachtsnelheid te bekomen kunnen van de verschillende bronnen een afzonderlijk deel van het bestand afgehaald worden. Indien dit langs strikt gescheiden netwerkpaden gebeurt, kan de downloadsnelheid oplopen tot de som van de capaciteit van de afzonderlijke paden en de snelheid met andere woorden ver boven de maximale bandbreedte van één enkel pad opgetrokken worden.

Een peer is verbonden met het netwerk via een of meerdere ouderknoten(s), waarvan hij informatie verkrijgt, en eventueel een aantal kinderknoten waaraan de mediapakketten worden doorgegeven. De voordelen van dergelijke architectuur liggen bij de lage kost en het eenvoudig onderhoud van de structuur. Bovendien is het netwerk meer gewapend tegen herconfiguratie, het vertrek of een fout in een knoop, door de ingebouwde redundantie van verbindingen die te wijten is aan een hogere diversiteit aan mogelijke netwerkpaden.

Het stromen van multimedia over deze architectuur brengt op zijn beurt echter nieuwe problemen met zich mee. Multimedia is inherent sequentieel geëncodeerd en wordt ook op deze

manier afgespeeld. Pakketdistributie en data aanvragen moeten ook op deze temporele ordening voorzien zijn. Doordat de gegevens weinig op voorhand beschikbaar zijn, zeker bij het stromen van de multimedia, limiteert dit de flexibiliteit in termen van doorvoer van gegevens in de architectuur. Deze restrictie kan gemakkelijk worden opgevangen door bijvoorbeeld afspeelbuffers in te bouwen en bij de pakketselectie stukken aan het begin van de weer te geven data te prefereren.

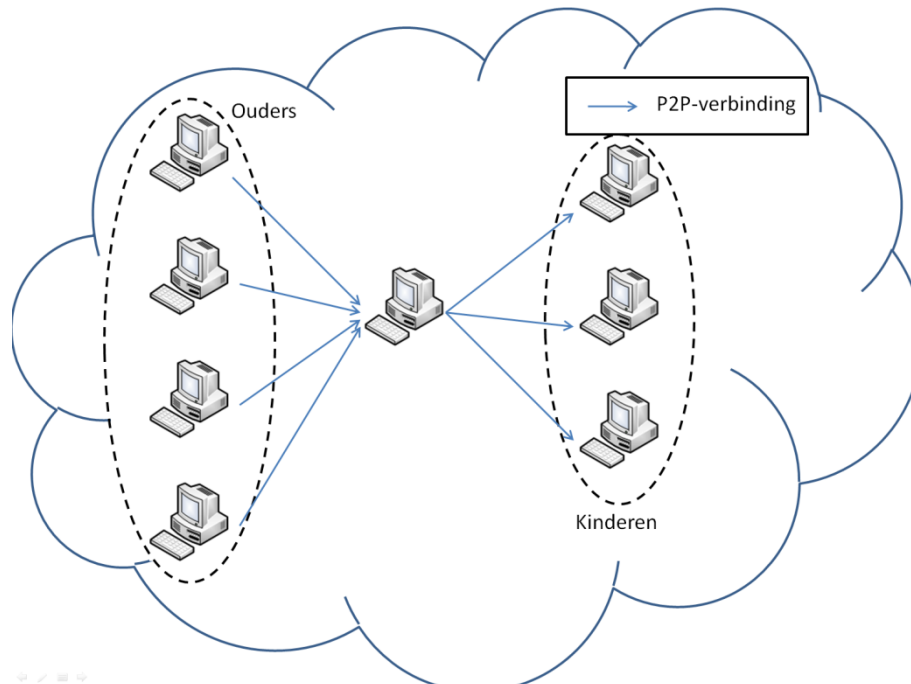


Fig. 2.7: Net-overlay voor media distributie in een P2P-netwerk.

2.5 CONCLUSIE

Een enkelvoudige boomstructuur is een architectuur die duidelijk niet robuust genoeg is. Ze vertoont veel gelijkenissen met een multicast opstelling, maar door de inherent veranderlijke aard van een P2P-netwerk is een fout in één enkele knoop of verbinding voor deze configuratie fataal. Hoewel een meervoudige boom dit probleem oplost, blijkt deze configuratie onnodig complex en is de performantie van dergelijk netwerk bovendien sterk afhankelijk van de onderliggende topologie van het netwerk. Netgestructureerde P2P-architecturen gaan het best om met de vergankelijke aard van een P2P-netwerk en zijn bovendien gemakkelijk te configureren.

Uit de historische evolutie van P2P-protocollen kunnen enkele belangrijke bouwstenen voor een nieuw multimedia streaming protocol gehaald worden. Een zuivere P2P aanpak zoals bij Gnutella is om twee redenen niet haalbaar. Allereerst zou net zoals bij de oorspronkelijke Gnutella implementatie een netwerk van bepaalde grootte verstoppen door een overvloed aan controleverkeer. Bovendien zal multimedia vaak ook verspreid worden vanaf een centrale autoriteit die enige controle zal willen over de distributie. Deze verdeelpunten zullen de huidige streaming servers vervangen en zijn dus krachtigere machines dan de doorsnee peer in het netwerk. Het grote verschil met de huidige uni-/multicast situatie is dat om eenzelfde aantal gebruikers te bedienen een kleiner aantal van dergelijke multimedia servers nodig zijn.

De verdeelpunten kunnen zonder veel moeite vergeleken worden met het fenomeen van de superknoten bij het FastTrack protocol. Superknoten zijn individuele peers die als relay- en

proxyserver dienen voor een groep gebruikers en voor deze gebruikers de datastroom en connecties beheren. De semi-gedistribueerde aanpak maakt het mogelijk data te decentraliseren zonder een te grote overhead bij elke afzonderlijke peer te creëren. Dit gebeurt ten koste van een verhoogde werklust bij de superknope zelf, in termen van consumptie van CPU tijd en bandbreedte. Het gebruik van dergelijke verdeelpunten sluit echter niet uit dat peers hun eigen data beschikbaar kunnen stellen. De superknoop zal voor elke peer in zijn groep automatisch alle gedeelde data beheren.

Om gemakkelijk deelstromen van de multimedia te verzenden, kan gebruik worden gemaakt van downloaden in stukjes (chunks) zoals dat bij het BitTorrent protocol het geval is. Bovendien kan de transmissie van deze stukjes gebeuren zoals bij een netarchitectuur als het BitTorrent protocol en de P2P-bestandsoverdracht op deze wijze aanzienlijk versneld worden. De locatie van de nodige stukken en de coördinatie van de overdrachten gebeuren vanuit de superknoop voor een gegeven peergroep.

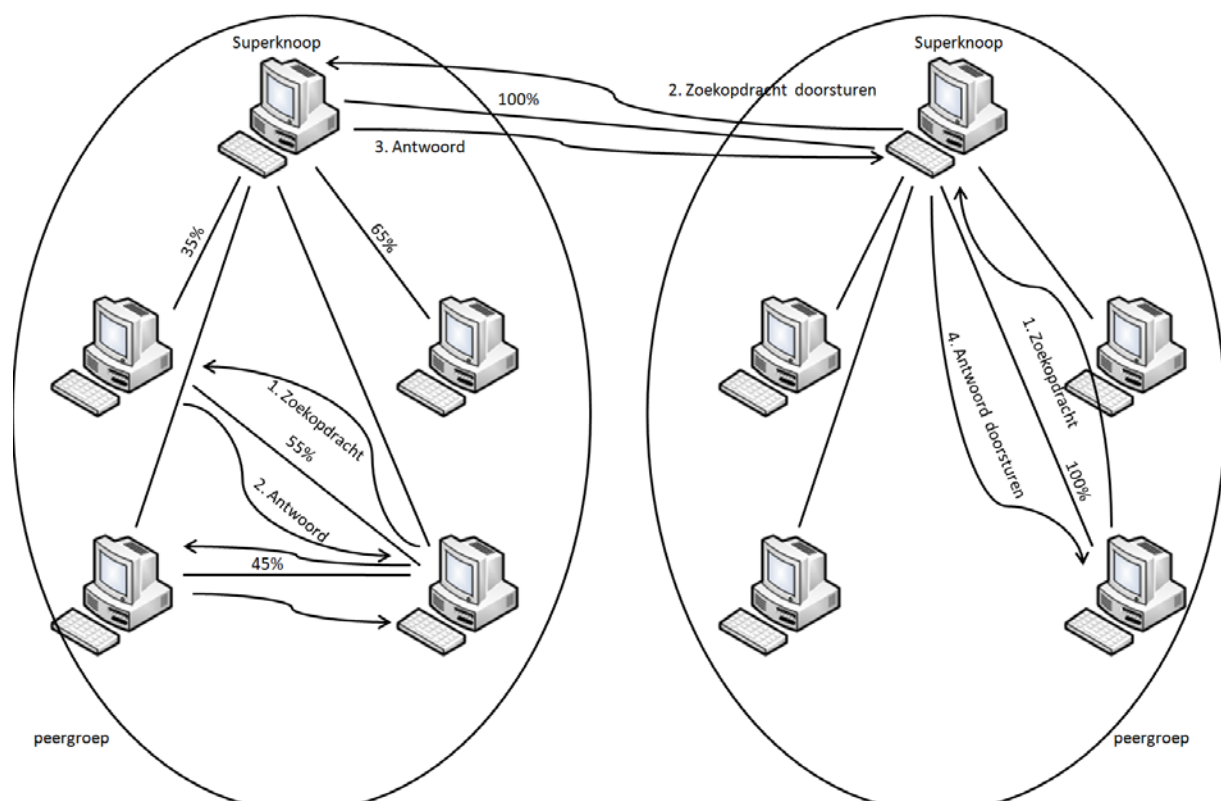


Fig. 2.8: Schets van een mogelijke combinatie van verschillende P2P-technieken.

Fig. 2.8 schetst de interne werking van een hybride protocol ontstaan uit de aggregatie van de verschillende geselecteerde bruikbare elementen van andere P2P-protocollen. In de linkerpeergroep op de figuur is een interne zoekopdracht en dataoverdracht voorgesteld, terwijl vanuit de rechterpeergroep een zoekopdracht en bijhorende download van in een andere peergroep wordt weergegeven. Bij deze laatste gegevensoverdracht is het ook de bedoeling het belang van de superknope als proxies duidelijk te maken.

HOOFDSTUK 3

P2P MULTIMEDIA STREAMING PROTOCOL: SPECIFICATIE

Na een overzicht gegeven te hebben van de mogelijkheden van P2P-systemen is het de bedoeling om in dit hoofdstuk aan de hand van de voorgestelde technieken, mits enkele optimalisaties, een protocol te beschrijven dat geschikt is voor het stromen van multimedia over een P2P-architectuur en dat bovendien de mogelijkheid biedt samenstellingen van de media te personaliseren.

Bij het ontwerp van de architectuur wordt er van uitgegaan dat verschillende types multimedia over het netwerk moeten getransporteerd kunnen worden, niet enkel een scala aan video- en audiogecodeerde bestanden maar ook verschillende tekstbestanden, presentaties, etc. Het is bijgevolg niet de bedoeling om een protocol te ontwikkelen dat geoptimaliseerd is voor één bepaald type of codec, noch om aanpassingen in de verschillende coderingen aan te brengen om de overdracht te versnellen. Het is de intentie enkele collectieve kenmerken van digitale objecten die onder de noemer "multimedia" vallen te distilleren en een algemene P2P-architectuur op te stellen die geoptimaliseerd is voor het stromen van een bestand met dergelijke kenmerken.

Het personalisatieprobleem van de multimedievoorstelling zorgt ervoor dat niet enkel met de multimediaopslag en -distributie rekening gehouden moet worden. Er is bovendien het vraagstuk over het fysiek samenstellen en weergeven van de presentaties. De nodige metadata bij de multimedia kan ervoor zorgen dat informatie over de inhoud en karakteristieken van de bestanden bijgehouden en onderzocht kunnen worden. De precieze integratie van metadata in het systeem wordt in het tweede deel van deze masterproef uitvoerig uit de doeken gedaan. Voorlopig zal met vage omschrijvingen naar deze functionaliteit van het protocol verwezen worden.

3.1 VOOROPGESTELDE VEREISTEN

3.1.1 P2P-ARCHITECTUUR

Allereerst is het de bedoeling dat voor de verspreiding van de data een P2P-architectuur wordt gebruikt. Zoals besproken in het vorig hoofdstuk zijn de voornaamste redenen om te opteren voor dergelijk systeem het ontlasten van de gebruikelijke servergeoriënteerde multimedia streaming services en de robuustheid van de netwerkarchitectuur.

De verdeling van de verwerkingsopdracht is inherent aan het P2P-devis. Het aanbieden van multimediestromen (streaming media) is een systeembronintensief proces en neemt aanzienlijk wat servercapaciteit in beslag. Het uitbouwen van een schaalbare serverarchitectuur voor het leveren van dergelijke diensten gebeurt dus ten koste van een aanzienlijk prijskaartje. Als wat van de bronintensieve bewerkingen naar de clients kan verschoven worden, zal op deze manier het kostenplaatje voor de mediaproviders gedrukt worden. Bovendien is ook geen gespecialiseerde hardware meer nodig om media te verspreiden en kunnen zelfs gewone gebruikers hun bijdrage leveren aan de inhoud van het netwerk.

Een P2P-architectuur is ook bevorderlijk voor de robuustheid van het distributienetwerk. In dergelijke opstelling zijn zowel de data als de infrastructuur die nodig is voor het verdelen van de data verspreid over verschillende peers. Bij het falen van een deel van het distributienetwerk

of zelfs de distributieserver, kan de rest van het netwerk autonoom verderwerken. Met andere woorden zal niet elk van de afzonderlijke knopen nodig zijn voor het correct verspreiden van de multimedia.

3.1.2 MULTIMEDIA KENMERKEN

Voor het stromen van multimedia in het bijzonder moet met enkele eigenschappen van dergelijke digitale objecten rekening gehouden worden.

Allereerst is zowel alle multimedia sequentieel gecodeerd en moet ze bijgevolg ook op die wijze weergegeven worden. Om (een deel van) een mediabestand te kunnen voorstellen moet dus voldoende opeenvolgende data beschikbaar zijn. Indien dit niet het geval is kan de weergave incorrect zijn of is er helemaal niets voor te stellen.

Verder hebben multimediabestanden doorgaans omvangrijkere bestandsgroottes dan andere bestandstypes, zoals bijvoorbeeld platte tekstbestanden. Dit in combinatie met hun veelal strikt sequentiële aard leidt ertoe dat het stromen van dergelijke bestanden relatief grote bandbreedtes vereist. Bij het stromen van data moet immers rekening gehouden worden met de tijdslimiet voor het verkrijgen van de nodige data om het tijdig en correct weergeven van de multimedia te kunnen garanderen. Een parallelle bestandsoverdracht van verschillende fragmenten van het weer te geven multimediabestand is bijgevolg niet weg te denken uit een efficiënte P2P-architectuur. Door de parallelle overdracht via een heterogeen P2P-netwerk zal de volgorde van de segmenten na het downloaden wellicht verstoord zijn. Om een correcte weergave uit een samenstelling van deze segmenten te bekomen, zullen ze dan ook na de overdracht opnieuw geordend moeten worden.

Beide problemen kunnen eenvoudig opgelost worden door een herorderingsbuffer (reorder buffer) in te voeren. Allereerst heeft die tot doel grotere sequentiële segmenten uit de gedownloade stukken samen te stellen, om dus de gedownloade segmenten te rangschikken. Bovendien zorgt deze buffer ervoor dat enkel zinvolle combinaties van segmenten zullen vrijgeven worden. Hij zal bijvoorbeeld enkel worden vrijgegeven als hij vol genoeg is om een weergeefbare multimediapresentatie af te leveren.

3.1.3 PERSONALISATIE MOGELIJKHEDEN

Verder is het de bedoeling dat de multimedia in de voorgestelde applicatie gepersonaliseerd kan worden. Met de term personaliseren wordt bedoeld dat selecties en/of samenstellingen van media uit de volledige multimediatebibliotheek van het netwerk kunnen gemaakt worden die aan de specifieke noden van de gebruikers voldoen. De noden van de gebruiker kunnen niet alleen betrekking hebben op de inhoud van de multimedia maar ook rekening houden met de technische beperkingen van het afspeelmedium.

