

Erzeugung von hoch skalierbaren virtuellen Welten

Seminar
Algorithmen zur Unterstützung von Immersive Gaming

von
Philip Mildner
Oktober 2008

Universität Mannheim
Lehrstuhl Praktische Informatik IV
Prof. Dr. Wolfgang Effelsberg
68159 Mannheim
Germany

Inhaltsverzeichnis

1	Einführung	3
1.1	Motivation	3
1.2	Anforderungen an ein NVE	3
1.3	Spezielle Anforderungen an skalierbare NVEs	4
1.4	Einordnung der Herangehensweisen	4
2	Interessenmanagement	5
3	SimMud	7
3.1	Netzwerk	7
3.2	Interessenmanagement	8
3.3	Koordinatoren	8
3.4	Bemerkungen	9
4	Voronoi-Based Adaptive Scaleable Transfer	9
4.1	Voronoidiagramme	10
4.2	Hinzufügen von Nutzern	11
4.3	Bewegungen	12
4.4	Entfernen von Nutzern	12
4.5	Sonstige Interaktionen	13
4.6	Ergebnisse	13
4.7	Bemerkungen	14
5	Fazit	14

Abbildungsverzeichnis

1	Einteilung des NVE in Regionen	6
2	SimMud: Regionen	8
3	VAST: Nachbarschaftstypen	11

1 Einführung

In dieser Arbeit werden Konzepte zur Erstellung von hoch skalierbaren Onlinewelten vorgestellt. Im Gegensatz zum weit verbreiteten Client-Server-Modell basieren die hier vorgestellten Verfahren auf Peer-to-Peer (P2P) Modellen um so die Ressourcen dynamisch an die Nutzerzahlen anpassen zu können. Um die Last bei den einzelnen Knoten gering zu halten wird außerdem das Konzept des Interessenmanagements eingeführt. Anschließend werden die Konzepte anhand zweier Implementationen erläutert und diskutiert.

1.1 Motivation

In den vergangenen Jahren sind Onlinewelten immer populärer geworden, sei es durch Spiele wie „World of Warcraft“ oder soziale Plattformen wie „Second Life“. Traditionell basieren diese sogenannten „vernetzten virtuellen Welten“, oder *Networked Virtual Environments* (NVEs), auf dem Client-Server-Prinzip, bei dem sich alle teilnehmenden Nutzer zu einem Server verbinden. Mit steigenden Nutzerzahlen wird dieser Ansatz allerdings unpassend, da ein Server nur begrenzte Kapazitäten zur Verfügung stellen kann. Konzepte wie das Spiegeln der Onlinewelt auf mehrere getrennte Server bieten zwar Abhilfe, sind aber mit hohen Kosten verbunden und haben zudem den Nachteil, dass nicht alle Nutzer miteinander kommunizieren können, da sie sich auf getrennten Servern befinden. Um hoch skalierbare NVEs zu erschaffen bietet sich deshalb die Nutzung von P2P-Netzen an. In den folgenden Abschnitten werden verschiedene Verfahrensarten vorgestellt und die Anforderungen an ein solches NVE aufgelistet.

1.2 Anforderungen an ein NVE

Bei der Erschaffung eines NVE müssen verschiedene Faktoren berücksichtigt werden. Im Einzelnen sind dies:

- **Konsistenz:** Alle Nutzer sollten die selbe Repräsentation des NVE erleben. Dazu ist es nötig Aktionen zu synchronisieren und Statusinformationen aktuell zu halten.
- **Ansprechbarkeit:** Ein NVE sollte in ausreichend schneller Zeit auf Benutzereingaben reagieren. Die maximale Wartezeit ist dabei abhängig von der Art der Anwendung.
- **Sicherheit:** Um Betrugsversuche auszuschließen sollte ein NVE geeignete Mechanismen zu Verfügung stellen um sowohl die legitimen Nutzer zu authentifizieren, als auch nicht erlaubte Aktionen innerhalb des NVE zu verhindern.

- Skalierbarkeit: Ein NVE sollte mit der Zahl seiner Nutzer skalieren um mehr Nutzern zu ermöglichen miteinander zu interagieren.
- Persistenz: Sowohl Nutzerdaten als auch veränderliche Daten im NVE müssen über Sitzungen hinweg gespeichert werden.
- Verfügbarkeit: Ähnlich wie jeder Onlinedienst sollte ein NVE ohne Ausfälle zur Verfügung stehen.

In dieser Arbeit liegt der Hauptaugenmerk auf dem Punkt der Skalierbarkeit, die anderen Punkte spielen deshalb eine untergeordnete Rolle. So genügt es beispielsweise bei der Sicherheit die Nutzer zu authentifizieren. Die hier vorgestellten Implementationen speichern zudem keine Informationen über Sitzungen hinweg, vernachlässigen also den Punkt der Persistenz.

