

Mobiilien moninpelien reaaliaikaisuuden tavoittaminen matkapuhelinverkoissa

Johannes Niemelä

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos/tko
Pro gradu -tutkielma
Ohjaaja: Roope Raisamo
6.6.2008

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos/tko
Johannes Niemelä: Mobiilien moninpelien reaaliaikaisuuden tavoittelemisen
matkapuhelinverkoissa
Pro gradu -tutkielma, 91 sivua
Kesäkuu 2008

Tiivistelmä

Kartoitan pro gradu -tutkielmassani moninpelien ja matkapuhelinten liittoa, eli moninpelattavia mobiilipelejä. Tällaiset pelit käyttävät matkapuhelinverkkoja, jotka kuitenkin aiheuttavat rajoitteita reaaliaikaisen moninpelin saavuttamiselle. Työni kartoittaa matkapuhelinverkkojen ominaisuudet, jotka ovat tärkeitä moninpeleille. Mobiilimoninpelin resursseja rajoittaa kaksi tahoja, joista toinen on matkapuhelinverkot ja toinen itse mobiilipäätelaite, jolla tarkoitetaan tässä tutkielmassa matkapuhelinta. Tutkielma keskittyy tarkastelemaan verkkojen resurssirajoitusten kompensatiotekniikoiden soveltuvuutta mobiileille moninpeleille. Esittelen myös moninpeli konseptin, joka on suunniteltu havainnollistamaan sosiaalisen pelaamisen mahdollisuuksia matkapuhelimilla. Konseptia olen testannut prototyypillä, joka toimii käyttäen nykyisiä matkapuhelimia ja -verkkoja.

LYHENTEET

2G 2nd Generation

3G 3rd Generation

AMPS Advanced Mobile Phone Service

API Application Programming Interface

ARP Autoradiopuhelin

AuC Authentication Center

BCS Base Station Controller, tukiasemaohjain

bps Bits per second, bittiä sekunnissa.

BTS Base Transceiver Station, tukiasema

CDMA Code division multiple access

DIA Distributed Interactive Application

DIS Distributed Interactive Simulation

EDGE Enhanced Data rates for GSM Evolution

EDR Enhanced Data Rate

EIR Equipment Identification Register

FPS First Person Shooter. Ensimmäisen persoonan näkökulmasta kuvattu ammuskelupeli.