De gedeelde multimedia hoeft hiervoor niet enkel geannoteerd te zijn. De architectuur moet het ook mogelijk maken om een voorstelling dynamisch samen te stellen. Verschillende fragmenten moeten kunnen worden geselecteerd, samengevoegd en weggeknipt uit de multimediabestanden van verschillende types en coderingen. Vanuit het voornemen om niets aan de fysieke representatie van de multimedia te wijzigen, is het uitermate belangrijk dat de samenstelling gebeurt zonder een transcoding van de multimediasegmenten naar een homogeen formaat uit te voeren.

De details over de verwerking van de metadata-annotaties in het P2P-systeem staan uitvoerig beschreven in het tweede deel van dit proefschrift. Bij de beschrijving van het selectie- en overdrachtsmechanisme verder in dit hoofdstuk wordt de samenstelling van de verschillende multimedia volgens het voorgestelde protocol uitgewerkt.

3.2 TRADITIONELE P2P-ACTOREN IN EEN HYBRIDE SYSTEEM

Enkele veelgebruikte begrippen in verband met P2P zullen eerst besproken worden, zoniet zullen ze telkens opnieuw uitgelegd moeten worden wanneer ze gebruikt worden in de bespreking van het protocol. Om niet telkens een context te hoeven schetsen worden de standaard bouwstenen van reeds geschetste⁶ hybride P2P-protocol hier dus kort uiteengezet.

3.2.1 PEERGROEP

Een peergroep is letterlijk niet meer dan een groep peers die als een subnetwerk in het complete P2P-netwerk geconfigureerd zijn. De bedoeling van dergelijke constructie is de complexiteit van het onderhoud van een groter netwerk te omzeilen door dit netwerk op te delen in kleinere, meer controleerbare deelnetwerken.

De interne organisatie van een peergroep is een pure P2P-architectuur waardoor alle voordelen van dergelijke structuur kunnen gebruikt worden en tegelijk, door het verdelen in kleinere netwerken, de groep bestuurbaar blijft. Door het relatief kleine aantal knopen in dergelijke subnetwerken zal het controleverkeer immers beperkt blijven en het gebruikelijke probleem bij deze structuur, het isoleren van een knoop door accumulatie van controleverkeer, zich niet voordoen. Om de verschillende peergroepen te combineren in één overkoepelend hybride-P2P netwerk wordt een systeem van superknopen gebruikt.

Een soort gelijk mechanisme wordt bij BitTorrent gebruikt, waar een peergroep een “zwerm” (swarm) wordt genoemd. Een zwerm is een verzameling peers die verbonden zijn met dezelfde superknoop, wat in het torrentprotocol een “tracker” is, en die eenzelfde bestand downloaden. Een peergroep zal uiteraard op een andere manier moeten samengesteld worden voor een optimale prestatie van het protocol beschreven in dit proefschrift, aangezien elke gebruiker een eigen presentatie van de multimedia wil samenstellen. Dit probleem wordt in het tweede deel van dit werk uitvoerig onderzocht.

3.2.2 SUPERKNOOP

In P2P-netwerken, is een superknoop een normale peer⁷ die tevens dienst doet als proxyserver voor de peergroep. Naast het bieden van de mogelijkheid om knopen in verschillende subnetwerken transparant met elkaar te verbinden, zal deze knoop ook de gegevensstroom en verbindingen voor andere peers in het netwerk beheren. Door de coördinerende functie van de superknoop kan via een semi-gedistribueerde architectuur gegevens worden gedecentraliseerd zonder buitensporige logica in elke knoop te moeten voorzien. De toegenomen werklast op de superknopen vertaalt zich over het algemeen in een extra bandbreedte- en CPU-gebruik voor deze peers.

Sommige P2P-structuren laten het aan de gebruiker om te beslissen of hun knoop als superknoop fungeert, andere toepassingen zullen het besluit intern afdwingen zonder menselijke tussenkomst. Bij Skype (16) werden bijvoorbeeld standaard alle gebruikers

⁶ Zie Fig. 2.8: Schets van een mogelijke combinatie van verschillende P2P-technieken.

⁷ Zie 3.2.3 Normale peer

geconfigureerd als een superknoop, wat behoorlijk controversie teweeg bracht. Dergelijke configuratie is vooral vrij bandbreedteconsumerend en niet alle peers zijn uitgerust voor zulke taken. Ondanks de kritiek bleef Skype volhouden dat superknoepen op hun netwerk slechts gebruikt werden om informatie over wie er online was op een gegeven moment bij te houden, en niet gebruikt voor het routeren van oproepen tussen gebruikers. Vanaf versie 3.0 maakte Skype het mogelijk de superknoopmodus uit te zetten door een wijziging in het Windows-register aan te brengen.

Iedere peergroep die niet enkel binnen zijn eigen subnetwerk wil opereren zal van ten minste één superknoop moeten worden voorzien, aangezien de semi-server niet enkel interne communicatie coördineert maar ook de interactie tussen verschillende peergroepen. Het is logischerwijs aangewezen om voor grotere peergroepen meerdere superknoepen te voorzien om er voor te zorgen dat de superknoop binnen een aanvaardbare tijd de verschillende peers van zijn subnetwerk kan bedienen.

3.2.3 NORMALE PEER

Elke peer die geen superknoop is, heeft dezelfde minimale verantwoordelijkheden en valt onder de noemer “knoop”, “peer” of “normale peer”. Deze peers zijn verbonden met een peergroep via een superknoop en moeten enkel de mogelijkheid bieden om multimedia te delen en media van een andere peer op te vragen. In het geval van het streaming protocol zoals in dit werk beschreven zal de peer ook moeten kunnen beslissen wanneer kan begonnen worden met het weergeven van de data die wordt afgehaald.

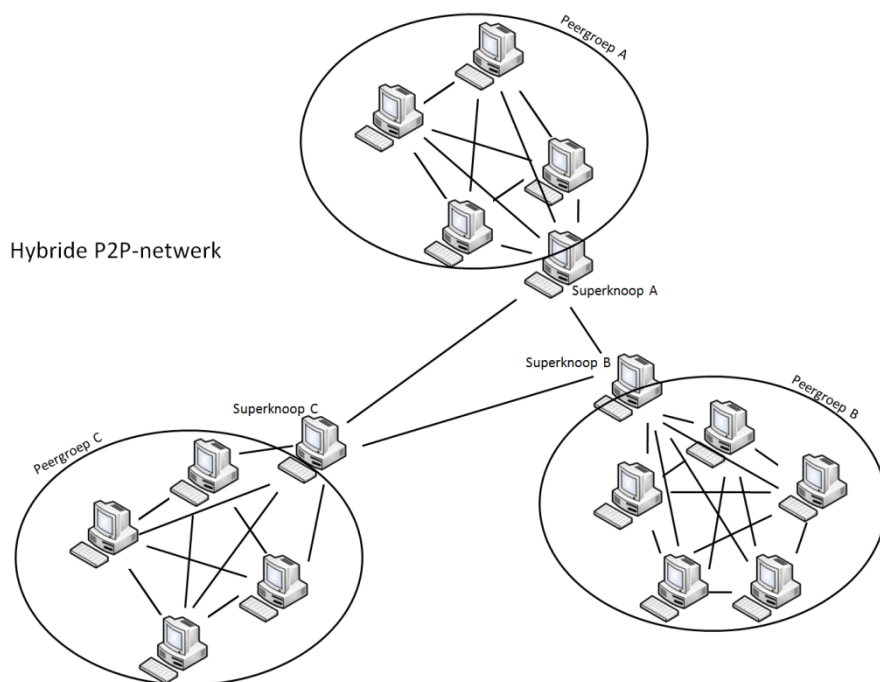


Fig. 3.1: Schets van interactie van traditionele bouwstenen in een hybride P2P-configuratie.

3.3 SPECIFICATIE

Nu de algemene vereisten voor een protocol dat P2P-streaming van gepersonaliseerde multimedia mogelijk maakt afgelijnd zijn, kan een meer gedetailleerde beschrijving van de specifieke interacties van verschillende onderdelen in het systeem geïntroduceerd worden. Hieronder worden de basisfunctionaliteiten volgens de demoapplicatie stap voor stap uitgelegd. De activiteiten zijn in chronologische volgorde naar de activiteiten van een peer in het netwerk weergegeven. De demotoepassing blijft een proof-of-concept en zal slechts enkele mogelijkheden van het protocol reflecteren. Aanvullingen kunnen zonder grote moeite op een later tijdstip toegevoegd worden, maar liggen buiten de kern van het onderzoek in deze masterproef.

3.3.1 AANMELDEN

Een peer moet zich uiteraard bij een netwerk van andere peers, i.c. via een peergroep, kunnen vervoegen. Eerst moet in een globaal register worden nagaan of dergelijke groep reeds aanwezig is, afhankelijk van het bestaan van de peergroep zijn nu twee scenario's mogelijk.

Als een dergelijke groep nog niet bestaat, zal die moeten worden aangemaakt alvorens een peer tot de groep kan toetreden. Een peer die een groep aanmaakt, wordt best meteen een superknoop voor dit subnetwerk zodat de groep via deze superknoop direct in de globale catalogoog kan worden geregistreerd. Hoewel dit mechanisme niet met absolute zekerheid kan garanderen dat verschillende peers niet tegelijk een gelijkaardige groep aanmaken, zal het de kans om een aangemaakte peergroep te detecteren aanzienlijk vergroten. Er is bij deze aanpak met andere woorden een reële kans dat verschillende subnetwerken worden aangemaakt voor peers die in eenzelfde peergroep zouden kunnen ondergebracht worden. Dit veroorzaakt echter geen onoverkomelijke problemen. De verschillende netwerken kunnen via hun respectievelijke superknoopen met elkaar communiceren en een goede peergroepcatalogoog zal ervoor zorgen dat de verdeling van aanmeldende peers over de groepen op een uniforme wijze verloopt.

In het geval dat de benodigde peergroep al bestaat, kan de peer eenvoudigweg het lidmaatschap aanvragen. Wanneer de knoop om de één of andere reden geen lid kan worden van de groep moet die zonder meer reageren alsof de groep nog niet bestaat en een eigen groep oprichten. Voor een optimale interne werking van de peergroepen zal een nieuwe peer in het netwerk moeten controleren of de verhouding tussen superknoopen en andere (normale) peers nog op peil is. Zoniet kan die alsnog gepromoveerd worden en als superknoop fungeren.

Fig. 3.2 geeft een schematisch overzicht van het aanmeldingsproces van een peer bij een bestaande peergroep. Het geval waar de peer zelf nog eerst een nieuwe peergroep moet aanmaken alvorens ze tot de groep kan toetreden is vrij gelijkaardig voor te stellen. In plaats van te verbinden met de superknoop en eventueel de knopen van de geselecteerde peergroep, zal de peer zijn eigen groep oprichten met als eerste lid en superknoop zichzelf waarna de nieuwe peergroep bij het globale peergroep register wordt ingeschreven.

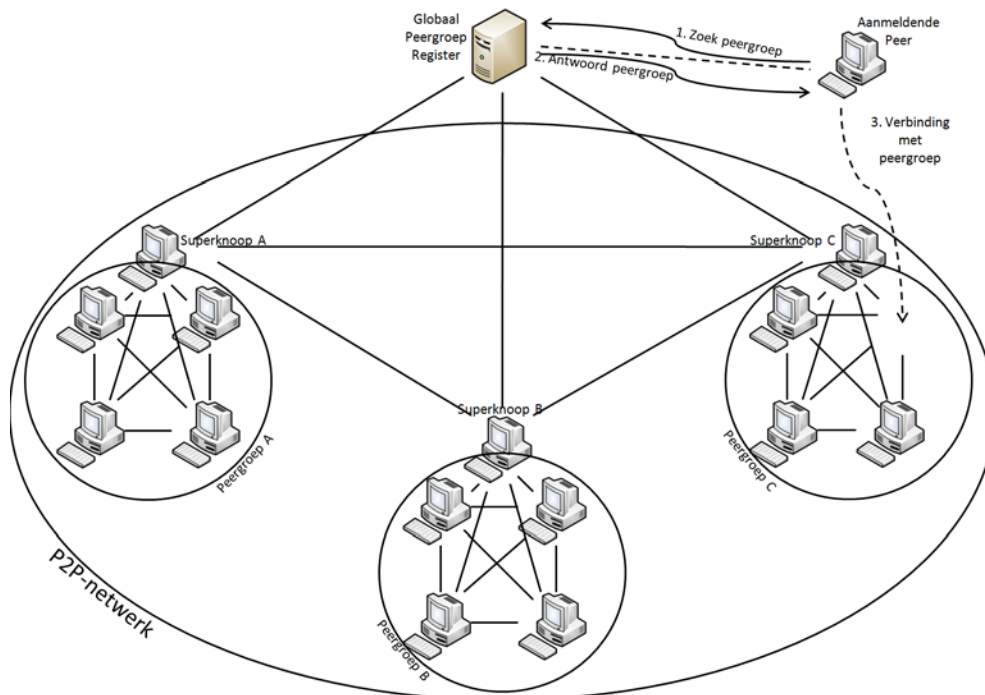


Fig. 3.2: Aanmelden van een peer bij het P2P-netwerk door lid te worden van een peergroep.

3.3.2 DELEN

Eens een peer lid is van een bepaalde peergroep en dus verbonden met een P2P-netwerk is het de bedoeling dat de knoop multimedia over het netwerk gaat delen. De te verspreiden data kan bestaan uit eerder gedownloade fragmenten of eigen inhoud die de peer via het netwerk wil distribueren. In beide gevallen moet de media van de nodige metadata voorzien zijn die duidelijk zowel de inhoud als de technische aspecten (codering, resolutie, etc.) van het fragment beschrijven. De specifieke taak van het annoteren behoort niet tot de kernactiviteiten van het onderzoek verricht tijdens deze masterproef, bijgevolg worden de metadata-annotaties als voorhanden beschouwd.

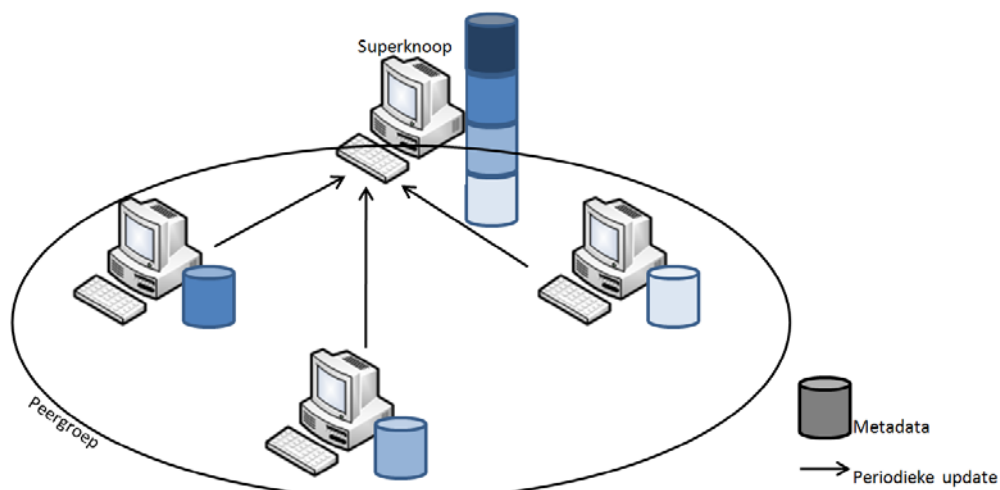


Fig. 3.3: Metadata-advertentie van een peergroep.

De informatie in de metadata zal gebruikt worden om de bestanden te adverteren zodat andere geïnteresseerde partijen weten waar en in welke hoedanigheid de data terug te vinden is. Zoals weergegeven op Fig. 3.3 houdt elk van de peers informatie bij over de bestanden die door hen

gedeeld worden. Om te kunnen delen over de grenzen van een peergroep heen moet de superknoop ook periodiek op de hoogte worden gebracht van de gedeelde multimedia in elk van de knopen in zijn groep.

3.3.3 ZOEKEN

Zoekopdrachten kunnen volgens het zoekdomein in twee types ingedeeld worden, in de categorieën intern en extern zoeken.