1.3 Spezielle Anforderungen an skalierbare NVEs

Um hoch skalierbare NVEs erstellen zu können müssen zwei Punkte implementiert werden:

1. Die nutzbaren Ressourcen müssen mit der Anzahl der Teilnehmer steigen.
2. Neu hinzugefügte Teilnehmer dürfen keine Ressourcen verbrauchen, die nicht mit skalieren.

1.4 Einordnung der Herangehensweisen

Um ein NVE skalierbar zu der Nutzerzahl zu machen gibt es verschiedene Ansätze. Diesen reichen von klassischen Client-Server- zu verteilten P2P-Modellen. Die wichtigsten Ansätze sind folgend aufgeführt:

Client-Server Hierbei läuft die komplette Kommunikation über einen Server. Übersteigt die Nutzerzahl die Kapazität des Servers wird ein neuer Server bereitgestellt, der den gleichen Inhalt bietet. Nutzer können sich entscheiden zu welchem Server sie sich verbinden wollen. Sie sehen auf einem Server aber nur eine Parallelwelt unter mehreren, und können so nicht mit allen Nutzern kommunizieren.

Grid-Client-Server Bei diesem Ansatz werden mehrere Server miteinander vernetzt, entweder über ein LAN oder über ein Computergrid. Je nach Implementation kann so das NVE auf mehrere Server verteilt werden, oder die Server teilen sich nach Aufgaben auf, die bei dem Betrieb des NVE anfallen.

Multicast-Systeme Um die Netzwerklast zu verringern kann auf Multicastübertragungen zurückgegriffen werden. Dafür muss allerdings die geeignete Infrastruktur bereitgestellt werden.

hybride P2P-Netze Diese P2P-Netze setzen auf verschiedene Arten von Peers, normale und privilegierte, sogenannte *Super Peers*. Diese Super Peers haben ein gesonderten Status und können sowohl für die Verwaltung des Netzes als auch zur Verteilung der Last zuständig sein. Normalerweise sind Super Peers einfache Clientrechner, die durch ein Auswahlverfahren für die Dauer einer Sitzung zum Super Peer werden.

voll verteilte P2P-Netze Bei reinen P2P-Netzen sind Clientrechner gleichwertig miteinander vernetzt. Dadurch kann eine gute Skalierbarkeit erreicht werden, da sich die Kapazitäten dynamisch an die Nutzerzahlen anpassen. Anders als beim Client-Server-Modell bedeutet dieser Ansatz allerdings höheren Verwaltungsaufwand, sowohl bei der Verwaltung des P2P-Netzes als auch beim Status des NVE, der bei allen Teilnehmern konsistent gehalten werden muss.

Alle Konzepte gründen auf einem von zwei Verfahren zur Erhöhung der Skalierbarkeit, diese sind:

1. Erhöhung der Kapazität
2. Verringerung der Last

Während beide Client-Server-Modelle alleine auf dem ersten Grundsatz aufbauen, spielt bei P2P-Systemen die Verringerung der Last ebenfalls eine große Rolle. Dadurch, dass sich nur normale Clientrechner im Netz befinden, muss gewährleistet werden, dass diese mit der Verwaltung des Netzes und des NVE nicht überlastet werden. Im folgenden Abschnitt wird deshalb das Konzept des *Interessenmanagements* vorgestellt.

2 Interessenmanagement

Um NVEs aufzubauen, die auf P2P-Netzen basieren, ist es notwendig die Last bei den einzelnen Peers möglichst gering zu halten. Solange alle Nutzer miteinander verbunden sind, liegen die Kosten für Nachrichtenübermittlungen bei $O(n^2)$. Techniken wie Kompression der Datenpakete können zwar die Datenlast verringern, für hohe Skalierbarkeit ist es aber notwendig die Anzahl der übermittelten Nachrichten zu senken.

Aus diesem Grund wird das sogenannte *Interessenmanagement* eingeführt. In Widerspiegelung zu realen Welt müssen die Nutzer in einem NVE nicht die ganze