GB GameBoy

GBA GameBoy Advance

GBC GameBoy Color

GC GameCube

GGSN Gateway GPRS Supporting Node

GPRS General Packet Radio Service

GPS Global Positioning System

HLR Home Location Register

HSCSD High-Speed Circuit-Switched Data

HSPA High-Speed Packet Access

IEEE Institute of Electrical and Electronics Engineers

IrDA Infrared Data Association

ITU International Telecommunication Union

Java ME Java Platform, Micro Edition

KVM Kilo Virtual Machine

LAN Local Area Network

MAN Metropolitan Area Network

Mbps Megabits per second, megabittiä sekunnissa

ME Mobile Equipment

MIDP Mobile Information Device Profile

MP3 MPEG-1 Audio Layer 3

MS Mobile Station

MSC Mobile Switching Center

NMT Nordic Mobile Telephone

NSS Network Switching Subsystem

PAN Personal Area Network

PDA Personal Digital Assist

PDP Packet Data Protocol

PS2 PlayStation 2

PS3 PlayStation 3

PSP Playstation Portable

PSTN Public Switched Telephone Network

RTT Round Trip Time

SIM Subscriber Identity Module

SKU Stock Keeping Unit

TDD Time Division Duplexing

ULP Bluetooth Ultra Low Power Bluetooth

UMTS Universal Mobile Telecommunications System

VLR Visitor Location Register

WCDMA Wideband Code Division Multiple Access

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless Local Area Network

SISÄLLYS

Lyhenteet	iii
1 Johdanto	1
2 Pelien ja mobiililaitteiden määrittely	6
2.1 Peli	6
2.2 Mobiilipeli	8
2.3 Verkkomoninpelit	14
2.4 Mobiililaitteet	18
3 Langattomat verkot	23
3.1 Verkkoteknologiat	23
3.2 Pelien erityisvaatimukset verkoille	27
3.3 Matkapuhelinverkot	31
3.3.1 Kaista	33
3.3.2 Viive, huojunta ja häviö	35
3.4 Mobiilipelien verkkoarkkitehtuurit	37
4 Pelin tilan synkronointi	40
4.1 Konsistenssi, vaste ja skaalautuvuus	41
4.2 Resurssirajoitusten kompensatiotekniikat	49
4.2.1 Protokollan optimointi	49
4.2.2 Liikkeen ennustaminen	52
4.2.3 Paikallinen havainnointisuodatus	56
4.2.4 Kiinnostuksen alueen mukainen suodatus	60
4.2.5 Synkronoitu simulaatio	62
4.3 Kompensatiotekniikoiden yhteenveto	63
5 Flostere	67
5.1 Konsepti	67
5.2 Prototyyppi	70
5.3 Mobiilipelien pelisuunnittelumallit	75
5.4 Jatkokehityssuunnitelma	76
6 Pohdintaa ja yhteenveto	79
Lähdeluettelo	81

1 JOHDANTO

Matkapuhelimet ovat kaikkialla läsnä olevaa tietotekniikkaa. Lähes jokaisella läntisten teollisuusmaiden kansalaisella on matkapuhelin. Myös kehittyvissä maissa, kuten Kiinassa ja Intiassa, puhelimet yleistyvät nopeasti. Matkapuhelimia pidetään hyvin tarpeellisina välineinä. Matkapuhelimien tärkeimpiin ominaisuuksiin kuuluu kyky muodostaa tiedonsiirtoyhteyksiä - puhe- tai datayhteyksiä. Tämän lisäksi matkapuhelinten ominaisuuksia laajentaa niiden kyky suorittaa käyttäjän asentamia sovelluksia. Yksi sovellusalue on mobiilit pelit, joista *ajanvietepelit* (casual games) ovat iso osa. Vaikka matkapuhelimet ovat alusta asti mahdollistaneet yhteyksien muodostamisen matkapuhelinverkkojen kautta, eivät mobiilit moninpelit ole vielä yleistyneet. Tämä johtuu osittain matkapuhelinverkkojen ominaisuuksista, osittain pelikehittäjistä ja osittain itse käyttäjistä.

Tässä pro gradu -tutkielmassa kartoitan matkapuhelimien ja matkapuhelinverkkojen aiheuttamia haasteita mobiilien moninpelien kehittämiseksi. Haasteiden ja rajoitusten tunnistamisen jälkeen selvitan, millä tavoin näitä ongelmia voi lähestyä tai kiertää ohjelmistoteknisillä ja pelisuunnittelullisilla ratkaisuilla. Tutkielman tavoitteena ei pelkästään ole esitellä ongelmia, vaan samalla selvittää mahdollisuuksia ja vahvuuksia, joita matkapuhelimet ja -verkot tarjoavat mobiileille moninpeleille. Tutkielman aihealue perustuu täten tietoverkkoihin ja ohjelmistoarkkitehtuureihin sekä pelisuunnitteluun. Kohdeilmionä on ajanvietteelliset mobiilit moninpelit (casual multiplayer mobile games).

Taloudelliselta kannalta tarkasteltuna ajanvietemobiilipelien markkinoiden arvioidaan olevan yli miljardi dollaria ja pelaajia on maailmanlaajuisesti 180 miljoonaa [MarketResearch.com, 2006]. Maailmanlaajuiset mobiilipelimarkkinat olivat 3,1 miljardia dollaria vuonna 2006 ja niiden ennustetaan kasvavan 17,6 miljardiin dollariin vuoteen 2011 mennessä [Gibson, 2006a]. Tuoreemman tutkimuksen valossa vuoden 2007 mobiilipelien lataus- ja tilauslaskutukseen perustuvien markkinoiden arveltiin olevan hiukan alle 5 miljardia dollaria. Samassa tutkimuksessa todetaan myös, että markkinoiden arvellaan kasvavan lähes 16 miljardiin dollariin vuoteen 2012 mennessä. [Holden, 2007].