Intern zoeken gebeurt binnen de lokale peergroep waartoe de zoekende peer behoort. De peergroep is zoals reeds herhaaldelijk vermeld intern geconfigureerd in een pure P2P-opstelling. Het intern zoeken zal dus gebeuren zoals bij het Gnutella protocol. Wanneer een peer een bestand met bepaalde eigenschappen wil zoeken binnen de groep, zal hij een zoekopdracht sturen naar al zijn burens, die zullen het bericht op hun beurt doorsturen naar hun burens, etc. De knopen die gedeelde multimedia hebben waarvan de beschrijving in de metadata voldoet aan de zoekopdracht zullen een verbinding maken met peer waarvan de zoekopdracht werd uitgevoerd en de knoop via directe communicatie antwoorden welke van hun bestanden aan de aanvraag voldoet.

Wanneer buiten de grenzen van de lokale peergroep wordt gezocht, spreekt men van extern zoeken. Bij extern zoeken zal een zoekende peer een zoekopdracht naar de superknoop van de peergroep waartoe die behoort zenden. De superknoop zal de zoekopdracht afleveren aan de superknopen van andere peergroepen waarmee hij verbonden is. Aangezien de superknopen metadata bijhouden van praktisch⁸ alle multimedia kun zij een nauwkeurig beeld geven van welke multimedia zich in hun deelnetwerk bevindt. De ontvangende superknoop zal een antwoord formuleren en via de lokale superknoop zal het antwoord bij de vragende peer terecht komen. Het zoekmechanisme bij extern zoeken is dus gemodelleerd op het zoeken in het FastTrack protocol.

In veel gevallen zal eerst intern gezocht worden alvorens een externe zoekopdracht wordt begonnen. Een externe zoekopdracht zal meer tijd in beslag nemen omdat tussen twee verschillende peergroepen gecommuniceerd wordt via een enkele verbinding. Als de peergroepen bovendien geografisch zijn samengesteld, als intern zoeken bijvoorbeeld over een LAN gebeurt en de communicatie tussen twee superknopen over een WAN van derden, zal extern zoeken over deze gebruikelijk trager connectie wel degelijk een betekenisvolle grotere latentie introduceren. Daarom zal enkel wanneer een interne zoekopdracht geen of onvoldoende antwoorden oplevert, beslist worden om een externe zoekopdracht uit te voeren.

De beide zoektechnieken worden in *Fig. 3.4* visueel voorgesteld. Om de figuur licht en leesbaar te houden zijn enkel de noodzakelijk netwerkverbindingen weergegeven. Men mag er van uitgaan dat binnen een peergroep de peerconnecties gemodelleerd zijn op een complete graaf. In peergroep A wordt een interne zoekopdracht uitgevoerd. Een gegeven peer zendt een zoekopdracht naar al zijn burens en krijgt op die wijze een antwoord terug van alle peers of ze al dan niet multimedia hebben die aan deze opdracht voldoet. In peergroep B is er een peer die geen inhoud intern vindt en daarom zijn superknoop vraagt een zoekopdracht te lanceren in een andere peergroep. Hier wordt de superknoop A bevraagd. Superknoop A antwoordt terug aan de superknoop van B en deze zal het antwoord doorsturen naar de juiste peer in peergroep B.

⁸ De metadatacatalogus van een superknoop wordt periodiek aangepast. Een aanvraag tussen twee dergelijke momenten kan dus op een verouderde versie van de informatie gebeuren.

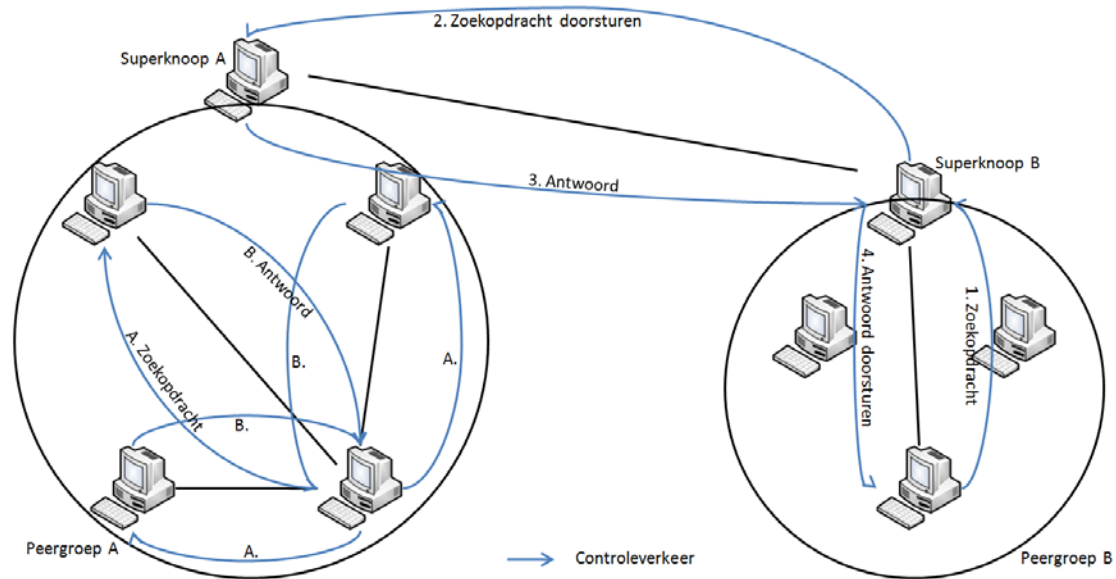


Fig. 3.4: Intern en extern zoeken in peergroepen.

3.3.4 BESTANDSOVERDRACHT

Nu de peer de gewenste multimedia kan lokaliseren, moeten de bestanden nog steeds op de één of andere manier naar de eigen opslag getransporteerd worden. Zoals reeds vermeld zullen voor de efficiënte overdracht van multimediatebestanden, die over het algemeen een omvangrijkere bestandsgrootte hebben, meerdere parallelle verbindingen gebruikt worden. Elk van deze verbindingen kan tegelijk een ander fragment van de multimedia van een verschillende peer downloaden.

Los van het feit dat het controleverkeer binnen een BitTorrent zwerm tot een minimum beperkt blijft, is de zwerm bovendien bijzonder schaalbaar, efficiënt en eenvoudig te implementeren. Deze interessante eigenschappen maken BitTorrent een aantrekkelijke keuze om een P2P multimediatestreamingprotocol op te modeleren (17). Daarenboven vertonen BitTorrent zwermen enkele eigenschappen die eveneens essentieel zijn bij een P2P-netwerk voorzien op het stromen van multimedia. Bij zowel BitTorrent als multimediatestreaming moet de inhoud worden verdeeld in meerdere stukken of segmenten om te kunnen worden gedownload van de server. Bovendien is het inherent aan het protocol dat de segmenten op een parallelle manier worden overgedragen.

Hoewel de overeenkomsten niet te weerleggen zijn, is BitTorrent op zich niet geschikt voor het stromen van multimedia. Het protocol houdt immers helemaal geen rekening met de tijdsgevoeligheid bij het overbrengen van de data die aan het stromen van multimedia verbonden is. In de eerste plaats zullen peers de segmenten niet in volgorde afhalen, wat ervoor zorgt dat de multimedia slechts weergegeven kan worden als de volledige download voltooid is. Ten tweede incorporeert BitTorrent een mechanisme dat de eerlijkheid van het P2P-principe, iedereen draagt zijn steentje bij, moet garanderen. Eenvoudig uitgedrukt krijgen peers die een betere bijdrage hebben geleverd tot het systeem, peers waarvan de verhouding upload/download hoger is, eerder de kans om een snellere connectie te verkrijgen bij het downloaden van segmenten. Op het moment dat een peer begint met stromen zal deze nog geen segmenten beschikbaar hebben om te delen. Bijgevolg zal de toename van de bijdrage maar met mondjesmaat op gang komen en zal deze eigenschap dus niet bijdragen tot de vloedigheid van het

opstarten van een stroom. Eerlijkheid binnen het P2P-principe moet dus op een andere manier worden afgedwongen. Het eenvoudigste mechanisme is om na het afspelen het multimediasegment te blijven delen in de peergroep tot de bijdrage aan een bepaalde verhouding voldoet en zo een bijdrage af te dwingen nadat de data is weergegeven in plaats van alvorens het wordt afgehaald.

In het onderzoek van Shah en Pâris (18) wordt voorgesteld helemaal geen rekening te houden met de bijdrage van de peer aan het P2P-systeem en worden de peers waarvan gedownload wordt willekeurig uit een lijst met beschikbare verbindingen gekozen. In plaats van deze willekeurige selectie kan een meer eenvoudige methode gehanteerd worden, een gretige aanpak die eveneens geen rekening houdt met de bijdrage van de peer in de multimediaspreiding en telkens de snelste beschikbare verbinding verkiest. Om het downloadproces verder te optimaliseren moet dus een historiek over de verbindingen worden bijgehouden. Wanneer dan x aantal peers een bepaald fragment aanbieden, kan in deze statistiek worden nagegaan welke van de verbindingen in het verleden over het algemeen de snelste overdracht bood en deze connectie voorrang geven op een andere.

Verder zullen de mediasegmenten dus in temporele volgorde moeten worden gedownload en niet volgens het BitTorrentprincipe waarbij het zeldzaamste segment eerst wordt verkregen. De stukken mogen dan nog in volgorde gereserveerd worden doordat de parallelle overdracht van de segmenten over een heterogeen netwerk gebeurt, zullen de stukken niet in die volgorde afgeleverd worden. De reeds voorgestelde herorderingsbuffer⁹ moet de segmenten in de eerste plaats terug in de originele volgorde plaatsen. In Fig. 3.5 wordt een beeld van de werking van de herorderingsbuffer weergegeven. Bij de downloadaanvraag van de segmenten wordt rekening gehouden met de weergavevolgorde zodat de stukken min of meer in de juiste volgorde aankomen en een relatief kleine herorderingsbuffer efficiënt genoeg blijkt. Hoe groter immers de spreiding van de aankomst van de verschillende opeenvolgende segmenten uit het originele bestand, des te groter de buffer logischerwijs moet zijn om sequentie van dezelfde grootte opnieuw te kunnen samenstellen. Wanneer opeenvolgende segmenten dichter na elkaar aankomen zal niet enkel de herorderingsbuffer minder groot moeten zijn, de weer te geven multimediasequentie zal ook sneller voorhanden zijn.

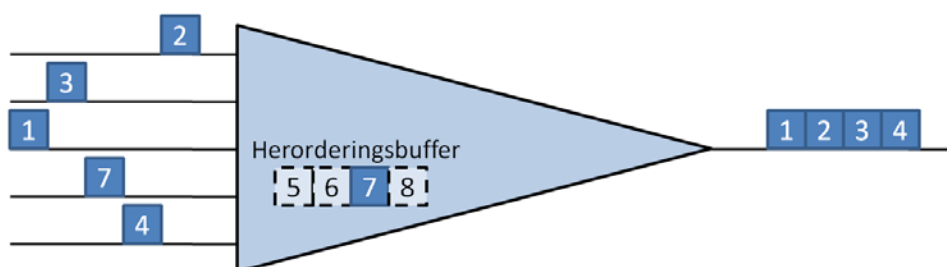


Fig. 3.5: Werking van een Herorderingsbuffer.

Beide optimalisaties zijn een goed idee om de P2P-download te versnellen en houden het streaming aspect van het protocol in acht. Eens de weergave van de download echter is gestart, is het de bedoeling dat deze blijft afspelen. Er moet dus met andere woorden een zekere maat van vooruitgang kunnen worden gegarandeerd. Eenmaal de weergave is gestart, zal een minimale overdrachtbandbreedte moeten geleverd worden. Om van deze minimumvereiste

⁹ Zie 3.1.2 Multimedia kenmerken

geen harde, te strenge systeemlimiterende factor te maken zullen in de weergavemodule allerlei mechanisme worden verkend om deze eis af te zwakken.

3.3.5 WEERGAVE

Een tweede functie van de herorderingsbuffer is het bijhouden van de segmenten terwijl ze gecombineerd worden tot een groter afspeelbaar multimedia fragment. In *Fig. 3.5* zal gepoogd worden 4 opeenvolgende segmenten te combineren alvorens ze langs de bufferuitgang passeren. Idealiter zal de buffer pas vrijgegeven worden als deze volledig vol is. Bij een te lage bandbreedte tijdens het stromen van multimedia echter kunnen alle afgewerkte buffersegmenten weergegeven zijn alvorens de huidige buffer compleet is. In dit geval kan gekozen worden voor twee verschillende continuïteitsoplossingen wanneer één of meerdere segmenten niet op tijd kan worden afgeleverd. In een eerste geval wordt de herorderingsbuffer vrijgegeven onafhankelijk van het feit dat deze incompleet is en bepaalde stukken van de voorstelling al dan niet ontbreken. De segmenten die wel voorhanden zijn worden zo goed en kwaad mogelijk voorgesteld. Een tweede oplossing bestaat erin de weergave te onderbreken tot een nieuwe volledige buffer vrijgegeven kan worden. Een voldoende grote buffer is niet enkel groot genoeg om een weergeefbaar fragment te cachen, maar kan ook anticiperen op bandbreedtefluctuaties zodat de weergavestroom eens aangevat zo weinig mogelijk onderbroken wordt.

Aangezien de multimedia van verschillende types kan zijn zal een gepersonaliseerde weergave uit segmenten met verschillende coderingen bestaan. Om dit probleem te omzeilen kan gebruik worden gemaakt van een afspeellijst waaraan de af te spelen stukken kunnen worden toegevoegd ongeacht hun codering. Deze afspeellijst moet dan van boven naar onder worden weergegeven door middel van een multimediamviewer die al deze coderingen kan verwerken.

3.3.6 AFMELDEN

Een P2P-systeem heeft een vergankelijke staat. Het is eigen aan het netwerk dat peers zich aanmelden en vertrekken uit de configuratie op eigen initiatief. Het protocol voor het aanmelden werd reeds hierboven beschreven. Bij het afmelden zal de peer de compleet tegengestelde bewerkingen uitvoeren. Allereerst zullen de advertenties voor de gedeelde bestanden zowel lokaal op de knoop als globaal op de superknoop moeten worden neergehaald. Op deze manier zullen de zoekopdrachten niet langer rekening houden met de data op de peer en zal deze niet meer worden gebruikt voor multimediaoverdracht. Wanneer dit gebeurd is kan de peer het netwerk verlaten.

HOOFDSTUK 4

P2P MULTIMEDIA STREAMING PROTOCOL: IMPLEMENTATIE

Voor de implementatie van het P2P-protocol voor het stromen van multimedia in een proof-of-concept applicatie werd gekozen voor de JXTA API¹⁰ van Sun Microsystems. JXTA, uitgesproken als "juxta", komt van de Engelse term "juxtapose" wat niet meer betekent dan "bij elkaar plaatsen". JXTA omvat operaties waardoor peers tijdelijke associaties maken om zich in een P2P-netwerk te configureren, ze plaatsen zich met andere woorden bij elkaar. JXTA is geen ontwerpfilosofie voor software, noch is het een softwaretoepassing. Het is een API die bestaat uit een verzameling protocollen die softwareontwikkelaars kunnen gebruiken binnen een eigen technologie naar keuze om P2P-functionaliteit toe te voegen aan hun toepassingen.

4.1 WAAROM JXTA?

Het JXTA project werd oorspronkelijk opgestart door Sun Microsystems. Aan het ontwerp werkten een klein aantal experts van academische instituten en de industrie mee. Uitgaande van wat beschouwd werd als tekortkomingen in vele bestaande P2P-protocollen of P2P-systemen die in ontwikkeling waren, werden enkele doelstellingen (19) vooropgesteld. Net deze eigenschappen zijn de reden waarom voor JXTA werd gekozen boven andere oplossingen als het Microsoft .Net Framework 3.5¹¹ of GUNet¹².