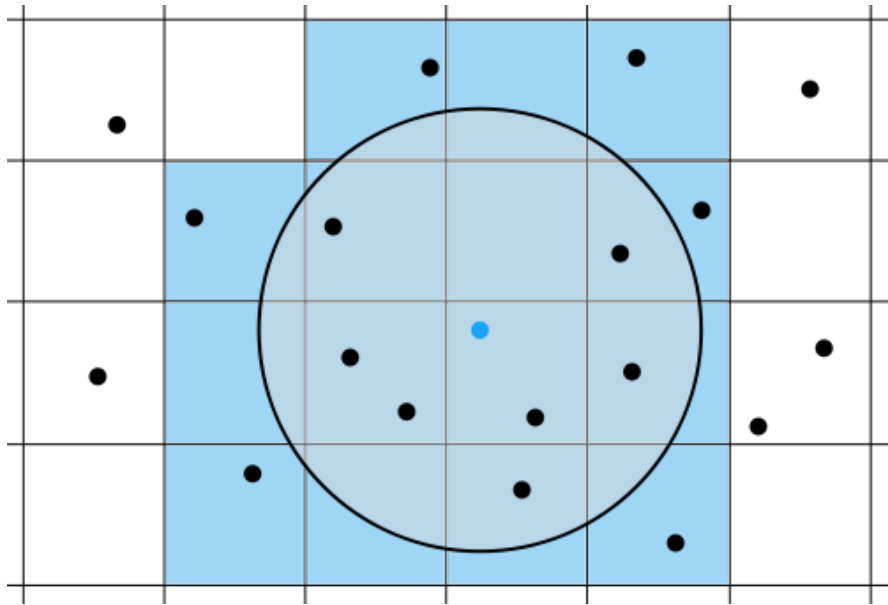


Abbildung 1: Der betrachtete Knoten (blau) verbindet sich zu den Regionen, die sich mit seinem AOI überschneiden (hier hellblau).

Welt kennen, sondern können sich auf den Bereich beschränken, in dem sie sich aktuell befinden. Zur Positionsbestimmung wird dabei nicht die physikalische Netzwerktopologie verwendet, sondern die Position des Nutzers innerhalb des NVE. Neue Bereiche können über Nachbarknoten entdeckt werden, die selber wieder mit weiteren Nachbarn verbunden sind. Dabei muss das NVE sicherstellen, dass das NVE nicht in unabhängige Bereiche zerfällt, in denen nicht alle Nutzer Kontakt halten können.

Zur Abbildung dieses Konzeptes wird der Begriff des „Interessenradius“, oder *Area of Interest* (AOI), eingeführt. Der AOI beschreibt die Umgebung um einen Nutzer, die für ihn relevante Informationen enthält. Alle Ereignisse außerhalb des AOI sind somit irrelevant für einen Nutzer. Um den AOI auf ein NVE abbilden zu können gibt es zwei Herangehensweisen, im Einzelnen sind dies:

Feste Regionen Hierbei wird das NVE in Regionen fester Größe unterteilt. Ein Nutzer hat Kontakt zu allen Nutzern aus Regionen, die sich mit dem AOI des Nutzers überschneiden. Wenn der Nutzer seinen Standort verändert und sein AOI sich somit mit neuen Regionen überschneidet, verbindet er sich zu den Nutzern der neuen Region und trennt dementsprechend die Verbindung zu Nutzern aus Regionen, die sich nicht mehr mit seinem AOI überschneiden. Je nach Größe der Regionen kann der Nutzer gleichzeitig zu einer oder mehreren Regionen verbunden sein (siehe Abbildung 1).

Dynamische Regionen Das NVE wird in Regionen eingeteilt, die sich dynamisch mit den Nutzern bewegen. So kann jedem Nutzer eine eigene Region zugeteilt sein, die sich mit ihm durch das NVE bewegt. Wie beim ersten Ansatz kann sich so ein Nutzer zu den Regionen verbinden, die sich mit seinem AOI überschneiden.

Zwar ist der erste Ansatz leichter zu implementieren, da die Regionen statisch bestimmt werden können, allerdings spiegelt er das Konzept des AOI nicht voll wieder. Sind die Regionen größer als der AOI der Nutzer empfangen diese unnötige Nachrichten von Nutzern, die sich außerhalb ihres AOI befinden. Genauso kann es passieren, dass sich Nutzer bei zu klein gewählter Regionsgröße zu vielen Regionen verbinden müssen, was wiederum die Netzwerklast erhöht.

In den folgenden Abschnitten werden zwei Implementierungen vorgestellt, die auf den verschiedenen Grundsätzen aufbauen.

3 SimMud

SimMud [5] ist eine Implementation eines NVE, das auf dem P2P-Netz *Pastry* [6] aufbaut. Es unterstützt das Konzept des Interessenmanagements und setzt dies durch ein *hybrides P2P-Netz* mit Multicastunterstützung durch.

3.1 Netzwerk

Das zugrundeliegende P2P-Netzwerk für SimMud ist Pastry. Pastry verwendet verteilte Hashtabellen, oder *Distributed Hash Tables* (DHTs), als Adressraum. Dabei werden sowohl Peers als auch Daten auf den selben Adressraum abgebildet. Die Daten werden dabei von den Peers verwaltet, zu denen sie im Adressraum numerisch am nächsten liegen.