Markkinat jakautuvat 38 prosenttisesti Aasiaan, 31 prosenttisesti Eurooppaan, 22 prosenttisesti Pohjois-Amerikkaan ja 9 prosenttia muualle [Gibson, 2006a]. Aasian absoluuttinen myynninosuus tulee entisestään nousemaan, koska Kiinassa ja Intiassa on meneillään nopea matkapuhelimien yleistyminen. Eurooppa ja Pohjois-Amerikka ovat sen sijaan jo läpi penetroituneita matka-

puhelintihyeksiltään. Esimerkiksi Suomessa on noin 5,96 miljoonaa matkapuhelinliittymää noin 5,3 miljoonaa asukasta kohden [Viestintävirasto, 2007, STT, 2008]. Vaikka matkapuhelin löytyykin jo lähes jokaiselta läntisen teollisuusmaan kansalaiselta, ennustetaan peleistä saatavan tulon silti noin kolminkertaistuvan Euroopassa ja Pohjois-Amerikassa vuoteen 2011 mennessä, kun se Aasiassa samassa ajassa noin tuplaantuu [Tercek, 2007a].

Janne Vihinen [2007] väittää, että maailmanlaajuiset pelimarkkinat olisivat vuosituloiltaan arvokkaampia kuin elokuvamarkkinat. Tämä ei kuitenkaan pidä paikkaansa ja on aika yleinenkin harhakäsitys markkinoiden suhteesta. Tämän suuntaisilla lausunnoilla on haluttu liioitella pelien asemaa merkittävänä teollisuutena. Merkittävät markkinat ja teollisuus ne ovatkin mutta eivät ylitä elokuvan tuottoa. Elokuvien lipunmyynnin voi pelimyynti peitotakin Yhdysvalloissa, mutta esimerkiksi elokuvien jälleenmyynti sekä vuokraus video- ja DVD-levityksenä huomioon ottaen jäävät pelit vielä paljon jälkeen. Elokuvien kaikenkertyneet tuotot huomioiden, ne ovat noin kolme kertaa suuremmat kuin pelien tuotot. [Kerr, 2006, Dovey and Kennedy, 2006].

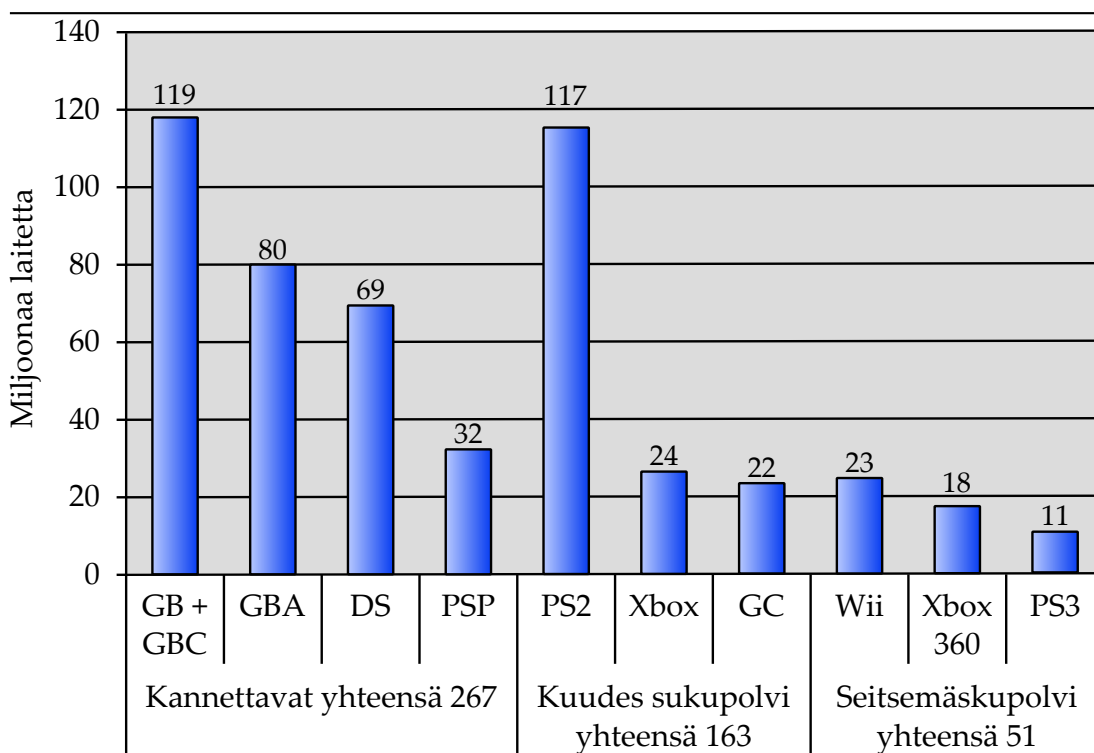
Matkapuhelimien on sanottu olevan nopeimmin ja laajimmalle yleistynyt kuluttajatuote. Niitä myydään vuosittain enemmän kuin autoja ja tietokoneita yhteensä [Vihinen, 2007]. Väite on helppo uskoa, ainakin kun verrataan matkapuhelimien myyntimääriä tietokoneiden ja pelikonsolien myyntimääriin. Jo vuodesta 1998 alkaen matkapuhelimia on myyty joka vuosi maailmanlaajuisesti yli 200 miljoonaa kappaletta. Nykyisin matkapuhelimia myydään vuosittain yli 1,2 miljardia kappaletta. [IDC, 2007, IDC, 2008a, Gartner, 2008a, Leidenius, 2008]. Tietokoneita myydään tänä vuonna noin 300 miljoonaa kappaletta ja vuonna 2012 tietokoneiden myynnin arvellaan olevan noin 430 miljoonaa kappaletta. [Gartner, 2008b, IDC, 2008b]. Matkapuhelimia myydään vuosittain siis yli nelinkertainen määrä verrattuna tietokoneisiin.