4.1.1 INTEROPERABILITEIT

Veel van de bestaande P2P systemen worden ontwikkeld met oog op het leveren van één bepaalde soort dienst. Zo werden Napster, Gnutella, BitTorrent, etc. ontworpen specifiek voor het delen van bestanden en is ICQ voor instant messaging bedoeld. Gegeven de diverse karakteristieken van deze diensten en het gebrek aan een gemeenschappelijk onderliggende P2P-infrastructuur, maakt elke P2P-softwareontwikkelaar een systeem dat volgens een eigen set van protocollen werkt en dus incompatibel is met dat van anderen. Rond elk systeem is zo een afzonderlijke gemeenschap van gebruikers ontstaan, waardoor er ettelijke energie wordt verspild omdat er telkens oplossingen worden uitgedacht voor basiselementen die eigenlijk gemeenschappelijk zijn voor alle P2P-systemen. Bovendien, wil een peer deelnemen aan de meerdere gemeenschappen van de verschillende P2P-ontwikkelaars, dan moet die peer alle implementaties ondersteunen, elk voor een bepaald P2P systeem of gemeenschap.

Dit doet denken aan de situatie van het internet, vooraleer de browser op het toneel kwam. Om internettoegang te verkrijgen, moest men zich inschrijven bij AOL, Prodigy of CompuServe. De gebruiker was daarna verbonden aan de ene gemeenschap waarvoor die had gekozen en de service providers moesten hun diensten of informatie aanpassen aan de wijze waarop deze gemeenschap werkte.

Het JXTA project wil voor de P2P-wereld betekenen wat HTTP en de browser voor het internet hebben gedaan.

¹⁰ <http://jxta.kenai.com>

¹¹ <http://msdn.microsoft.com/en-us/library/cc297274.aspx>

¹² <http://www.gnu.org/software/gnunet>

4.1.2 PLATFORMONAFHANKELIJKHEID

De meeste P2P-systemen bieden hun diensten aan via een set Application Program Interfaces (API's), die werken onder één bepaald besturingssysteem en gebruik maken van een gegeven netwerkprotocol. Zo zal het ene systeem een set C++ API's aanbieden, oorspronkelijk enkel op Windows draaien en het TCP/IP netwerkprotocol gebruiken. Een ander systeem zal voor een combinatie van C en Java API's opteren, draaien op verschillende Unix-systemen en naast TCP/IP ook HTTP vereisen. Een P2P-ontwikkelaar zal dus niet enkel moeten beslissen voor welke set API's hij zal programmeren, maar ook voor welke soort P2P-gebruikers zijn software bedoeld zal zijn: PC, Linux of Mac.

Omdat er weinig kans is dat de verschillende systemen inter-operabel zijn, zal een programmeur, wil hij zijn dienst aan beide gemeenschappen aanbieden, dezelfde dienst twee keer moeten ontwikkelen voor de twee platformen of moet hij een soort adapter tussen de beide systemen implementeren. JXTA poogt onafhankelijk zijn van de programmeertaal, de ontwikkelingsomgeving of het uitvoeringsplatform.

4.1.3 ALOMTEGENWOORDIGHEID

Het is de bedoeling dat de JXTA-technologie toegepast kan worden op elk toestel dat een digitaal circuit bevat. Tot deze categorie kan alles van sensoren, gebruikselektronica, GSM's, PDA's, huishoudtoestellen tot netwerkroulers, desktop computers, datacenters en opslagservers gerekend worden.

JXTA wou de trend bij de ontwikkeling van veel P2P-systemen, vooral degene die door jonge bedrijfjes ontworpen werden en waarbij veelal voor Microsoft Windows als doelplatform voor de software gekozen werd, onderdrukken. De reden voor deze aanpak was eenvoudig: op die manier kon de grootste basis worden bereikt, wat dus de snelste weg naar winst betekende. Dit leidde er echter onvermijdelijk toe dat heel wat Wintel-afhankelijke kenmerken in het P2P-systeem slopen.

Deze manier van ontwikkelen resulteerde dus in een kortzichtige benadering van het P2P-probleem, P2P staat immers niet voor PC-to-PC maar peer-to-peer. De eerste demonstraties van P2P mogen dan wel op Wintel-machines gegeven zijn, al snel leek duidelijk dat de grootste integratie van P2P-technologie zich zou voordoen aan de twee extremen van het gebruikersspectrum: P2P voor grote systemen van bedrijven en de kleine toepassingen bij een eindgebruiker. Hoewel de configuraties van beide systemen vrij uiteenlopend zijn, wil JXTA ervoor zorgen dat de software door middel van een gemeenschappelijke basis aan gelijkaardige protocollen wordt opgebouwd.

4.1.4 MOTIVATIE

Om even kort samen te vatten zijn de JXTA protocollen ontworpen om onafhankelijk van de onderliggende implementatie opereren en vereisen de protocollen niet:

- het gebruik van een bepaalde programmeertaal of besturingssysteem.
- het gebruik van een bepaald netwerktransport of topologie.
- het gebruik van een bepaalde authenticatie, encryptie of beveiligingsmodel.

JXTA voorziet bovendien in een eenvoudig en generiek P2P-platform met alle basisfunctionaliteit die nodig is voor het leveren van allerhande P2P-netwerkdiensten:

- JXTA wordt gedefinieerd door een klein aantal protocollen. Elk protocol is eenvoudig te implementeren en te integreren in de P2P-diensten en -toepassingen. Zo kunnen de diensten aangeboden door één softwareontwikkelaar transparant gebruikt worden door de gebruikers van een ander P2P-systeem.
- De JXTA-protocollen horen onafhankelijk te zijn van een programmeertaal, zodat ze kunnen worden geïmplementeerd in C/C++, Java, Perl, en tal van andere talen. Heterogene apparaten met een compleet andere softwareconfiguratie kunnen toch samenwerken met behulp van de JXTA-protocollen.
- De JXTA-protocollen zijn ontworpen om onafhankelijk te zijn van transport protocollen. Ze kunnen worden uitgevoerd bovenop TCP/IP, HTTP, Bluetooth en vele andere protocollen.

Dit zijn de belangrijkste redenen waarom voor de proof-of-concept implementatie voor de JXTA API werd geopteerd.

4.2 JXTA WERKING

JXTA is een open source project en geniet dusdanig een voortdurende ontwikkeling. De technische concepten en eigenschappen ervan worden door Sun Microsystems beheerd. Bij de aanvang van het JXTA project werden reeds bestaande P2P-software architecturen geanalyseerd en vastgesteld dat er een gemeenschappelijk laag op het conceptueel niveau was, zoals afgebeeld op *Fig. 4.1*.

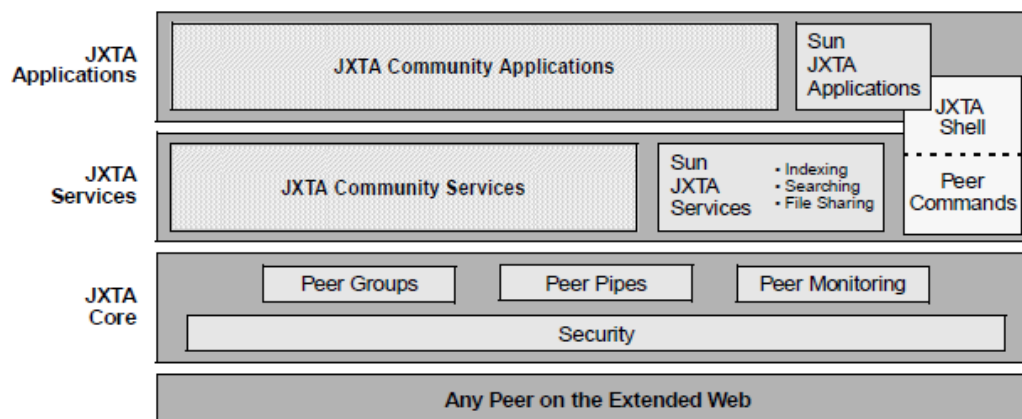


Fig. 4.1: P2P-softwarearchitectuur.

Een typische P2P-softwarearchitectuur kan gewoonlijk in 3 lagen opgedeeld worden. Onderaan in de kernlaag (JXTA Core) wordt gezorgd voor het tot stand brengen van de basisfunctionaliteit van een peer, communicatiebeheer zoals routing, en andere eenvoudig loodgieterswerk. De middelste laag is de dienstlaag (JXTA Services) die de concepten van een hoger niveau behandelt. In deze laag zitten functionaliteit als het indexeren, zoeken, en het delen van bestanden vervat. Deze onderdelen, die zwaar gebruik maken van de basisfunctionaliteit in de kernlaag, kunnen zelfstandig gebruikt worden maar worden meestal ook als componenten in een algemeen P2P-systeem opgenomen. Bovenaan bevindt zich de applicatielaag (JXTA Applications) met bijvoorbeeld toepassingen voor het delen van bestanden, instant messaging en natuurlijk het stromen van multimedia.

De JXTA is ontworpen om een laag bovenop de dienst- en toepassingslaag te voorzien. De laag moet zo dun mogelijk blijven, maar tegelijk interessante en krachtige basiselementen aanbieden die gebruikt kunnen worden in P2P-diensten en -toepassingen. Door deze laag minimaal te houden, kan een optimale interoperabiliteit tussen verschillende P2P-systemen behouden blijven, dit terwijl de ontwikkelaars niet in hun innovativiteit geremd worden.

4.2.1 JXTA ONDERDELEN

Op het hoogste abstractieniveau is JXTA niets meer dan een verzameling protocollen. Elk protocol wordt gedefinieerd door één of meer boodschappen die door de deelnemers in het protocol uitgewisseld worden. Elke boodschap heeft dan weer een op voorhand vastgelegd formaat en kan verschillende datavelden bevatten.

In dit opzicht is JXTA verwant aan het TCP/IP-protocol. TCP/IP verbindt internetknooppunten met elkaar, net zoals het JXTA-protocol de peers met elkaar verbindt. TCP/IP is platformafhankelijk omdat de specificatie is vastgelegd in een set protocollen, net als JXTA. Meer nog: JXTA is netwerkafhankelijk en kan TCP/IP gebruiken maar ook andere netwerkprotocollen.

Concepten

De JXTA concepten zien er opzettelijk zeer simplistisch uit aangezien de specificatie uitdrukkelijk dicteert deze eenvoudig moeten blijven gehouden. Een belangrijkere reden is dat er veel domeinen zijn waar niet één enkele juiste manier bestaat om iets gedaan te krijgen of waar wat moet gedaan worden kan afhangen van de aard en de context van de toepassing. Dit fenomeen is vooral aanwezig in de beveiliging van P2P-applicaties waar elke toepassing een andere authenticatiemethode kan hebben, een andere manier om communicatiebeveiliging te verzorgen, een andere encryptiealgoritme voor de beveiliging van gegevens, een andere regeling voor het authenticeren met een digitale handtekening of een ander beleid voor toegangsbeheer. Binnen deze domeinen zal JXTA trachten te onderspecificeren om zo de nadruk op mechanismen te leggen in plaats van het beleid zodat de ontwikkelaars van applicaties een maximale vrijheid genieten om te innoveren en concurreren bij het zoeken naar oplossingen voor deze problemen.

Identificatoren (identifiers) worden in JXTA gebruikt om te verwijzen naar een entiteit (peer, een advertisement, een service). De uniciteit van een identificator kan gemakkelijk gegarandeerd worden voor een lokale omgeving. Bij een globale omgeving met miljoenen gebruikers is dit vanzelfsprekend niet mogelijk. Aangezien een identificator gebonden wordt aan een naam en een netwerkadres zal deze tekortkoming echter geen reëel probleem vormen.

Advertenties (advertisements) zijn XML-documenten die voldoen aan voorgedefinieerde XML-schema's en gebruikt worden om systeembronnen (een peer, een pipe of een service, etc.) te benoemen, beschrijven en te publiceren.

Een *peer* is elke entiteit die alle protocollen bevat die nodig zijn om als peer te functioneren. Een peer kan zich manifesteren in de vorm van een processor, proces, machine of gebruiker.

Berichten (messages) zijn ontworpen om bovenop een asynchroon, onbetrouwbaar en unidirectioneel medium gebruikt te worden. Het bericht is gemodelleerd als een datagram en bezit een header, een berichtveld en optioneel een bron- en bestemmingsadres. De eindpunten worden door een transportlaag verbonden aan een fysiek adres. Het berichtformaat is ontworpen om verschillende transportstandaarden te ondersteunen.

Een JXTA *peergroep* is een entiteit die de peergroepprotocollen implementeert. Gewoonlijk zal een peergroep een verzameling van peers zijn die gemeenschappelijke diensten aanbiedt. De relatie tussen een peer en een peergroep is metafysisch. De specificatie dicteert niet hoe een groep moet worden aangemaakt, wat voor type een groep is of wat de voorwaarden zijn om lidmaatschap van de groep te verkrijgen. Een peer kan tegelijk tot meerdere groepen behoren die hij kan ontdekken door het Peer Discovery Protocol te gebruiken. Een speciale groep, de wereldpeergroep (World Peer Groep), bevat alle JXTA peers. Hoewel alle peers automatisch deel uitmaken van deze groep betekent dit niet dat ze allen elkaar kunnen ontdekken of kunnen communiceren met elkaar. Ze kunnen nog altijd gescheiden worden door verschillende netwerkpartities.

Pijplijnen (pipes) zijn asynchrone communicatiekanalen voor het verzenden en ontvangen van berichten. Ze zijn unidirectioneel, dus er bestaan twee soorten pijplijnen: inputpijplijnen en outputpijplijnen. Pijplijnen zijn een virtueel begrip in die zin dat een eindpunt van de pijplijn kan verbonden worden één of meerdere eindpunten in het netwerk. Pijplijnen worden dynamisch gebonden aan een peer via het Pipe Binding Protocol. Ze kunnen dus hergebruikt en aan verschillende peers gebonden worden gedurende de uitvoering. Dit is een handig mechanisme wanneer er ter bevordering van fouttolerantie gedurende de uitvoering een falende peer aan een pijplijn gewisseld moet worden. JXTA legt de interne werking van de pijplijn niet vast, gelijk welke unicast of multicast protocollen, algoritmes en hun combinaties kunnen gebruikt worden.

Protocollen

Momenteel worden er 6 protocollen binnen JXTA gedefinieerd:

Het eerste is het *Peer Discovery Protocol*, een protocol dat het mogelijk maakt voor een peer om de advertenties op een andere peer te vinden. Het protocol kan gebruikt worden om elke peer, peergroep of advertentie te lokaliseren. Het Peer Discovery Protocol is een standaard groep voor alle peergroepen inclusief de wereldpeergroep en zorgt ervoor dat alle JXTA peers elkaar kunnen verstaan op het meest elementaire niveau.

Ten tweede is er het Peer Resolver Protocol dat het mogelijk maakt voor een peer om generieke zoekopdrachten te verzenden en ontvangen om peers, peergroepen, pijplijnen en andere informatie respectievelijk te zoeken en vinden. Gewoonlijk zal dit protocol enkel geïmplementeerd worden op peers die toegang hebben tot dataopslag en dus geavanceerde zoekmogelijkheden aanbieden.

Het *Peer Information Protocol* stelt peers in staat informatie in te winnen over de mogelijkheden en statussen van andere peers. Een peer kan bijvoorbeeld een andere peer pingen om te zien of die beschikbaar is.

Als vierde protocol is er het *Rendezvous Protocol* dat het mogelijk maakt een bericht te propageren binnen het bereik van een peergroep.

Het *Pipe Binding Protocol* zorgt er dan weer voor dat een peer een advertentie aan een eindpunt van een pijplijn kan verbinden waarmee kan aangeduid worden waar een bericht over de pijplijn verzonden kan worden. Een pijplijn kan gezien worden als een abstracte berichtwachtlijn die enkele abstracte operaties zoals creëer, open, sluit, zend en ontvang ondersteunt. De binding gebeurt bij het openen van een pijplijn, wanneer een pijplijn gesloten wordt zal de advertentie terug ontbonden worden.

Het zesde en laatste protocol is het *Endpoint Routing protocol*. Dit protocol zal de beschikbare routes beheren voor het verzenden van een bericht naar een doelpeer. In vele gevallen zullen twee communicerende peers niet rechtstreeks met elkaar verbonden zijn. Wanneer bijvoorbeeld twee peers niet hetzelfde netwerkprotocol gebruiken, of als de communicerende partners zich aan weerszijden van een firewall of NAT bevinden. Peerrouters beantwoorden de aanvragen met beschikbare routinginformatie, die bestaat uit een lijst van gatewayadressen langs de route. Elke peer kan beslissen een peerrouter te worden door het Endpoint Routing Protocol te implementeren.