Zur Multicastunterstützung wird *Scribe* [2] verwendet. Scribe bietet Multicast auf Applikationsebene, aufbauend auf dem Pastry-Netzwerk. Dies hat den Vorteil, dass im physischen Netzwerk keine Multicastunterstützung vorhanden sein muss. Wenn Peers eine Multicastgruppe bilden, wird diese als neues Objekt auf den Adressraum des Pastry-Netzwerks abgebildet. Die Vereinigung der Routen von den Gruppenteilnehmern zum Gruppenobjekt bildet dann die Route für den Multicast.

Als einzige zentrale Ressource dient ein einzelner Server. Dieser kann als Authentifizierung für Nutzer des NVEs und gleichzeitig als *Bootstrap*-Server zum Einstieg in das P2P-Netz dienen.

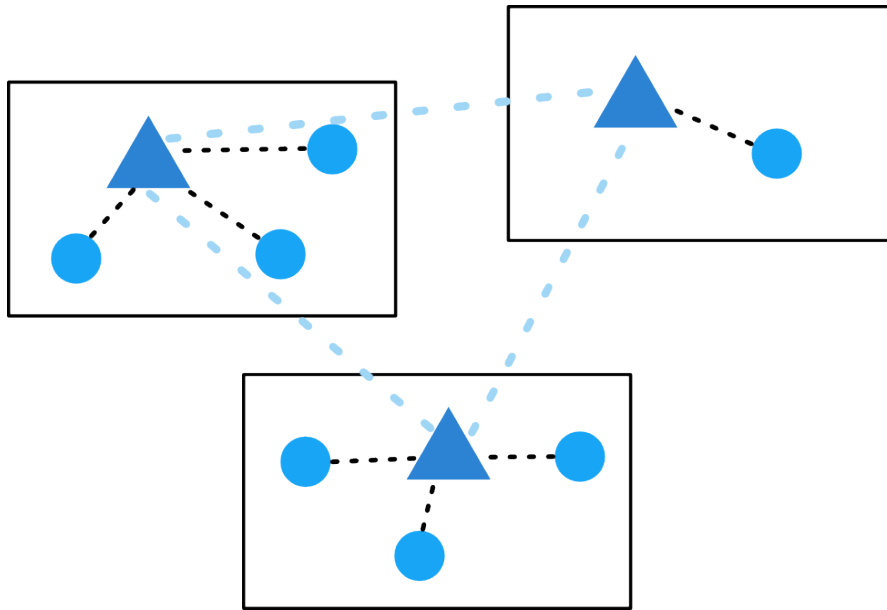


Abbildung 2: Jeder Koordinator (Dreieck) verwaltet eine Region des NVE. Nutzer (Kreise) kommunizieren über die Multicastgruppe, die der Koordinator verwaltet.

3.2 Interessenmanagement

SimMud implementiert das Interessenmanagement durch die Einteilung des NVE in Regionen fester Größe. Betritt ein Teilnehmer eine neue Region verbindet er sich dementsprechend zu allen Teilnehmern dieser Region. Die Kommunikation findet dabei über die Multicastgruppe statt, der alle Nutzer der Region angehören (siehe Abbildung 2). In festen Zeitintervallen teilen alle Teilnehmer ihre Position den anderen mit um die Region konsistent zu halten.

Zusätzlich zur globalen Multicastgruppe der Region kann es noch weitere Gruppen geben, der nur ein Teil der Teilnehmer angehören. So ist es für normale Aktivitäten innerhalb des NVE ausreichend die Aktionen der Teilnehmer zu kennen, die sich im eigenen AOI, beispielsweise der Sichtweite, befinden. Anhand der Positionen aller Teilnehmer können alle Teilnehmer feststellen zu welchen Gruppen sie beitreten müssen, damit sie alle wichtigen Informationen erfassen können.

3.3 Koordinatoren

In SimMud gibt es außer normalen Teilnehmern noch sogenannte „Koordinatoren“, die die Aufgabe von *Super Peers* übernehmen. Jeder dieser Koordinatoren verwaltet genau eine Region und dient als Einstiegspunkt für Teilnehmer, die sich neu zu dieser Region verbinden. Außerdem kann durch die Koordinatoren eine

gewisse Sicherheit gewährleistet werden. Da die Kommunikation zu allen Objekten in einer Region über den Koordinator läuft, kann dieser sicherstellen, dass nur legitime Informationen weitergegeben werden. Verändert sich der Status eines Objektes, teilt der Koordinator dies allen Teilnehmern per Multicast mit.

3.4 Bemerkungen

Das hier vorgestellte Verfahren zeigt gute Ansätze um für hoch skalierbare NVEs eingesetzt zu werden. Durch den Einsatz von Pastry als zugrundeliegendes Netzwerk und Verwendung von Multicast können Nachrichten verlässlich übertragen werden. Zudem berücksichtigt SimMud beide Punkte an die Anforderungen skalierbarer NVEs. Durch die Verwendung von Super Peers kann ein gewisser Grad an Sicherheit geboten werden, was für den Erfolg eines NVE essentiellen Einfluss haben kann.