Miten pelikonsolien levinneisyys sitten suhteutuu matkapuhelimiin? Myyntimäärät maailmanlaajuisesti selviävät kuvasta 1.1. Uusimman sukupolven konsolien myynnin arvellaan olevan tähän päivään mennessä yhteensä hieman yli 50 miljoonaa yksikköä. Tästä määrästä Nintendon Wii vie noin 23, Microsoftin Xbox 360 noin 18 ja Sony'n PlayStation 3 (PS3) noin 11 miljoonaa yksikköä. Yhteenlaskettu määrä on kuitenkin vielä noin 100 miljoonaa jäljessä aikaisemman sukupolven konsolien myynnistä, jossa ryhmää johtaa selvästi PlayStation 2 (PS2) 117 miljoonan myynnillään. Xboxia on myyty noin 24 ja Nintendon GameCubea (GC) noin 22 miljoonaa kappaletta.

Yllättäen kannettavat käsikonsolilaitteet ovat myyneet enemmän kuin kuu-

dennen ja seitsemännen sukupolven konsolit yhteensä. Käsikonsolien yhteenlaskettu myynti on noin 270 miljoonaa kappaletta. Luvussa on toki mukana vanhemmat Nintendon käsikonsolit, kuten GameBoy (GB) ja sen Color-malli (GBC) sekä Advance-malli (GBA). Silti uudemmat käsikonsolitkin, kuten Nintendon DS ja Sonyn Playstation Portable (PSP), ovat yhteensä myyneet kaksinkertaisesti sen, mitä uusimman sukupolven konsolit ovat myyneet yhteensä. [VG Chartz, 2008].

Käsikonsolilaitteiden suosio osaltaan heijastelee potentiaalia, joka matkapuhelimilla pelilaitteina ja mobiilipelaamisella on. Vielä kun laskuihin otetaan matkapuhelinten huima määrä verrattuna tietokoneisiin ja kaikkiin konsoleihin yhteenlaskettunakin, on mobiilipelaamisella todella potentiaalia – ainakin laitteiden yleisyyden perusteella. Tähän samaan asiaan on kiinnittänyt huomiota I-playn toimitusjohtaja David Gosen kyseenalaistaessaan pelien todellisten markkinoiden olevan matkapuhelimilla konsolien sijaan [Gibson, 2006b].



Kuva 1.1: Konsolien myyntimäärät maailmanlaajuisesti

Toisen sukupolven matkapuhelimien käyttäjiä arvellaan olevan 2,6 miljardia ja kolmannen sukupolven 400 miljoonaa. Yhteensä matkapuhelimen käyttäjiä arvellaan tänä vuonna olevan siis 3 miljardia. [MarketResearch.com, 2008].