Beveiliging

Het beveiligen van een P2P-systeem is zeer vergelijkbaar met dat van elke ander computersysteem. De drie dominante vereisten zijn de vertrouwelijkheid, integriteit en beschikbaarheid van het systeem. Deze uit zich in zeer specifieke functionaliteitsvereisten die ondermeer authenticatie, toegangscontrole, audit, encryptie en beveiligde communicatie omvatten.

De JXTA Java-binding die voor de demoapplicatie zal worden gebruikt bevat zelf een groot aantal van deze ingebouwde beveiligingsfuncties die kunnen bijdragen aan de veiligheid van de toepassingen die ermee ontwikkeld worden. Hoewel deze geenszins voldoende zijn voor het bouwen van een waterdichte applicatie, kunnen ze mogelijk een basis vormen waarop beveiligde JXTA-toepassingen (20) kunnen gebouwd worden.

Momenteel worden de volgende beveiligingsfuncties door het JXTA-platform ondersteund:

- Transport Layer Security¹³ (TLS), ook bekend als Secure Sockets Layer (SSL) V3.1, is gebaseerd op een algoritme met publieke sleutel. Het JXTA-platform biedt TLS aan als medium voor veilige communicatie. Toepassingen kunnen de TLS-functionaliteit van het platform benutten door het gebruik van beveiligde pijplijnen (secure pipes), die door intern gebruik van TLS beveiliging garanderen tegen passieve aanvallen.
- Peercertificaten zijn in de TLS-laag vereist om de werking ervan te garanderen. Bijgevolg genereert elke peer zijn eigen certificaat en fungeert als zijn eigen Certification Authority (CA). Dit certificaat, het zogenaamde hoofdcertificaat (root certificate), wordt gebruikt om de dienstcertificaten (service certificates) te ondertekenen voor elke dienst die de peer ondersteunt. Het hoofdcertificaat wordt samen met de peeradvertentie gedistribueerd. Daarom kan elke andere peer altijd controleren of een advertentie wel degelijk van de peer komt die beweert deze te hebben afgegeven.
- Elke peer is beschermd door een peer-ID en wachtwoord. Deze worden gebruikt om de private sleutel van een beveiligde gebruiksomgeving te decoderen. Deze versleuteling fungeert als een eerste lijn van verdediging tegen een lokale aanvaller, d.i. een aanvaller met fysieke toegang tot de machine die de JXTA-peer uitvoert.

4.3 IMPLEMENTATIE

Doordat de meeste van de basiselementen van een P2P-netwerk reeds in de JXTA-protocolset geïmplementeerd zijn, zal in weinig basisfunctionaliteit zelf moeten worden voorzien. Dat JXTA ook op het programmeer-technisch gebied de correcte keuze was, blijkt al snel uit de efficiëntie van de code die geproduceerd werd bij het ontwerp van de demoapplicatie. Met een minimum

¹³ <http://tools.ietf.org/html/rfc4346>

aan programmacode kan een applicatie met behoorlijke functionaliteit uitgebouwd worden zonder een teveel aan ontwikkelingsvrijheid te moeten inleveren.

Hieronder wordt de specifieke implementatie van de basisstappen die beschreven werden in de specificatie¹⁴ ingevuld. Bij elk van de benodigde operaties wordt uiteengezet door welke functionaliteit in de JXTA Javabindingen¹⁵ (JXSE) die kan worden geïmplementeerd. Deze aantekeningen kunnen dan worden gebruikt bij de uiteindelijke implementatie van de basisfunctionaliteit in een P2P protocol voor het stromen van multimedia.

Enkele codefragmenten uit de implementatie worden weergegeven ter bespreking. Het is hierbij niet de bedoeling dat een tutorial voor de JXSE API neer te schrijven. Basiscode zoals het opstarten van de JXTA-omgeving en het JXTA-CMS, etc. komt dus niet aan bod. Alleen bij fragmenten waar de protocolspecifieke code wordt geproduceerd, zal tekst en uitleg worden voorzien.

4.3.1 AANMELDEN

Zoals in de specificatie reeds werd aangegeven zal een peer verbinding maken met het P2P-distributienetwerk door lid te worden van een subnetwerk, d.i. een peergroep. In het globaal register kan worden nagegaan of er een peergroep bestaat die gedefinieerd wordt door een gelijkaardige stroomopdracht. Bij een JXTA-toepassing zal deze taak door de overkoepelende wereldpeergroep op zich worden genomen. De selectie van een peergroep binnen de JXTA-omgeving gebeurt op basis van de peergroepsnaam.

De peergroepsnamen moeten dus op een gestandaardiseerde wijze worden bepaald. In de implementatie van de proof-of-concept is er uitgegaan dat alle peers geïnteresseerd waren in de multimedia uit eenzelfde multimediatebibliotheek, d.i. de totale multimediateinhoud in één bepaalde peergroep of van één bepaalde distributieserver. Bijgevolg wordt de peerverdeling in peergroepen enkel bepaald door het tijdstip van aanmelden voor het stromen van multimedia. Wanneer echter al op voorhand bijvoorbeeld zou kunnen worden gespecificeerd in welke bibliotheek, of in grote lijnen rond welk onderwerp of welk type multimedia zal worden gezocht, kan met al deze parameters rekening worden gehouden bij het zoeken naar een gepaste peergroep. Zo zal deze meer op maat van de specifieke noden van de gebruiker kunnen gekozen worden. Aangezien een peergroep op peergroepsnaam wordt gezocht is de eenvoudigste manier om al deze aspecten in de naam te weerspiegelen en bijgevolg een serialisatiefunctie te voorzien die gegeven een aantal parameters een gestandaardiseerde naam kan genereren. De eenvoudigste manier om dit te doen is een samenstelling van de parameters aan een hashfunctie te voeren die hieruit een unieke representatie van een vaste lengte produceert. Een gebruiker zal bijvoorbeeld filmtrailers en multimedia (trivia, afbeeldingen, etc.) over nieuwste producties willen bekijken die gepubliceerd zijn door de Internet Movie DataBase¹⁶ (IMDB) website. In dit geval kan de peer ingedeeld worden in een peergroep met peers die geïnteresseerd zijn in informatie van dezelfde leverancier en die op een gelijktijdig tijdstip zich aanmeldden. De naam van de peergroep kan worden gegenereerd uit de naam van de informatieleverancier in combinatie met het tijdstip. Peers die dezelfde multimedia binnen eenzelfde tijdsinterval opvragen zullen zo zonder twijfel in dezelfde groep ingedeeld worden voor de bestandsuitwisselingen.

¹⁴ Zie 3.3 Specificatie

¹⁵ <https://jxta-jxse.dev.java.net>

¹⁶ <http://www.imdb.com>

Zoeken van een peergroep

Het zoeken van een peergroep is geïmplementeerd in de `peerGroupSearch()` weergegeven in *code 4.1*. De zoekfunctie zal de opgegeven groepsnaam als zoekparameter voor de Peer Discovery Service gebruiken om zo na te gaan of een dergelijke groep reeds is geadverteerd in de wereldpeergroep. Om de kans op het vinden van een peergroep in het globale register te maximaliseren, zal de zoekopdracht een opeenvolgend aantal keer uitgevoerd worden. Wanneer na een eindig aantal pogingen de groep niet gevonden is, wordt deze als onbestaand beschouwd en moet die dus worden aangemaakt.

```
private void peerGroupSearch(String groupName) {
    Enumeration adv = null;
    int count = 1;
    System.out.println("Searching for group advertisements");
    System.out.println("-----");
    boolean found = false;
    //Try 10 times to find a group before making a new one
    int times = 3;
    while (count <= times && !found) {
        try {
            System.out.println("Try Number: " + count);
            //Searching for group advertisements
            //First from local cache
            adv = globalDiscoveryService.getLocalAdvertisements(DiscoveryService.GROUP, "Name", groupName);
            if ((adv != null) && adv.hasMoreElements()) {
                System.out.println("Group found in local advertisement.");
                peerGroupAdv = (PeerGroupAdvertisement) adv.nextElement();
                localPeerGroup = netPeerGroup.newGroup(peerGroupAdv);
                joinPeerGroup(localPeerGroup);
                found = true;
            } //If not found then search from remote peers
            else {
                System.out.println("No group found in local advertisement.");
                System.out.println("Starting remote search...");
                globalDiscoveryService.getRemoteAdvertisements(null, DiscoveryService.GROUP, "Name", "group", 1);
            }
            Thread.sleep(timeout);
            //If group not found after couple of tries, create the group
            if ((count == times) && (adv == null || !adv.hasMoreElements())) {
                System.out.println("No group found, creating group.");
                localPeerGroup = createPeerGroup(groupName);
                joinPeerGroup(localPeerGroup);
                found = true;
            }
        } catch (IOException e) {
            e.printStackTrace();
        } catch (PeerGroupException e) {
            System.out.println("ERROR:" + e.getMessage());
            System.exit(-1);
        } catch (InterruptedException e) {
            System.out.println("ERROR:" + e.getMessage());
        }
        count++;
    }
}
```

Code 4.1: Implementatie van een zoekmechanisme voor peergroepen.

Creëren van een nieuwe peergroep

Wanneer de gezochte peergroep nog niet bestaat, zal die moeten worden aangemaakt. Elke JXTA-peer kan een nieuwe peergroep aanmaken. *Code 4.2* bevat de implementatie van een functie voor het aanmaken van een nieuwe peergroep. Vooraleer een groep kan worden aangemaakt moet een unieke identificator gegenereerd worden. In de implementatie van het protocol gebeurt dit op basis van de verkregen groepsnaam. Nadat een advertentie voor de groep is verkregen, kan in combinatie met de groepidentificator en de groepsnaam een nieuwe peergroep aangemaakt worden. Nu de groep is aangemaakt zal ze nog moeten geadverteerd worden. De Peer Discovery Service kan ervoor zorgen dat de peergroep wordt geregistreerd in de catalogoog van de wereldpeergroep. Een laatste actie die moet gebeuren bij het aanmaken van een nieuwe peergroep is ervoor zorgen dat de peer die deze actie ondernam ook als superknoop, of rendezvouspeer zoals daar in het JXTA-jargon naartoe wordt verwezen, fungeert.

```

public PeerGroup peergroupCreate(String groupName) {
    PeerGroup peerGroup = null;
    try {
        //Generating groupID
        PeerGroupID groupID = IDFactory.newPeerGroupID(groupName.getBytes());
        // Retrieving a module implementation advertisement
        ModuleImplAdvertisement moduleImplAdv = netPeerGroup.getAllPurposePeerGroupImplAdvertisement();
        peerGroup = netPeerGroup.newGroup(groupID, moduleImplAdv, groupName,
            "Streaming media peergroup");

        // Starting peer group
        if (Module.START_OK == peerGroup.startApp(new String[0])) {
            //Publishing new group advertisements
            peerGroupAdv = peerGroup.getPeerGroupAdvertisement();
            globalDiscoveryService.publish(peerGroupAdv);
            globalDiscoveryService.remotePublish(peerGroupAdv);
            peerGroup.getRendezVousService().startRendezVous();
            System.out.println("New peer group started successfully you are rendezvous!!!");
        } else {
            System.out.println("ERROR: New peer group not started !!!");
        }
    } catch (Exception e) {
        System.out.println("ERROR:" + e.getMessage());
        System.exit(-1);
    }
    return peerGroup;
}

```

Code 4.2: Implementatie van een functie voor het creëren van een nieuwe peergroep.

Lidmaatschap van een peergroep

Eens de gewenste peergroep is gelokaliseerd of nadat deze is aangemaakt, wil een peer uiteraard deel uitmaken van deze peergroep. *Code 4.3* bevat de *peerGroupJoin()* functie waarmee een opgegeven peer tot een subnetwerk kan toetreden. Allereerst moeten de credenties van de peer aangemaakt worden zodat de Peer Membership Service deze kan voorleggen aan de peergroep. Als deze door de peergroep geauthenticeerd worden, kan een peer de peergroep vervoegen. Eens de peer lid is van de groep zal hij zichzelf bekendmaken aan de andere peers en de rendezvouspeer door zijn peeradvertentie met respectievelijk de *publish()* en de *remotePublish()* functies aan de lokale Discovery Service te melden. Voor een optimale werking van de demoapplicatie wordt de advertentie ook bij de globale Discovery Service gepubliceerd. Dit is echter overbodig bij een eigen implementatie die met meer dan één peergroep werkt.

```

public void peerGroupJoin(PeerGroup peerGroup) {
    StructuredDocument creds = null;
    System.out.println("Joining group " + peerGroup.getPeerGroupName() + "...");

    //Authenticate and join to group
    try {
        AuthenticationCredential authCred = new AuthenticationCredential(peerGroup, null, creds);
        MembershipService membership = peerGroup.getMembershipService();
        Authenticator auth = membership.apply(authCred);
        if (auth.isReadyForJoin()) {
            Credential myCred = membership.join(auth);
            System.out.println("Group Details:");
            StructuredTextDocument doc = (StructuredTextDocument) myCred.getDocument(new MediaType("text/plain"));
            StringWriter out = new StringWriter();
            doc.sendToWriter(out);
            System.out.println(out.toString());
            System.out.println(peerGroup.getPeerName() + " now online");
            System.out.println("Getting group services.");
            //Publishing Peer Advertisements.
            localDiscoveryService = peerGroup.getDiscoveryService();
            System.out.println("Publishing peer advertisement.");
            localDiscoveryService.publish(peerGroup.getPeerAdvertisement());
            localDiscoveryService.remotePublish(peerGroup.getPeerAdvertisement());
            globalDiscoveryService.publish(peerGroup.getPeerAdvertisement());
            globalDiscoveryService.remotePublish(peerGroup.getPeerAdvertisement());
        } else {
            System.out.println("ERROR: Cannot join group!");
            System.exit(-1);
        }
    } catch (Exception e) {
        System.out.println("ERROR: " + e.getMessage());
        System.exit(-1);
    }
}

```

Code 4.3: Implementatie van een functie voor het toetreden tot een peergroep.

4.3.2 DELEN

Voor het delen van bestanden op het P2P-netwerk wordt het standaard JXTA Content Management System¹⁷ (JXTA-CMS) gebruikt. Het CMS is een eenvoudige service die bovenop andere JXTA-diensten, zoals de Peer Resolver Service en het Endpoint Routing protocol, is gebouwd en waarmee bestanden kunnen worden gedeeld en gezocht binnen een peergroep.

In de implementatie, weergegeven in codefragment *Code 4.4*, van de functionaliteit voor het delen van bestanden zal ervoor gekozen worden alle inhoud in een opgegeven bestandsmap te delen. Het is de bedoeling dat deze map alle te delen multimedia én de bijhorende metadata bevat. De code beschrijft hoe alle inhoud van de map wordt opgesomd waarna alle bestanden op de lijst één per één worden gedeeld. Bij het delen van de bestanden worden deze voorzien van een checksum, zodat na het downloaden de integriteit van het bestand kan worden gecontroleerd.

```
public void share() {  
  
    try {  
        //Sharing all files in shared directory  
        ContentManager contentManager = null;  
        contentManager = cms.getContentManager();  
  
        File[] list = sharedDir.listFiles();  
        CRC32 checksum = new CRC32();  
  
        for (int i = 0; i < list.length; i++) {  
            //Sharing Files and check sums in network  
            if (list[i].isFile()) {  
                contentManager.share(list[i], checksum.getChecksum(list[i]));  
            }  
        }  
  
        System.out.println("Shared Contents:");  
        System.out.println("*****");  
        //Viewing the shared contents  
        Content[] content = cms.getContentManager().getContent();  
        //Show the share contents in log area  
        for (int j = 0; j < content.length; j++) {  
            System.out.println("* " + content[j].getContentAdvertisement().getName() + "\tSum: " +  
                content[j].getContentAdvertisement().getDescription() + "\n");  
        }  
  
        System.out.println("Content successfully shared.");  
    } catch (IOException e) {  
        System.out.println("EXCEPTION: " + e.getMessage() + "\nMake sure file: \"Shares.ser\" is deleted before" +  
            " the start of the service.\n");  
    }  
}
```

Code 4.4: Functie voor het delen van data in een peergroep.

4.3.3 ZOEKEN

Het zoeken van multimediafragmenten zal ook met het JXTA-CMS gebeuren. Aangezien de standaard ondersteunde zoekfunctionaliteit voor metadata niet voldoende diepgang biedt, zal een uitbreiding hiervan moeten ontworpen worden. De details van het ontwerp en implementatie van de uitgebreide zoekarchitectuur staan uitgebreid uitgeschreven in het tweede deel van dit werk.