Gerade die Super Peers bergen allerdings auch einen Schwachpunkt. Da prinzipiell jeder Teilnehmer zum Super Peer werden kann, kann es passieren, dass die Rechen- oder Netzwerklast für diesen Rechner zu groß wird und er somit zum Flaschenhals wird. Auf Grund der besonderen Verantwortung der Super Peers muss zudem sichergestellt werden, dass nur vertrauenswürdige Teilnehmer als Kandidaten herangezogen werden.

Ein Nachteil der Einteilung des NVE in Regionen fester Größe ist, dass diese Einteilung den tatsächlichen AOI der Teilnehmer nicht gut wieder spiegelt. Hält sich ein Teilnehmer am Rand einer Region auf kann er keine Teilnehmer der nächsten Region sehen, da er zu diesen nicht verbunden ist. Um das zu vermeiden wäre es nötig, dass sich ein Teilnehmer in mehreren Regionen anmeldet, was die Netzwerklast wiederum unnötig erhöht.

Zudem ist SimMud nicht für Anwendungen geeignet, bei denen es auf schnelle Reaktionszeiten ankommt, beispielsweise *First Person Shooter*. Multicast in SimMud ist über Scribe implementiert, somit müssen die Nachrichten über das Pastry-Netz geroutet werden. Die Anzahl der Hops beträgt dabei maximal $\log(n)$, was bei typischen Nutzerzahlen von NVEs zu großen Verzögerungen führt. Punkt-zu-Punkt-Verbindungen können nur in Ausnahmesituationen hergestellt werden, etwa bei der Unterhaltung zweier Nutzer.

4 Voronoi-Based Adaptive Scaleable Transfer

Voronoi-Based Adaptive Scaleable Transfer (VAST) [1] ist eine Implementation eines NVE, das auf einem Voronoidiagramm-basierten P2P-Netzwerk [3] aufbaut. Zunächst wird das Konzept der Voronoidiagramme vorgestellt, anschließend wird das Verfahren beschrieben und diskutiert.

4.1 Voronoidiagramme

Voronoidiagramme dienen dazu einen Raum in Teilbereiche zu unterteilen. Da für die Zwecke von VAST eine Fläche genügt, wird hier allerdings nur der zweidimensionale Fall betrachtet.

Gegeben eine Menge von Punkten N , die auf einer Fläche verteilt sind, ordnet das Voronoidiagramm einem Punkt $m \in N$ alle Punkte der Fläche zu, die zu m einen geringeren Abstand haben als zu allen anderen Punkten in N . Der so entstandene Bereich wird als *Region* bezeichnet. Angewendet auf alle Punkte lässt sich damit eine Fläche auf deterministische Weise in Teilbereiche zerlegen. Bezogen auf VAST kann so das NVE von den Nutzern in Regionen unterteilt werden, wobei jeder Nutzer eine eigene Region besitzt. Diese Unterteilung wird sowohl für das Interessenmanagement als auch für die Verwaltung des Netzwerks genutzt.

Zur Aufrechterhaltung des Netzwerks genügt es die direkten Nachbarn zu betrachten. Verbindet sich jeder Teilnehmer zu all seinen direkten Nachbarn, so ist sichergestellt, dass es optimale Pfade zwischen zwei Knoten gibt. Dazu wird eine Nachricht jeweils an den Nachbarn weitergeleitet, der im *Overlay*-Netzwerk am nächsten zum Zielpunkt liegt. Sobald sich das Netzwerk ändert – durch Positionsveränderung der teilnehmenden Knoten oder durch Hinzufügen neuer Knoten – müssen weitere Nachbarn betrachtet werden. Dabei kommt eine positive Eigenschaft der Voronoidiagramme zum Tragen, da solche Veränderungen nur lokale Veränderungen haben. Ändert sich beispielsweise die Position eines Punktes, so müssen lediglich die Regionen der benachbarten Punkte aktualisiert werden.

Entsprechend des Interessenmanagements verbindet sich ein Knoten zu allen Nachbarn, die in seinem AOI liegen. Der AOI wird in VAST durch einen kreisförmigen Bereich um den Teilnehmer realisiert, der für alle Teilnehmer die gleiche fest gewählte Größe hat. Dadurch ergeben sich verschiedene Nachbarschaftstypen (siehe Abbildung 3). Diese sind:

1. Regionen, die eine gemeinsame Kante mit der Region eines Teilnehmers haben, werden als *direkte* Nachbarn bezeichnet.
2. Regionen, die sich mit dem Rand des AOI eines Teilnehmers schneiden, heißen *angrenzende* Nachbarn.
3. Regionen, die vollständig im AOI eines Teilnehmers liegen, aber keine gemeinsame Kante mit der Region des Teilnehmers haben, heißen *normale* Nachbarn.