Näistä pelejä ladanneita pelaajia arvellaan olevan kuitenkin vain 7,5 prosenttia. Tämä tekee pelaajien määräksi 225 miljoonaa. [Holden, 2007]. Iso-Britanniassa lataajien määrän on tutkittu olevan 4,8 prosenttia ja Yhdysvalloissa vain 3,3 prosenttia. Neljännes matkapuhelimien omistajista kuitenkin pelaa kuukausittain puhelimellaan. Pelatuista peleistä suuri osa on puhelimeen esiasennettuja pelejä. [Elliott, 2008]. Vaikka pelien lataajien määrä vaikuttaa pieneltä koko matkapuhelimen käyttäjien määrään verrattuna, on se vertailukelpoinen konsolien määriin tarkasteltuna. Joka neljännen matkapuhelimen käyttäjän pelaaminen kuukausittain tarkoittaa jopa 750 miljoonaa mobiilipelaajaa.

Tampereen yliopiston Hypermedialaboratorion pelikulttuurista Suomessa luotaava tutkimus [Kallio *et al.*, 2007] tukee tätä näkemystä. Tutkimuksen mukaan vastaajista lähes kuusi kymmenestä oli pelannut mobiilipeliä jossain elämänsä vaiheessa. Täten se oli suosituin digitaalisen pelaamisen muoto, sillä sen taakse jäivät niin tietokone-, konsoli- kuin käsikonsolipelaaminenkin. Tulosta osaltaan selittää Suomen matkapuhelinlaitetekannan laajuus. Tilanne kuitenkin muuttuu tutkimuksen mukaan tarkasteltaessa viimeisen kuukauden aikana tapahtunutta pelaamista. Viimeisen kuukauden aikana tietokonepelejä oli tutkimuksen mukaan pelannut noin joka viides vastaaja, mobiilipelejä hieman alle joka kuudes ja konsolipelejä noin joka kymmenes. Näiden tutkimustulosten perusteella on oikeutettua sanoa, että mobiilipelaaminen on merkittävä osa digitaalista pelaamista. Valitettavasti Kallion *et al.* tutkimukselle ei vielä ole kansainvälisiä vertailukohtia, mutta niitä toivottavasti tulee, koska tutkimus oli tarkoitettu ensimmäiseksi osaksi kansainvälistä pelikulttuurien kartoitusta.

Laitetekannan ja käyttäjämäärän takia pienikin prosentuaalinen kasvu nostaa mobiilipelaajamäärän aivan uudelle tasolle konsolipelaajiin ja tietokonepelaajiin verrattuna. Mobiilipelaamisen määrän on myös todettu olevan kasvussa, vaikkakin kasvu on ollut hidasta kasvua [Elliott, 2008]. Mobiilipelaaminen on myös ainoa vaihtoehto useilla nopeasti kehittyvillä alueilla, kuten Intiassa, jossa tietokoneita, konsoleita ja kiinteitä laajakaistaliittymiä ei ole kaikkien ulottuvilla, mutta matkapuhelimia ja niiden verkkoja on [Holden, 2007]. Kehittyneemmällä alueilla on todettu lasten pelaavan mobiilipelejä vaikka he eivät pelaisikaan tietokoneilla tai konsoleilla [Vincent *et al.*, 2005].

Tarkastelen luvussa 2 pelin, mobiilipelin ja verkkomoninpin määritelmiä sekä mobiililaitteita kirjallisuuskartoituksella. Luvuissa 3 ja 4 kirjallisuuskartoitus jatkuu matkapuhelinverkkojen ominaisuuksien kartoittamiseen ja resurssirajoitusten ohjelmistoteknisiin kiertomenetelmiin. Luku 5 esittelee tut-

kielman konstruktivisena osuutena toteutetun sosiaalisen mobiilipeli-
tyypin. Viimeisessä luvussa vedetään yhteen ja pohditaan tutkielman aiheita.

2 PELIEN JA MOBIILILAITTEIDEN MÄÄRITTELY

Tässä luvussa esittelen ensiksi pelin määritelmän. Pelin määrittelystä edetään todellisen mobiilipelin luonteen tarkasteluun ja sitä kautta verkkomonipeleihin. Viimeiseksi käsittelen tässä luvussa mobiililaitteita ja erityisesti matkapuhelimia merkittävimpänä ryhmän edustajana.