4.3.4 BESTANDSOVERDRACHT

Voor het transport van de multimedia wordt wederom beroep gedaan op het JXTA-CMS. Uit de advertenties van bestanden die aan de zoekparameters van een zoekopdracht voldoen, kan de CMS alle informatie halen die nodig is voor het transporteren van de bestanden en de bijhorende metadata naar de zoekende peer.

¹⁷ <https://jxse-cms.dev.java.net>

Hoewel het CMS intern de getransporteerde data zelf opdeelt in stukjes om een optimale overdracht te verzekeren, wordt er van uitgegaan dat de multimedia al is verdeeld in weergeefbare segmenten die voldoende geannoteerd zijn. Dergelijke aanpak vereenvoudigt de bestandsoverdracht aanzienlijk. Op deze wijze kan de bestandsselectie en volgorde voor de overdracht buiten de CMS om vastgelegd worden zonder er iets intern aan de API gewijzigd moet worden. De gepubliceerde multimediabestanden waarover tot hiertoe werd gesproken, worden dus eigenlijk steeds in segmentvorm gedeeld. De metadata bij elk segment bevat alle informatie om de oorspronkelijke multimediabestanden te reconstrueren. In de annotatie wordt beschreven tot welk bestand het segment behoort en waar in het bestand het segment gepositioneerd is. Al zal zelden een volledig multimediabestand gereconstrueerd moeten worden, het kan wel gebeuren dat twee of meer opeenvolgende fragmenten aan een zoekopdracht voldoen en dan moeten deze bij voorkeur na elkaar weergegeven worden.

Wanneer er sprake is van meerdere bronnen voor eenzelfde mediasegment, zal het CMS de verschillende bronnen combineren om zo de bandbreedte voor de overdracht te optimaliseren. Verder voorziet het CMS ook de routing van de dataoverdracht. Daar bij een applicatie voor het stromen van data de padselectie niet zal verschillen van deze bij een standaard P2P-toepassing heeft het ook weinig zin om zelf deze beslissingen te overschrijven.

De herorderingsbuffer wordt geïmplementeerd aan de hand van een systeem van lijsten. Ieder gedownload segment wordt tegen een lijst van te downloaden multimediafragmenten getoetst. Telkens als het eerste element op de downloadlijst afgehaald is, kan het van de lijst worden geschrapt en wordt dit bestand aan een afspeellijst toegevoegd. Dit proces zal zich herhalen tot de volledige downloadlijst is afgewerkt. Doordat ervoor gezorgd wordt dat beide lijsten in dezelfde volgorde worden samengesteld, zal de sequentie van de bufferinput ten opzichte van de bufferoutput gerespecteerd blijven.

4.3.5 WEERGAVE

De herorderingsbuffer, die overigens ook dienst doet als weergavebuffer, bestaat dus zoals hierboven beschreven uit een afspeellijst. Wanneer er zich minder dan een vooraf gedefinieerd aantal multimediasegmenten in de afspeellijst bevindt, zal de weergave onderbroken worden.

Voor elk type multimedia dat in de afspeellijst voorkomt, moet dus een decodering voorzien zijn in de weergavefunctionaliteit van de P2P-applicatie. Om deze functionaliteit in de demoapplicatie te implementeren, wordt een open source Javabinding van VLC gebruikt. VLCJ¹⁸ is een alternatief voor de VideoLAN¹⁹ Javabinding die het op een eenvoudigere manier mogelijk maakt een native VLC mediaspeler in een Java-applicatie te integreren. De keuze voor VLC valt te verdedigen vanuit het feit dat de mediaspeler de meeste afbeelding-, muziek- en filmbestanden zonder extra externe decodering kan weergeven. Op die manier kan een maximum aan multimediatypes weergegeven worden zonder een te complexe weergavefunctionaliteit te moeten voorzien. De applicatie is en blijft een proof-of-concept en een grafische weergave is niets meer dan een leuke bonus.

¹⁸ <http://code.google.com/p/vlcj>

¹⁹ <http://www.videolan.org>

DEEL II

PERSONALISERING VAN MULTIMEDIA VOOR EEN MULTIMEDIA STREAMING PROTOCOL

HOOFDSTUK 5

PERSONALISERING VAN MULTIMEDIA: STATE OF THE ART

Bij het personaliseren van multimedia aan de hand van metadata-informatie moet rekening gehouden worden met de twee aspecten die in dit probleem verwerkt zitten. Allereerst moet worden nagegaan uit welke metadata-omschrijving het best een multimediapresentatie op maat kan worden gemaakt. De beschrijvende eigenschappen, zowel op inhoudsomschrijvend als technisch vlak, van de woordenschat moeten met andere woorden voorzichtig worden afgewogen.

Daarenboven moet een multimediabewerkingomgeving gevonden worden die op een automatische wijze, aan de hand van de vastgelegde metadata-annotaties, gepersonaliseerde presentaties uit een selectie van de geannoteerde multimedia kan creëren.

5.1 METADATA-ANNOTATIES VOOR MULTIMEDIA

5.1.1 OVERZICHT VAN BESTAANDE STANDAARDEN

Er bestaan een aantal metadatastandaarden die tot op een bepaald niveau gebruikt kunnen worden voor het annoteren van multimediabestanden. Hieronder wordt een kort overzicht gegeven van enkele standaarden die (gedeeltelijk) in deze functionaliteit voorzien. Het is geenszins de bedoeling om een volledig overzicht van alle metadatastandaarden die mogelijks voor deze taak kunnen worden ingezet op te leveren, maar het is de intentie om een beeld te schetsen van enkele mogelijkheden die een applicatie heeft voor het bijhouden van meta-informatie bij zijn multimedia.

- *MPEG-7* (21) wil informatie voorstellen over de multimedia in plaats van de multimedia zelf. De standaard poogt hierbij het toepassingsdomein zo breed mogelijk te houden. Dit resulteert in een uitgebreide annotatie waarin een aantal domeinen, variërend van coderingsschemadescriptoren op laag niveau tot inhoudsdescriptoren op een hoog niveau worden samengevoegd. Omdat er niet wordt uitgegaan van een expliciete toepassingsdomein ondersteunt MPEG-7 meerdere manieren om de annotaties te structureren. Daarnaast kunnen van het toepassingsdomein afhankelijke descriptoren worden toegevoegd als de standaard descriptoren niet voldoende detail bieden.
- De *Dublin Core Element Set*²⁰ is een standaard voor een domeinoverschrijdende informatiebronbeschrijving. Er zijn geen fundamentele beperkingen aan de soorten bronnen waaraan de Dublin Core metadata kunnen worden toegewezen. De Dublin Core is een veelgebruikte annotatiestandaard over diverse domeinen. De standaard omvat een klein aantal relaties, die door deskundigen op het gebied van digitale bibliotheken zijn vastgelegd.
- *MIME*²¹ werd dan weer oorspronkelijk ontworpen om het voor e-mailtoepassingen mogelijk te maken het mediatype van e-mailinhoud te identificeren. Desalniettemin wordt het in tal van hedendaagse applicaties gebruikt zoals door webbrowsers en vele andere multimediatoepassingen.

²⁰ <http://dublincore.org/documents/dces>

²¹ http://www.w3schools.com/media/media_mimeref.asp

- *RDF* werd door het World Wide Web Consortium (W3C)²² oorspronkelijk ontworpen als een algemeen metadata datamodel. Het RDF datamodel is vergelijkbaar met klassieke conceptuele modelleringen zoals Entity-Relationship- of klasse-diagrammen. Aangezien het model gebaseerd is op het idee van het maken van uitspraken over bronnen (oorspronkelijk webbronnen) in de vorm van subject-predikaat-object uitdrukkingen of triples, kunnen relaties tussen het subject en het object gedefinieerd worden.

5.1.2 PROBLEMEN MET BESTAANDE STANDAARDEN

Er zijn dus verscheidene standaarden voorhanden waarmee multimedia op de een of andere manier kan geannoteerd worden. Het probleem echter met deze voorstellen is dat geen van hen een sluitend antwoord kan bieden op de drie courante problemen hieronder beschreven.

Probleem van specifieke annotaties gebonden aan het mediatype

Bij de algemene beschrijvingen die aan de totale inhoud van een multimediabestand worden toegekend, zullen er nog steeds een aantal typegebonden problemen de kop op steken. Allereerst zal het multimediatype dus moeten worden bepaald alvorens het multimediabestand door het metadata-systeem kan worden verwerkt. Een eenvoudige en veelgebruikte bibliotheek voor het identificeren van het type media is MIME. Andere kenmerken van de multimedia kunnen immers afhangen van het mediatype van het bestand. Zo zal bijvoorbeeld de resolutie van een multimediabestand een relevant kenmerk zijn voor foto- of filmbestanden, maar geen steek houden in de context van een geluidsbestand.

Ook kan de semantiek van een waarde of een eigenschap afhangen van het type media dat wordt beschreven. Een boek zal een auteur hebben, maar bij een film is het een regisseur die de het verhaal vertelt. Wanneer deze beiden bijvoorbeeld geannoteerd worden door het *dc:creator* veld kan hierbij verwarring ontstaan. Daarom hebben vele mediatypes reeds een eigen set van metadata-attributen ontwikkeld. Voor de annotatie van MP3 worden ID3²³ tags gebruikt en JFIF²⁴ voegt informatie toe aan JPEG bestanden. Deze verschillende standaarden verzamelen, laat staan aggregeren in één enkele metadata-annotatie is onbegonnen werk.

Probleem van specifieke beschrijving van de multimedia-inhoud

Vanuit het cliché “een foto zegt meer dan duizend woorden” kan al gemakkelijk afgeleid worden dat het beschrijven van multimedia aan de hand van een gestandaardiseerde bibliotheek vaak zal leiden tot lijkige en complexe beschrijvingen. Daarenboven zal weinig objectiviteit in de beschrijving gelegd zijn en zullen de gekozen annotaties met andere woorden sterk afhankelijk zijn van de persoon die ze heeft opgesteld.

De complexiteit en lijkigheid van de annotaties zijn eigenschappen die ook worden gereflecteerd in de beschrijvende bibliotheken. De MPEG-7 specificatie die werd ontworpen voor het opstellen van meta-informatie bij multimedia mag dan wel geschikt zijn voor het voorzien van dergelijke metadata bij een groot bereik aan mediatypes, toch gaat dit ten koste van de bondigheid en simpliciteit van de annotatie. Deze eigenschappen maken de standaard ook een stuk minder aantrekkelijk voor potentiële gebruikers en zullen op die manier de verspreiding ervan in de weg staan.

²² <http://www.w3.org>

²³ <http://www.id3.org/id3v2.3.0>

²⁴ <http://www.jpeg.org>

Probleem van de granulariteit van de annotaties

Het granulariteitsprobleem bij multimedia-annotatie houdt in dat een gebruiker in staat is te specificeren aan welk onderdeel van een multimediatebestand een bepaalde annotatie wordt verbonden. Wanneer bijvoorbeeld moet worden uitgedrukt dat de laatste drie minuten van een videobestand de aftiteling bevatten, zal deze annotatie uiteraard enkel mogen van toepassing zijn op de laatste drie minuten van het multimediatebestand.

Er zijn drie basistechnieken die met dit probleem komaf kunnen maken. Een eerste oplossing zal de verantwoordelijkheid bij het multimediatebestand zelf leggen. In dit geval wordt ervan uitgegaan dat in de fysieke opslag van het multimediatebestand een aantal zogehete “haken” (hooks) worden voorzien waaraan de metadata kan worden opgehangen. Dit betekent echter dat de gebruiker toegang moet hebben tot de fysieke representatie van de multimedia en tags moet kunnen toevoegen aan de codering van het multimediatebestand. Dergelijke technieken, die wijzigingen aanbrengen aan de fysieke opslag van een multimediatebestand, werden van het begin af in deze masterproef afgezworen.

Een tweede oplossing voor het granulariteitsprobleem zal het probleem verschuiven van de multimediatebestanden naar de bijhorende annotaties. In dit geval moet de metadata niet enkel in staat zijn de inhoud van een bestand te beschrijven maar ook kunnen situeren op welk deel van het bestand de annotaties betrekking hebben. Deze benadering wordt onder andere toegepast bij MPEG-7.

De derde aanpak lost het probleem op tijdens de constructie van links die een annotatie aan het doelbestand binden. Zo worden de benodigde uitdrukkingen verschoven van het annotatieschema naar de taal die de links beschrijft. Deze benadering wordt gebruikt bij het vastleggen van semantische relaties op het semantisch web, i.c. RDF. Fragmenten van een multimediatebestand kunnen worden geadresseerd door een zogenoemde fragmentidentificator (fragment identifier). Deze wordt aangeduid als het deel dat achter het ‘#’ komt in de URI. De betekenis van een fragmentidentificator hangt af van het MIME type van het doelbestand en vele types bevatten nog steeds geen dergelijke definitie.

5.2 GEAUTOMATISEERDE MULTIMEDIA PRODUCTIE

Idealiter zou een bewerkingsomgeving voor multimedia gevonden moeten worden die automatisch, aan de hand van de metadata-annotaties bij de mediabestanden, een aangepaste samenstelling van mediafragmenten kan maken die voldoet aan de eisen van een gebruiker.

In de eerste plaats zou dan moeten uitgekeken worden naar een platform dat een op voorhand vastgelegde metadatastandaard moet ondersteunen. Omdat uit de state of the art van de metadata-annotaties reeds blijkt dat geen enkel bestaande annotatie voldoet voor toepassing in een dergelijke applicatie, kan daaruit worden afgeleid dat een bruikbaar ontwerp van zo een systeem nog in ontwikkeling is.

Het zelf ontwikkelen van een geautomatiseerde bewerkingsomgeving voor multimedia is stof voor een masterproef op zich en zal niet tot het onderzoek bij dit werk behoren. Uiteraard zal dergelijke functionaliteit wel elementair zijn in de specificatie van een P2P-systeem dat gepersonaliseerde multimediatepresentaties wil opstellen. Bijgevolg zal in het specificatie- en implementatiehoofdstuk voor een eenvoudiger bewerkingsraamwerk gezorgd worden dat enkel

basisfunctionaliteit voor een geautomatiseerde bewerkingsomgeving verzorgt. Op die wijze kan alsnog tot een werkend protocol gekomen worden.

HOOFDSTUK 6

PERSONALISERING VAN MULTIMEDIA: SPECIFICATIE

Nadat in het vorige hoofdstuk een overzicht werd gegeven van enkele bestaande metadatastandaarden en waarom deze tekort kwamen voor het annoteren van multimedia, kan in dit hoofdstuk in de eerste plaats worden onderzocht welke kwaliteiten de metadata-annotatie wel zou moeten weerspiegelen.

Op het vlak van multimediabewerkingen werd reeds begrepen dat de geautomatiseerde multimediatechnologieomgeving nog in volle ontwikkeling zijn. Om de proof-of-concept alsnog te verwezenlijken, zal dus in dit hoofdstuk enkel de broodnodige functionaliteit worden gedistilleerd voor het implementeren van een dergelijke basisomgeving.

6.1 VOOROPGESTELDE VEREISTEN

Alvorens aan een specificatie van een mechanisme voor het personaliseren van multimedia kan begonnen worden, moeten eerst de vereisten voor de basisonderdelen ervan worden afgelijnd. De geautomatiseerde personalisatieomgeving bevat immers twee belangrijke componenten. Allereerst is er een de component die de metadata bij de multimedia moet verwerken zodat de gepaste multimedia kan geselecteerd worden. Daarnaast heb je de component die zorgt voor de fysieke creatie van de multimediatechnologie met de geselecteerde componenten.

6.1.1 VEREISTEN VOOR DE METADATA-ANNOTATIE

Uit de bevindingen neergeschreven in het state of the art hoofdstuk kunnen enkele basiseigenschappen worden afgeleid waaraan een bruikbare omgeving voor multimedia-annotatie moet voldoen (3).