In den folgenden Abschnitten werden nun die möglichen Aktionen innerhalb des NVE genauer erläutert.

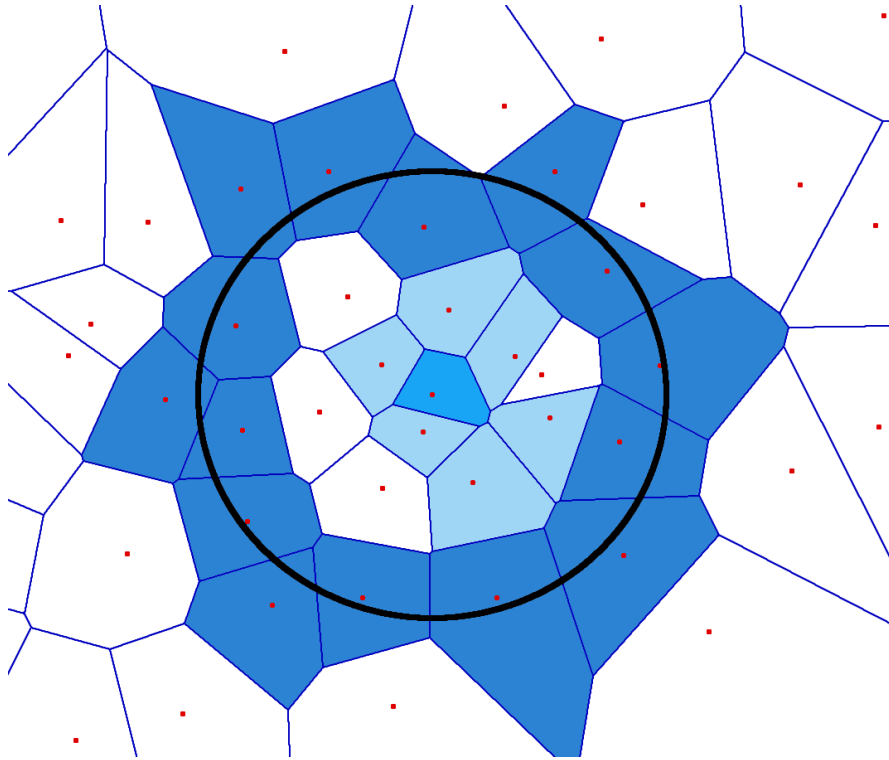


Abbildung 3: VAST unterscheidet zwischen direkten Nachbarn (hellblau) und angrenzenden (dunkelblau). Diese sind bezogen auf den AOI des betrachteten Knotens (blau). Quelle Originalbild [4].

4.2 Hinzufügen von Nutzern

Will sich ein neuer Teilnehmer N zu einem VAST-Netzwerk verbinden, muss er zunächst einen wohlbekanntem Server S innerhalb des NVE, einen sogenannten *Bootstrap*-Server, kontaktieren. Dieser kann, falls erforderlich, den Nutzer zunächst authentifizieren. Das weitere Verfahren wird in den folgenden Schritten erläutert:

1. S teilt N eine eindeutige ID zu, die einem Punkt auf der Fläche des NVE entspricht.
2. Der Nutzer fragt über S seine eigene ID ab. Diese wird über direkte Nachbarn vom Server zu dem Knoten M geleitet, der die Region verwaltet, in der die ID von N liegt.
3. N kontaktiert M . Dieser schickt N eine Kopie seiner kompletten Nachbarschaft. Dadurch kann N sein lokales Voronoidiagramm erstellen.

4. N kontaktiert alle Nutzer in seiner Nachbarschaft. Diese können so ihre Voronoidiagramme an die neue Situation anpassen.

4.3 Bewegungen

Bewegt sich ein Nutzer innerhalb des NVE müssen zwei Punkte berücksichtigt werden. Zum einen müssen die Regionen der umliegenden Nutzer aktualisiert werden. Zudem muss überprüft werden, ob sich der AOI des Nutzers mit Regionen unbekannter Nutzer überschneidet bzw. ob Regionen aus dem AOI fallen. Beides wird bei folgendem Verfahren berücksichtigt:

1. Der sich bewegende Nutzer B sendet die Zielkoordinaten seiner Bewegung an all seine Nachbarn. Nachrichten an die angrenzenden Nachbarn erhalten eine spezielle Markierung.
2. Alle Nachbarn aktualisieren ihre Voronoidiagramme. Handelt es sich bei einem Knoten A um einen angrenzenden Nachbarn, so prüft dieser ob sich der neue AOI von B mit Regionen überschneidet, die direkte Nachbarn von A sind. Ist dies der Fall sendet der Nachbar eine Benachrichtigung über die neuen Knoten an B .
3. Hat B Benachrichtigungen über neue Nachbarn erhalten, so verbindet er sich zu diesen. Analog dazu trennt er die Verbindung zu Knoten, dessen Regionen außerhalb seines neuen AOI liegen.