2.1 Peli

Pelin (game) määritelmää on lähestytty monesta eri näkökulmasta. Yksi lähestymistapa on ollut selventää pelin ja *leikin* (play) välisiä suhteita. Frans Mäyrän [2008] mukaan kaiken kattavan määritelmän muodostaminen peleille on vaikea tehtävä. Teoksessaan *An Introduction to Games Studies* Mäyrä kertoo myös leikkien määrittelemisen olevan vaikeaa. Hän jatkaa, että on kyseenalaista, voiko mikään yksittäinen pelien määritelmä olla ikinä niin täydellinen, ettei sitä voisi kumota jostain tietystä suunnasta tarkasteltuna. Vaikka pelin määrittely on hankalaa, se kannattaa Mäyrän mielestä tehdä ainakin analyysiä helpottamaan. Hänen mielestään analyysi helpottuu, koska muodollisesti tarkka määritelmä helpottaa ymmärrystä siitä, mitä tietyt pelit ovat ja mitä ne tarkoittavat.

Koska tutkielmani tarkoitus ei ole selventää pelin ja leikin välisiä eroja eikä tarkemmin ottaa kantaa lukuisiin pelin määritelmiin, käytän pelin määrittelyssä Salenin ja Zimmermanin kirjaa *Rules of Play* [2003]. He käyvät kirjassa läpi kahdeksan eri määritelmää¹. Heidän valitsemansa määritelmät ovat keskeisiä alan kirjallisuudessa ja ne ottavat hyvin huomioon pelien moninaisen kirjon ja tarkastelevat ilmiötä monelta suunnalta. *Rules of Play* on myös laajalti tunnettu ja käytetty teos. Mäyrän kirjan käsittelemät pelien määritelmät perustuvat lähes samoihin lähteisiin kuin Salenin ja Zimmermanin. Mäyrä esittää itsekin Salenin ja Zimmermanin määritelmän sekä tämän lisäksi uudempana Jesper Juulin määritelmän Juulin kirjasta *Half-Real: Video Games Between Real Rules and Fictional Worlds*. Eri määritelmien analyysin pohjalta Salen ja Zimmerman muodostavat oman määritelmänsä, joka kuuluu Tony Mannisen [2007] suomenoksen mukaan seuraavasti:

Peli on järjestelmä, jonka puitteissa pelaajat pyrkivät sääntöjoukon määrittä-

¹ Määritelmien kirjoittajat ovat D. Parlett, C. C. Abt, J. Huizinga, R. Caillois, B. Suits, C. Crawford, G. Costigyan sekä E. Avedon ja B. Sutton-Smith.

mänä ja ohjaamana ratkaisemaan keinotekoisen konfliktin, jonka ratkaisusta seuraa mitattavissa oleva lopputulos.

Salenin ja Zimmermanin määritelmä sisältää kuusi perusajatusta: järjestelmä, pelaajat, keinotekoinen, konflikti, säännöt ja mitattavissa oleva lopputulos. Tämä määritelmä ei kuitenkaan tee eroa digitaalisten ja ei-digitaalisten pelien välille. Salen ja Zimmerman eivät näe tarvetta erikseen määritellä digitaalista peliä. He toteavat sen sijaan, että fyysinen media, eli alusta, on tärkeä elementti peleissä, jos pelit nähdään järjestelminä. Pelin alusta ei heidän mielestään kuitenkaan riitä selittämään koko peliä. Pelejä pelataan eri alustojen avustamina. Palaan pelien fyysiseen mediaan mobiilipelejä käsittelevässä kohdassa 2.2. Salen ja Zimmerman luettelevat useita eri alustoja, joilla on digitaalisia pelejä. Näitä ovat tietokoneet, televisioon liitetyt pelikonsolit, käsikonsolit, yhden pelin käsilaitteet (elektroniikkapelit), matkapuhelimet, kämmenmikrot, pelihallien ja huvipuistojen pelit. Tätä listaa voi jatkaa esimerkiksi digisovittimilla, MP3-soittimilla ja jopa digikameroilla.