Minimaal en uitbreidbaar

Allereerst moet de specificatie minimaal en uitbreidbaar zijn. De omvang en complexiteit van benaderingen zoals MPEG-7 (21) zijn de belangrijkste struikelblokken die een bredere verspreiding van de standaard in de weg staan. Zelfs commercieel ontwikkelde producten zoals AAT²⁵ en IconClass²⁶ vergen een intensieve opleiding voordat ze op een efficiënte manier kunnen gebruikt worden. Voor veel toepassingen is een minimalistische benadering via een kernwoordenschat met een klein aantal vaak gebruikte multimedia-annotaties een veel gepastere aanpak. Deze kern moet echter voldoende flexibel zijn om te dienen als basis voor meer gedetailleerde, modulaire uitbreidingen. Een dergelijke modulaire aanpak zorgt niet alleen voor een minimalistische kern, maar ook voor minimalistische uitbreidingsmodules die een zeer specifieke aanvulling op de kern kunnen betekenen.

Hergebruik van bestaande standaarden

Voor de meest gebruikte eigenschappen is er gegarandeerd reeds een woordenschat die het attribuut bevat. De kern van de annotatiestandaard moet hergebruik van bestaande standaarden mogelijk maken in plaats van er nog een extra nieuwe te definiëren. Door te kiezen voor de

²⁵ De Art & Architecture thesaurus (AAT) is een gestructureerde bibliotheek bestaande uit ongeveer 34.000 begrippen. Hierin zijn 131.000 termen, beschrijvingen, bibliografische referenties en andere informatie met betrekking tot de beeldende kunst, architectuur, decoratieve kunsten, archiefmateriaal, en materiële cultuur vervat.

²⁶ Iconclass is een classificatiesysteem voor kunst en iconografie ontworpen aan Rijksbureau voor Kunsthistorische Documentatie in Den Haag.

integratie van de verschillende standaarden in een minimalistische kern, zullen er onvermijdelijk bepaalde eigenschappen in meerdere standaarden gedefinieerd zijn. Zo wordt de titel in de Dublin Core geannoteerd met *dc:title* en in MPEG-7 met *mpeg7:Title*. Met deze dubbelzinnige definitie kan op verschillende manieren worden omgegaan. Er kan bijvoorbeeld een hiërarchie onder de verschillende standaarden gedefinieerd worden. Bij een dubbele annotatie kan dan geopteerd worden voor de standaard die het hoogst in de hiërarchie geplaatst is. Een andere mogelijkheid is om vertalingsschemata tussen beide standaarden vast te leggen.

Verbinding van concepten aan de multimedia

De relatie tussen een concept en de multimedia moet expliciet worden uitgedrukt en vastgelegd op een interoperabele manier. De aard van deze relatie is afhankelijk van het type media en moet in twee richtingen werken: mediacentrische toepassingen moeten in staat zijn om bijbehorende annotaties te vinden, maar kennisgebaseerde toepassingen moeten tegelijk de mogelijkheid bieden om de media te identificeren aan de hand van de annotaties.

Gestructureerde annotaties

De standaard moet willekeurige groeperingen van annotaties toelaten, zodat de structuur van de multimedia ook kan worden gebruikt om de metadata te organiseren. Een expliciete definitie van de annotatiestructuur in een schema zou applicaties in staat stellen annotaties te valideren of kan zelfs leiden tot het ontwerp van een generieke annotatie-applicatie.

Vrij van licenties

Op zijn minst de kern van de samengestelde metadatabibliotheek moet vrij zijn van licenties die de vrije verspreiding van de toepassing ervan in de weg zouden staan.

Technische specificatie van multimedia

De technische specificaties van een multimediabestand moeten expliciet worden vermeld voor het geval de context, i.c. de gebruiker of het weergavemedium, bepaalde technische vereisten of beperkingen aan de media wil opleggen.

Het is echter niet realistisch, zoals uit het vorig hoofdstuk²⁷ blijkt, om een reeds bestaande metadatastandaard te vinden die aan alle bovenstaande vereisten voldoet. Daarom dat juist de tweede vereiste zo elementair is. De voorziene metadata bij de multimedia zal dus inderdaad geannoteerd worden door middel van een combinatie van verschillende standaarden.

6.1.2 VEREISTEN VOOR DE PERSONALISATIE VAN DE MULTIMEDIA

Ook het personaliseren van presentaties zal aan enkele voorschriften moeten voldoen. Aangezien geen voorgestelde raamwerk voor implementatie in het state of the art hoofdstuk gevonden werd, moeten enkele beperkingen voor een dergelijk systeem opgeworpen worden waarmee rekening moet worden bij het implementeren van een meer rudimentair mechanisme in het volgende hoofdstuk.

Integriteit van de fysieke opslag

Een eerste vereiste voor de personalisatie van de multimedia is dat niet geraakt wordt aan fysieke opslag van de gedeelde multimedia in de multimediate bibliotheek van het netwerk. Deze vereiste wordt bewust aan het systeem opgelegd om te vermijden dat hetzelfde multimediabestand onder verschillende fragmentsamenstellingen in het netwerk wordt

²⁷ Zie 5.1.2 Problemen met bestaande standaarden

aangeboden. Wanneer immers eenzelfde bestand integraal opgehaald bij een peer op een andere deelnemende peer in het netwerk in verknipte fragmenten wordt opslaan, zal dit de optimale verspreiding van de multimedia via P2P-overdracht aardig kunnen verstoren. Er is immers in dit geval geen sprake meer van meerdere bronnen waarvan dezelfde media parallel kan worden gedownload, maar een evolutie naar het klassieke client-servermodel dat we zo angstvallig wilden vermijden.

Heterogeniteit van de multimediapresentaties

De bewerkingen die gebruikt worden voor het samenstellen van een multimediapresentatie die voldoet aan de specifieke noden van een gebruiker moeten kunnen worden toegepast op elk multimediabestand in de multimediabibliotheek gedeeld op het netwerk, ongeacht het type of de codering van dit bestand. Aangezien het de intentie was een P2P-protocol te ontwikkelen dat zonder transcoding naar een geoptimaliseerd gemeenschappelijk tussenformaat werkt, moet rekening gehouden worden met de verschillende mediatypes en -coderingen die in het systeem gebruikt worden.

Vergankelijkheid van de multimediapresentatie

Het is niet nuttig een samengestelde multimediapresentatie integraal bij te houden of te delen op het netwerk als een nieuwe multimediabijdrage eens ze is weergegeven. Het persoonlijke karakter van de samenstelling maakt het in de eerste plaats onwaarschijnlijk dat verschillende gebruikers precies diezelfde weergave zouden aanvragen. Als er geen tot weinig geïnteresseerde andere partijen zijn, is het een verspilling van systeembronnen om deze presentatie bij te houden. Bijgevolg zal de samenstelling van de specifieke multimediapresentatie enkel op de vragende peer plaatsvinden en na de weergave geen nood meer zijn aan de configuratie waarin de verschillende fragmenten werden geplaatst.

6.2 SPECIFICATIE

Nu de vereisten voor de specificaties van de onderdelen voor het personalisatiemechanisme zijn afgelijnd, kunnen specificaties worden ingevuld die aan deze voorschriften voldoen.

6.2.1 SPECIFICATIE VAN DE METADATA-ANNOTATIE

Alle metadata gebruikt bij het ontwerpen van een proof-of-concept bij dit werk werd als voorhanden beschouwd. Een expliciete specificatie van een annotatiestandaard en de hieraan verbonden procedures vallen dan ook buiten het bereik van deze masterproef en zullen bijgevolg niet aan bod komen in dit werk. Een voorbeeld van een samengestelde multimedia-annotatie en de integratie ervan in het P2P-systeem zal in het volgende hoofdstuk over de implementatie ter illustratie aan bod komen.

6.2.2 SPECIFICATIE VAN DE PERSONALISATIE VAN DE MULTIMEDIA

Zoals reeds vermeld bestaat er nog geen rechtlijnige implementatie voor een geautomatiseerde multimediabewerkingsumgeving die werkt met de gespecificeerde metadata-annotatie. Omdat het ontwikkelen van dergelijk systeem een langdurig en complex probleem is, kan hier niet worden op ingegaan binnen het bereik van deze masterproef. Daarom wordt hieronder een nieuw algemene kernomgeving gespecificeerd voor het personaliseren van multimediavoorstellingen aan de hand van metadata-annotaties. Hiervoor wordt reeds uitgegaan van het bestaan van reeds gesegmenteerde en per fragment geannoteerde multimedia, waardoor de basisoperaties binnen het mechanisme heel eenvoudig kunnen gehouden worden. Een uitgebreide uiteenzetting over de specifieke implementatie van dit

rudimentair personalisatieraamwerk voor multimedia en de implementatie ervan in het P2P-systeem volgt wederom in het volgende hoofdstuk.

Selectie van deelfragmenten uit de multimedia

Een eerste bewerking die mogelijk moet zijn bij het samenstellen van een gepersonaliseerde multimediapresentatie is het selecteren van een voor de gebruiker interessant multimediafragment uit een volledig multimediabestand. De hele bedoeling achter het personalisatiemechanisme is juist de gebruiker te ontdoen van irrelevante informatie in een opgevraagde presentatie. Bijgevolg zal het mogelijk moeten zijn om de relevante fragmenten uit een mediabestand als het ware te knippen om deze later te kunnen samenvoegen tot een aangepaste presentatie die specifiek aan de vraag van de gebruiker beantwoordt. Bovendien moet dit alles gebeuren zonder iets aan de oorspronkelijke opslag van het bronbestand te wijzigen.

Samenstelling van fragmenten tot een nieuwe representatie

Nadat de relevante segmenten voor een specifieke gebruiker zijn geselecteerd, zullen deze moeten samengevoegd worden ongeacht het heterogene karakter van de multimediasegmenten. Zoals reeds in de vereisten werd duidelijk gemaakt, zal de multimediabibliotheek van het P2P-distributienetwerk media van allerhande types en coderingen bevatten. Toch moet er een of andere manier gevonden worden om deze verschillende segmenten te combineren tot een coherente eenheid.

Als deze twee specificaties geïmplementeerd worden, zal de proof-of-concept eenvoudige multimediapresentaties op maat van een gebruiker kunnen samenstellen en wordt daarmee meteen de mogelijkheid van het concept van automatisch samenstellen van multimediapresentaties aan de hand van metadata bewezen.

HOOFDSTUK 7

PERSONALISERING VAN MULTIMEDIA: IMPLEMENTATIE

In dit hoofdstuk wordt de implementatie en integratie van de functionaliteit voor het personaliseren van de multimediateweergave in de reeds ontworpen P2P-architectuur van deel I beschreven.

Eerst zal het gebruikte metadatamodel voorgesteld worden. Met de bevindingen uit de state of the art en de specificatie in het achterhoofd kan de keuze voor dit model gemotiveerd worden. Eens het metadatamodel vastligt, moet een Java raamwerk gekozen worden waarmee dit model in de applicatie zal worden geïmplementeerd.

Wanneer de metadatafunctionaliteit in het protocol verwerkt is, zal een omgeving voor multimediatebewerking gedreven door beschikbare metadata-annotaties kunnen worden opgesteld. Ook de toevoeging van dergelijk raamwerk aan het protocol voor het stromen van multimedia ontworpen in deel I wordt in dit hoofdstuk uitgelegd.

7.1 WAAROM RDF?

Uit de vorige hoofdstukken is gebleken dat de invulling van een werkbare multimedia-annotatieomgeving niet zo eenvoudig is. Omdat geen van de voorgestelde metadatawoordenschaten bleek te voldoen aan de vooropgestelde vereisten, werd al snel beslist een eigen standaardnotatie te gebruiken.

Als basis voor de eigen standaard voor het annoteren van multimedia leek RDF de beste oplossing. Het datamodel mag dan wel niet specifiek ontworpen zijn voor het beschrijven van multimedia, toch vertoont het enkele interessante karakteristieken die dergelijke toepassing mogelijk maken. De reeds vermelde Dublin Core is overigens gebaseerd op RDF, maar is eerder ontworpen voor het annoteren van tekstuele bronnen. Om deze lijn door te trekken naar meer uitgesproken multimedia objecten in, zullen dus extra attributen eigen aan multimedia moeten kunnen beschreven worden.

Aangezien RDF voldoet aan alle vereisten opgelegd aan een praktische metadataomgeving²⁸ in de specificatie, kan er op basis van dit metadatamodel een metadatawoordenschat gespecificeerd worden die in combinatie met andere standaarden wel voldoet aan de benodigde functionaliteit voor het annoteren van metadata.

7.1.1 SPARQL

SPARQL²⁹ is een recursief acroniem dat staat voor "SPARQL Protocol And RDF Query Language". Zoals de naam al impliceert is SPARQL een algemene term voor zowel een protocol als een querytaal.

In de meeste gevallen zal de afkorting SPARQL echter verwijzen naar de RDF querytaal. In dit geval is SPARQL een taal die sterk op de SQL querytaal lijkt. Ze kan gebruikt worden voor het bevragen van RDF-grafen via patroonherkenning. De mogelijkheden van de taal omvatten, maar zijn niet beperkt tot het zoeken op basis conjunctieve patronen, waardefilters, optionele patronen, en patroon disjunctie.

²⁸ Zie 6.1.1 Vereisten voor de metadata-annotatie

²⁹ <http://www.w3.org/TR/rdf-sparql-query>

Voorbeelden

SPARQL maakt het gebruikers mogelijk om ondubbelzinnige zoekopdrachten met een globaal bereik te produceren. De volgende query in *Code 7.1* zal bijvoorbeeld de namen en e-mails van elke persoon in de wereld opvragen.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?email
WHERE {
  ?person a foaf:Person.
  ?person foaf:name ?name.
  ?person foaf:mbox ?email.
}
```

Code 7.1: SPARQL query naar de namen en e-mailadressen van alle personen in de wereld.

In *Code 7.2* zal dan weer een SPARQL zoekopdracht lanceren met de vraag: "Wat zijn alle hoofdsteden van de landen van Afrika?"

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
  abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
  abc:isInContinent abc:Africa .
}
```

Code 7.2: SPARQL query naar de hoofdsteden van alle landen in Afrika.

Deze zoekopdrachten illustreren de semantisch web visie waarbij het internet wordt behandeld als een enorme wereldwijde databank. Als multimedia op ons P2P-netwerk op een gelijkaardige manier geannoteerd wordt, zal analoog een netwerkwijde databank voor multimedia uitgebouwd worden.

RDF is met andere woorden bovenop alle voordelen van de modelstructuur ook nog eens voorzien van een krachtige kant en klare querytaal voor het datamodel. SPARQL zal dus zeker en vast in de implementatie van het protocol kunnen gebruikt worden.

7.1.2 JENA

Jena³⁰ is een open source semantisch web raamwerk voor Java. Het raamwerk biedt een API om gegevens op te halen uit en te schrijven naar RDF-grafen. De grafen worden weergegeven als een abstract "model". Dit model kan worden opgemaakt uit gegevens in bestanden, databases, URL's of elke combinatie van de voorgaande. In een model kan worden gezocht via SPARQL zoekopdrachten en data erin kan worden bijgewerkt door middel van SPARUL opdrachten.

Jena is vergelijkbaar met Sesame, maar in tegenstelling tot Sesame biedt Jena ondersteuning voor OWL³¹ (Web Ontology Language). Verder ondersteund Jena serialisatie van RDF-grafen naar een relationele databank, RDF/XML, Turtle of een 3N notatie.

Omwille van de uitgebreidheid van het raamwerk werd Jena boven Sesame verkozen om de RDF functionaliteit in de proof-of-concept te implementeren.

³⁰ <http://jena.sourceforge.net>

³¹ <http://www.w3.org/TR/owl-features>

7.2 IMPLEMENTATIE VAN DE METADATA IN HET P2P-STREAMING PROTOCOL

In het eerste deel van de masterproef werd reeds een P2P-protocol voorgesteld voor het stromen van multimediabestanden. Nu zal worden uitgelegd hoe precies de metadatafunctionaliteit gespecificeerd in het tweede deel in dit protocol kan worden geïntegreerd.

7.2.1 INTEGRATIE VAN DE METADATACATALOOG IN HET P2P-SYSTEEM

In het eerste deel van dit werk werd beschreven dat de metadata-annotatie samen met het bijhorende multimediabestand op het netwerk beschikbaar werd gesteld vanuit een gedeelde bestandsmap op de lokale opslag van de peer. Wanneer een naamconventie voor het metadatabestand wordt ingevoerd, kan op een eenvoudige manier een referentie tussen multimediabestand en metadata-annotatie worden bijgehouden.