4.4 Entfernen von Nutzern

Das Entfernen von Nutzern aus dem NVE kann zwei Gründe haben. Entweder beenden Nutzer die Verbindung zum NVE explizit, oder die Verbindung reißt ab, etwa durch Netzwerkprobleme. Im Falle eines gewollten Verbindungsabbaus gilt folgendes Verfahren:

1. Der Nutzer V , der die Verbindung beendet, sendet eine entsprechende Nachricht an all seine Nachbarn, die eine Liste seiner direkten Nachbarn enthält.
2. Alle angesprochenen Knoten aktualisieren ihre Voronoidiagramme. Handelt es sich bei V für den betroffenen Knoten um einen angrenzenden Nachbarn, so kann der Knoten mit Hilfe der direkten Nachbarn von V einen neuen angrenzenden Knoten finden.

Stellt ein Knoten fest, dass einer seiner Nachbarn nicht mehr vorhanden ist, sendet er eine Anfrage an all seine Nachbarn ihm eine Liste mit direkten Nachbarn zu schicken. Dadurch können die Voronoidiagramme auf den aktuellen Stand gebracht werden.

4.5 Sonstige Interaktionen

Neben den Bewegungen, die die wichtigste Aktion innerhalb des NVE darstellen, kann es je nach Implementation noch weitere Interaktionen zwischen Nutzer und NVE oder zwischen mehreren Nutzern geben. Diese betreffen je nach Art nur einzelne Nutzer oder die Umgebung um den Ort der Aktion. Dementsprechend werden Nachrichten solcher Aktionen nur zu einzelnen Nutzern oder an die Nachbarschaft eines Nutzers geschickt.

So genügt es beispielsweise bei Unterhaltungen zwischen zwei Nutzern die Nachrichten nur zwischen den beteiligten Nutzern auszutauschen. Sichtbare Aktionen innerhalb des NVE, etwa der Kampf eines Nutzers gegen einen Nichtspielercharakter, müssen hingegen allen Nachbarn mitgeteilt werden.

4.6 Ergebnisse

In diesem Abschnitt werden die Ergebnisse einer Simulation mit VAST von Backhaus et al. [1] vorgestellt und dabei Stärken und Schwächen des Verfahrens aufgezeigt. Grundlage der Ergebnisse ist eine Simulation, bei der ein NVE erstellt und mit 500 virtuellen Nutzern gefüllt wurde. Dabei wurden bei der Bewegung der Nutzer zwei Strategien verfolgt:

1. In einem ersten Aufbau suchten sich die Nutzer unabhängig voneinander ein zufälliges Ziel innerhalb des NVE aus und bewegten sich zu diesem. Diesen Vorgang wiederholten sie bis zum Simulationseende. Diese Strategie wird als *Random Waypoint* bezeichnet.
2. Im zweiten Aufbau schlossen sich die Nutzer zu Gruppen mit je 12 Teilnehmern zusammen. Gemeinsam als Gruppe suchten sie sich bis Simulationseende zufällig gewählte Ziele heraus und bewegten sich dort hin. Dementsprechend wird diese Strategie als *Group-based Random Waypoint* bezeichnet.

Bei der Betrachtung der Netzwerkkonsistenz ergaben sich zwischen beiden Ansätzen gravierende Unterschiede. Zur Überprüfung wurde das globale Voronoidiagramm mit den lokalen der Nutzer verglichen. Während im *Random Waypoint* Modus das Netzwerk über die Dauer der Simulation konsistent blieb, verloren einzelne Nutzer im *Group-based Random Waypoint* Modus Kontakt zu ihrer Gruppe, bis zu dem Punkt, an dem die Gruppe nahezu vollständig aufgelöst war. Für diese Inkonsistenz gibt es primär zwei Gründe:

Schnelle Bewegungen Bewegt sich ein Nutzer, so prüfen dessen angrenzende Nachbarn, ob sich der AOI des bewegenden Nutzers mit Regionen von deren direkten Nachbarn überschneidet. Diese Überprüfung kann jedoch unvollständig sein, wenn die Distanz der Bewegung zu groß ist, oder wenn die

umliegenden Regionen zu klein sind. Ist dies der Fall kann sich der AOI des bewegenden Nutzers mit Regionen überschneiden, die sich hinter den direkten Nachbarn der angrenzenden Nachbarn befinden. Zu diesen Nutzern kommt keine Verbindung zustande und dementsprechend entsteht ein inkonsistentes NVE.