Aangezien Jena een RDF-graaf kan serialiseren van een URI-bron kan de informatie in de metadatabestanden door een advertentie in het JXTA-CMS beschikbaar worden gesteld voor alle andere peers in dezelfde peergroep en de superknoop van dit netwerk. Naar een advertentie wordt immers met een URI verwezen. Op deze eenvoudige manier kan een soort gedistribueerde databank geïmplementeerd worden zonder dat hiervoor een configuratie van een extra databankproduct moet worden uitgevoerd.

7.2.2 ZOEKEN IN DE METADATACATALOOG VAN HET P2P-SYSTEEM

Door het gebruik van een systeem met advertenties voor het beschikbaar stellen van de metadata, kan met een uitbreiding van de standaard JXTA-CMS deze informatie bevraagd worden door middel van SPARQL zoekopdrachten op de RDF-graafvoorstelling van de advertentie.

De bestanden die aan deze zoekopdrachten voldoen zullen dan zoals bij de standaard tekstuele zoekopdrachten van de JXTA-CMS worden gedownload. Nu moet er wel voor worden gezorgd dat ook de metadata bij het bestand wordt meegezonden. Dit mag echter geen probleem vormen want juist om die redenen werd de naamconventie ingevoerd. Eens ook deze metadata zijn bestemming heeft bereikt, moet deze nog op zijn nieuwe locatie worden geadverteerd. Doordat de metadata samen met de multimedia door het P2P-netwerk reist, zal dus ook op een gelijkaardige wijze de superknoop op periodieke intervallen alle nieuwe advertenties van een peer moeten ontvangen. Op deze wijze blijft de superknoop op de hoogte van de inhoud van zijn netwerk.

7.3 IMPLEMENTATIE VAN DE GEAUTOMATISEERDE MULTIMEDIAPUBLICATIE IN HET P2P-STREAMING PROTOCOL

Voor het maken van aangepaste multimediapresentaties kan zoals reeds vermeld uitgegaan worden van op voorhand verdeelde multimediabestanden waarbij elk van de fragmenten voldoende is geannoteerd.

Implementatie van de selectie van deelfragmenten uit de multimedia

Aangezien de multimediabestanden reeds in de kleinst weergeefbare fragmenten verdeeld zijn, zal de selectie plaatsvinden op queryniveau. Selectie van een deelfragment van een multimediabestand is met andere woorden niets anders dan het selecteren van de fragmenten van dit multimediabestand die aan de zoekopdracht voldoen.

Implementatie van de samenstelling van fragmenten tot een nieuwe representatie

Om de geselecteerde fragmenten in een bepaalde configuratie weer te geven, moeten deze eenvoudigweg in deze volgorde aan de afspeelbuffer (i.c. herorderingsbuffer) afgeleverd worden. Deze zal immers zonder enige vorm van discriminatie alle multimediafragmenten die op de lijst werden geplaatst in sequentiële volgorde weergeven.

```
@prefix nsa: <http://ninsuna.elis.ugent.be/rdf/ontologies/ninsuna/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix : <http://ninsuna.elis.ugent.be/rdf/p2p/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

:Scene rdf:type owl:Class .
:Shot rdf:type owl:Class .

<http://example.org/exmedia.mp4> rdf:type nsa:MediaBitstream .

:exmedia rdf:type nsa:AnnotatedMultimedia ;
  nsa:isRepresentedBy <http://example.org/mediafile.mp4> ;
  nsa:hasTemporalSegment :exmedia_scene1;
  nsa:hasTemporalSegment :exmedia_scene2;
  nsa:hasTemporalSegment :exmedia_scene3 .

:exmedia_scene1 rdf:type :Scene, nsa:TemporalSegment ;
  nsa:start "00:00:00.00" ;
  nsa:duration "???" ;
  nsa:hasTemporalSegment :exmedia_scene1_shot1 ;
  nsa:hasTemporalSegment :exmedia_scene1_shot2 ;
  nsa:hasTemporalSegment :exmedia_scene1_shot3 .

:exmedia_scene1_shot1 rdf:type :Shot, nsa:TemporalSegment ;
  nsa:start "00:00:00.00" ;
  nsa:duration "???" ;
  dc:genre <http://dbpedia.org/resource/Category:Politics>, <http://dbpedia.org/resource/Category:Biography> ;
  dc:subject <http://dbpedia.org/resource/Herman_Van_Rompuy> ;
  dc:source <sc001sh001> .

:exmedia_scene1_shot2 rdf:type :Shot, nsa:TemporalSegment ;
  nsa:start "???" ;
  nsa:duration "???" ;
  dc:genre <http://dbpedia.org/resource/Category:Politics>, <http://dbpedia.org/resource/Category:Biography> ;
  dc:subject <http://dbpedia.org/resource/Herman_Van_Rompuy> ;
  dc:subject [ rdf:type foaf:Person; foaf:name "Geertrui Windels" ] ;
  dc:coverage <http://dbpedia.org/resource/European_Parliament> ;
  dc:source <sc001sh002> .
```

Code 7.3: Voorbeeld RDF gebaseerde metadata-annotatie bij een videosegment

HOOFDSTUK 8

CONCLUSIE

Ter afronding van deze masterproef wil ik nog even een overzicht geven van de geïmplementeerde functionaliteit die in mijn proof-of-concept is vervat en naar welke optimalisaties en toevoegingen aan het protocol in de toekomst nog mag worden uitgekeken.

8.1 GEÏMPLEMENTEERDE FUNCTIONALITEIT VAN HET PROTOCOL

Aangezien de demotoepassing op één enkele machine is ontworpen en de werking van het protocol hierdoor enkel op een virtuele netwerkstructuur werd getest, blijft de implementatie ervan vrij rudimentair. Op de eerste plaats was het de bedoeling om de grote lijnen voor een P2P-protocol voor het stromen van gepersonaliseerde multimedia uit te tekenen, en te bewijzen dat het ontwerpen en implementeren van een dergelijk protocol een realistische optie is. In dit werk werd dan ook de nadruk gelegd op het leveren van een antwoord op de twee vooruitgeschoven deelproblemen bij het ontwerp van het protocol: het transporteren van multimediasstromen via een P2P-structuur en het personaliseren van de multimedia. In termen van efficiëntie, adaptiviteit en veiligheid zal het voorgestelde protocol duidelijk te kort komen.

In het eerste deel van de masterproef werd het P2P-transportprobleem aangepakt. Een hybride P2P-architectuur werd geformuleerd waarin de beste eigenschappen van verschillende P2P-aanpakken werden gecombineerd. Om het stromen van multimedia over de bekomen hybride P2P-structuur mogelijk te maken, werden enkele optimalisaties ingebouwd om het sequentiële karakter van de multimedia te behouden. Verder werden ook technieken voorzien om de vooruitgang bij het stromen van een multimediamiaweergave te maximaliseren.

In deel twee werd een eenvoudige oplossing voor het personalisatieprobleem voorgesteld. Er werd uitgegaan van een reeds in fragmenten verknijpte en op segmentniveau RDF-geannoteerde multimediambibliotheek. Hierdoor kon de multimediambibliotheek door middel van de hoogst flexibele querytaal SPARQL bevroagd worden. Doordat associaties werden gelegd die in een niet-semantische omgeving onmogelijk waren, konden vrij specifieke multimediasegmenten geselecteerd worden die daarna samengevoegd werden tot een unieke presentatie op maat.

Al deze geïmplementeerde functionaliteit werd succesvol gecombineerd tot één protocol voor het stromen van multimedia. Wat meteen betekende dat er een oplossing bestond voor het probleem die voldeed aan de vooropgestelde vereisten. Het protocol mag dan wel doen waarvoor het werd ontwikkeld, het doet dit op een allesbehalve optimale wijze.

8.2 UITBREIDINGEN VAN HET PROTOCOL NAAR DE TOEKOMST TOE

Ondanks de rudimentaire vorm van de implementatie en het feit dat het beschreven protocol nog verre van toepasbaar is, biedt het mits enkele uitbreidingen voldoende potentieel naar de toekomst toe. Ter afsluiting van deze masterproef wil ik nog enkele ideeën kort toelichten die zouden bijdragen tot de bruikbaarheid van de implementatie.

8.2.1 P2P-OVERDRACHT

Op het vlak van de P2P-overdracht valt er op de architectuur zelf weinig op te merken. De meest optimale kenmerken van verschillende P2P-systemen werden immers in het protocol gecombineerd.

Rond een efficiënte indeling in peergroepen kan echter wel nog wat onderzocht worden. In de eerste plaats is er de vraag, "Hoeveel peers bevat een peergroep met de optimale zoek- en distributieprestaties?". De oplossing op dit probleem zal sterk beïnvloed worden door de aard van de peers (lees: hardwareconfiguratie) en de netwerktopologie. Hierdoor zal dus geen kant en klaar antwoord op deze vraag geformuleerd kunnen worden. Een zelfverbeterend heuristisch algoritme kan na verloop van tijd wel tot een optimale peercapaciteit voor een specifieke peergroep komen. De implementatie van een dergelijk algoritme brengt natuurlijk de nodige toegevoegde complexiteit in de peer met zich mee. Voor de samenstelling van peergroepen op basis van gemeenschappelijke eigenschappen van de peers moet voor die peers informatie over deze eigenschappen toegankelijk zijn alvorens ze lid worden van een groep. In deze informatie kan bijvoorbeeld een zoekgeschiedenis bijgehouden worden zodat peers met gelijke interesses in eenzelfde groep terecht komen en op die wijze efficiënter multimedia kunnen uitwisselen.

Daarnaast kan ook nog geëxperimenteerd worden met de verhouding tussen normale peers en superknoopen. Elke superknoop zal een maximaal aantal peers hebben die hij binnen een aanvaardbaar tijdsinterval kan bedienen. Wanneer te weinig peers door een superknoop worden bediend, betekent dit dat er mogelijks ergens onnodig een peer als superknoop fungeert en die peer bijgevolg overbodige complexiteit bevat. De eenvoudigste oplossing voor dit probleem is om bij het opstarten van een superknoop te bepalen wat de maximale bedieningscapaciteit van de knoop is. Dit kan op een vrij eenvoudige manier. De capaciteit van een superknoop kan bijvoorbeeld worden bepaald door een eenvoudige benchmark uit te voeren. Wanneer deze wordt overschreden, zal de volgende aanmeldende knoop automatisch een superknoop worden.

Een laatste optimalisatie van het P2P-protocol heeft te maken met de buffergrootte van de afspeelbuffer bij de gestroomde multimediapresentatie. In de huidige implementatie zal deze bij creatie een vaste grootte toegekend krijgen. Een te grote buffer zal niet alleen onnodig systeembronnen innemen, hij zal ook de start van de weergave onnodig ophouden. Wanneer het te lang duurt voordat de buffer vol is en kan vrijgegeven worden, zal de multimediaweergave worden onderbroken. Idealiter zou de buffergrootte gedurende de uitvoering dynamisch kunnen worden aangepast aan de bandbreedtefluctuaties van de overdracht.

8.2.2 PERSONALISATIE VAN DE MULTIMEDIA

Bij de personalisatie van de multimedia, het tweede deel van de masterproef, werd ervan uitgegaan dat de data reeds verdeeld was in fragmenten en was voorzien van de nodige metadata. Een volledig bruikbaar protocol zou een mechanisme moeten hebben dat gegeven een multimediabestand met een bijhorende metadata-annotatie dit bestand en zijn metadata automatisch in verschillende consistente fragmenten zou kunnen verdeeld worden.

Een tweede extra functionaliteit bij de personalisatie van metadata zou kunnen bestaan uit de integratie van een geïntegreerd metadata-raamwerk. In dit raamwerk moeten verschillende types reeds bestaande metadata-annotaties bij multimediabestanden kunnen worden omgezet naar het gemeenschappelijk RDF intern formaat dat bij de specificatie van dit protocol werd voorgesteld. Op die manier kan reeds met een andere standaard geannoteerde multimedia geïntegreerd worden in het systeem zonder dat deze helemaal opnieuw moet worden omschreven. Een gelijkaardig systeem wordt reeds in de Edutella (22) applicatie voorgesteld.

BIBLIOGRAFIE

1. *Internet Protocol (IP) Multicast*. [PDF Document] sl : Cisco Systems, Inc., 2000.
2. **Savetz, Kevin, Randall, Neil en Lepage, Yves.** *MBONE: Multicasting Tomorrow's Internet*. 1e editie. sl : John Wiley & Sons, Inc., 1996. ISBN: 978-1568847238.
3. **Geurts, Joost, van Ossenbruggen, Jacco en Hardman, Lynda.** *Requirements for practical multimedia annotation*. [PDF document] Amsterdam : CWI, 2005.
4. **Verstryngne, Jérôme.** *Practical JXTA*. 1e editie. sl : DrawingStreams, Inc, 2008. pp. 25-26. 978-1-4092-1564-6.
5. **Verstryngne, Jérôme.** *Practical JXTA*. 1e editie. sl : DrawingStreams, Inc., 2008. p. 26. 978-1-4092-1564-6.
6. **Tyson, Jeff.** How the Old Napster Worked. *HowStuffWorks.com*. [Online] HowStuffWorks. [Citaat van: 21 02 2010.] <http://computer.howstuffworks.com/napster.htm/printable>.
7. **Verstryngne, Jérôme.** *Practical JXTA*. 1e editie. sl : DrawingStreams, Inc., 2008. pp. 27-28. 978-1-4092-1564-6.
8. *The Gnutella Protocol Specification v0.4*. [PDF document] sl : Clip2, 2003.
9. **Verstryngne, Jérôme.** *Practical JXTA*. 1e editie. sl : DrawingStreams, Inc., 2008. p. 28. 978-1-4092-1564-6.
10. giFT-FastTrack home page. *The FastTrack Protocol*. [Online] giFT-FastTrack, 27 07 2004. [Citaat van: 21 02 2010.] <http://cvs.berlios.de/cgi-bin/viewcvs.cgi/gift-fasttrack/giFT-FastTrack/PROTOCOL?revision=1.19>.
11. **Foster, Ian en Iamnitchi, Adriana.** *On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing*. [PDF document] Chicago : University of Chicago, 2002.
12. **Verstryngne, Jérôme.** *Practical JXTA*. 1e editie. sl : DrawingStreams, Inc., 2008. p. 29. 978-1-4092-1564-6.
13. **Cohen, Bram.** The BitTorrent Protocol Specification. *BitTorrent.org*. [Online] BitTorrent, 10 01 2008. [Citaat van: 21 02 2010.] http://bittorrent.org/beps/bep_0003.html.
14. **Dhoedt, Bart en De Turck, Filip.** Peer-to-peer systems. *Ontwerp van geDistribueerde Software*. Gent : sn, 2009.
15. **Jurca, Dan, et al.** *Enabling Adaptive Video Streaming in P2P Systems*. [PDF document] Lausanne : Ecole Polytechnique Fédérale de Lausanne.
16. **Guha, Saikat, Daswani, Neil en Jain, Ravi.** *An Experimental Study of the Skype Peer-to-Peer VoIP System*. [PDF document] Ithaca : Cornell University, 2006.
17. **Xiaojun, Hei, Lui, Yong en Ross, Keith W.** *IPTV over P2P Streaming Networks: the Mesh-pull Approach*. [document] Brooklyn : Polytechnic University, 2008.

18. **Purvi, Shah en Pâris, Jehan-François.** *Peer-to-Peer Multimedia Streaming Using BitTorrent*. [PDF document] Houston : University of Houston, 2007.
19. **Gong, Li.** *Project JXTA: A Technology Overview*. [PDF document] Palo Alto : Sun Microsystems, Inc., 2002.
20. **Brookshier, Daniel, et al.** *JXTA: Java P2P Programming*. 1e editie. Canada : Sams Publishing, 2002. pp. 253-254. 978-0672323669.
21. **Martínez, José M.** MPEG-7 Overview. *INTERNATIONAL ORGANISATION FOR STANDARDISATION*. [Online] INTERNATIONAL ORGANISATION FOR STANDARDISATION, 10 2004. [Citaat van: 14 04 2010.] <http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>.
22. **Nejdl, Wolfgang, et al.** *EDUTELLA: A P2P Networking Infrastructure Based on RDF*. [PDF Document] Hannover : University of Hannover, 2001.