Einseitige Nachbarschaften Ein weiteres Problem kann bei entweder zu großem AOI oder zu kleinen Regionen entstehen. So kann es passieren, dass ein Nutzer A ein angrenzender Nachbar eines Nutzers B ist, B allerdings kein Nachbar von A ist. Dadurch werden Verbindungsversuche von B an A beim Nutzer A verworfen und es kommt wiederum keine Verbindung zustande.

4.7 Bemerkungen

VAST berücksichtigt beide Anforderungen an skalierbare NVEs. Außer dem Bootstrap-Server, der nach dem Verbindungsaufbau nicht mehr benötigt wird und somit nicht mit skalieren muss, werden keine zentralen Ressourcen verbraucht. Da alle Knoten im Netzwerk die gleiche Funktionalität haben, sind Flaschenhälse im Netzwerk so gut wie ausgeschlossen. Zwar kann es passieren, dass einzelne Nutzer eine höhere Last haben, etwa wenn viele Nutzer auf engem Raum gedrängt sind, doch dies stellt die Ausnahme dar.

Einen großen Vorteil, den VAST speziell im Vergleich zu SimMud bietet, ist die zusammenhängende Fläche des NVE. Während bei SimMud feste Regionen existierten, die das Konzept des AOI nicht voll widerspiegeln, kann bei VAST mit Hilfe der Voronoidiagramme der AOI effektiv umgesetzt werden. Zudem baut VAST nicht auf einem existierenden P2P-Netz als *Overlay* auf, sondern nutzt direkt die Voronoidiagramme für die Netzwerkkommunikation. Dadurch können minimale Reaktionszeiten erreicht werden, deren Kosten bei Aktionen innerhalb des AOI bei $O(1)$ liegen.

Da sich das globale Voronoidiagramm ständig verändert, müssen die lokalen Diagramme bei den Nutzern effektiv berechnet werden können. Der schnellste bekannte Algorithmus bewältigt dies in $O(n \log n)$. Aufgrund der lokalen Berechnung der Voronoidiagramme ändert sich mit steigender Nutzerzahl des NVE nur wenig an der Anzahl der Nachbarn und somit der Größe der Berechnungen. Bis auf Situationen, bei denen sich sehr viele Nutzer auf engem Raum befinden, können die Voronoidiagramme somit in ausreichender Zeit berechnet werden.

5 Fazit

In dieser Arbeit wurden Konzepte zur Erzeugung von hoch skalierbaren NVEs vorgestellt. Die Konzepte basieren darauf P2P-Netze für die Kommunikation im NVE

zu verwenden, da diese sowohl ohne kostspielige Infrastruktur aufgebaut werden können und zudem gut mit der Anzahl der Nutzer skalieren. Um die Last bei den einzelnen Netzwerkknoten gering zu halten wurde das Verfahren des Interessenmanagement vorgestellt, dass die Kenntnisse der Netzwerktopologie bei den Nutzern auf lokale Inhalte reduziert.

Die hier vorgestellten Implementationen zeigten vielversprechende Ansätze zur Konstruktion hoch skalierbarer NVEs. Beide wiesen allerdings verschiedene Schwächen auf, die zunächst behoben werden müssen, bevor solche Systeme produktiv eingesetzt werden können.

Literatur

- [1] BACKHAUS, H. & S. KRAUSE: *Voronoi-based adaptive scalable transfer revisited: gain and loss of a Voronoi-based peer-to-peer approach for MMOG*. In *NetGames '07: Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*, Seiten 49–54, New York, NY, USA, 2007. ACM.
- [2] CASTRO, M., P. DRUSCHEL, A. MARIE KERMARREC & A. ROWSTRON: *SCRIBE: A large-scale and decentralized application-level multicast infrastructure*. *IEEE Journal on Selected Areas in Communications (JSAC)*, 20, 2002.
- [3] HU, S.-Y. & G.-M. LIAO: *Scalable peer-to-peer networked virtual environment*. In *NetGames '04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, Seiten 129–133, New York, NY, USA, 2004. ACM.
- [4] ICKING, C., R. KLEIN, P. KÖLLNER & L. MA: *VoroGlide*. <http://www.pi6.fernuni-hagen.de/GeomLab/VoroGlide/>, 2001.
- [5] KNUTSSON, B., H. LU, W. XU & B. HOPKINS: *Peer-to-Peer Support for Massively Multiplayer Games*. In *INFOCOM*, 2004.
- [6] ROWSTRON, A. I. T. & P. DRUSCHEL: *Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems*. In *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, Seiten 329–350, London, UK, 2001. Springer-Verlag.