Lueteltujen laitteiden välille voi mielestäni tehdä eron. Erottavana tekijänä on se, onko laite ensisijaisesti tehty pelaamista vai jotain muuta toimintaa varten. Esimerkiksi tietokoneella ja matkapuhelimella on muu käyttö suunniteltu ensisijaiseksi. Nämä laitteet kuitenkin mahdollistavat myös pelaamisen yhtenä toimintonaan. Sen sijaan pelikonsolit ja elektroniikkapelit ovat nimen omaan pelilaitteita, vaikka pelikonsoleilla voi nykyisin tehdä monia muitakin asioita. Raja on kuitenkin hämärtyvässä. Asia riippuu ennen kaikkea käyttäjän katsontakannasta eli mikä on käyttäjän intentio.

Tietokonepelin määritelmiä on pelin määritelmien tapaan useita. Määrittelyn hankaluutta kuvaa videopelin erilaiset määritelmät, joita yhdestä kirjasta löytyy jopa 71 kappaletta [Newman and Simons, 2004]. Videopeli ja tietokonepeli ovat toisiaan lähellä olevia käsitteitä. Jouni Smed ja Harri Hakonen [2003] eivät erikseen määrittele digitaalista peliä, mutta he määrittelevät tietokonepelin peliksi, joka toteutaan tietokoneohjelman avulla. He toteavat määrittelyn olevan sen verran väljä, ettei se edellytä koko pelin tapahtuvan tietokoneessa. Esimerkkinä he käyttävät shakkia, jossa peliä voidaan pelata oikealla laudalla tai näytöllä huolimatta siitä, onko vastustaja tietokoneohjelma. Keskeistä Smedin ja Hakosen määritelmässä on tietokoneohjelma. Täten pelkkä fyysisen tietokonelaitteiston käyttäminen pelielementtinä ei tee pelistä tieto-

konepeliä. Esimerkiksi tietokonelaitteiston, vaikkapa keskusyksikön heittäminen mahdollisimman pitkälle ei ole siten tietokonepeli.

Smedin ja Hakosen määritelmä sopii tämän tutkielman tarkoitukseen muutamalla muokkauksella. Ensinnäkin, tietokone korvataan matkapuhelimella, josta saadaan mobiilipelin määritelmä. Toiseksi, määritelmään lisätään osuus, joka määrittää pelaajan ja pelin välisen syötteiden sekä tulosteiden vaihdon tapahtuvaksi matkapuhelimen avulla. Syötteet voivat olla esimerkiksi näppäinten painalluksia, äänikomentoja tai vaikka laitteen liikuttamista. Laitteen liikuttaminen voi tapahtua paikallisesti kallistemalla, kääntämällä tai heiluttamalla. Laaja-alaisempi liikuttaminen käsittää laitteen kuljettamisen rakennuksen tai kaupungin sisällä paikasta toiseen tai jopa kaupunkien, maiden ja mantereiden välisen kuljettamisen. Näin laajassa liikkumisessa on kyse laitteen ja sitä kautta pelaajan paikantamisesta.

Vastaavasti tulosteet eivät ole rajoittuneet vain näytöllä näkyvään grafiikkaan, vaan voivat sisältää myös muita tulosteita, kuten esimerkiksi ääntä tai tuntopalautetta. Syötteistä ja tulosteista puhuminen on vanhanaikainen näkökanta. Jouhevampaa olisi puhua vuorovaikutuksesta, *interaktiosta*, jossa pelaaja ja peli reagoivat jatkuvasti toisiinsa. Smedin ja Hakosen määrittely ei itse asiassa eroa paljoakaan Salenin ja Zimmermanın määrittelystä. Jälkimmäisten määrittelyyn tarvitsee tämän tutkielman tarkoitusta palvellakseen lisätä, että pelaajat osallistuvat keinotekoiseen konfliktiin matkapuhelimella toimivaa ohjelmaa käyttäen.

2.2 Mobiilipeli

Edellä esitetty mobiilipeleille soveltuva määritelmä on hieman yksipuolinen. Määritelmän mukaan mikä tahansa digitaalinen peli suoritettuna matkapuhelimessa riittää tekemään siitä mobiilipelin. Osittain tämä pitääkin paikkansa, mutta tutkittaessa mobiiliutta voidaan huomata seikkoja, jotka jäävät suoraviivaisessa määrittelyssä varjoon. Yksipuoliseksi määritelmän tekee katsontakanta, joka tarkastelee mobiilipeliä teknisemmästä näkökulmasta.

Seuraavaksi tarkastelen mobiilipelin määrittelemistä huomioiden mobiilipelialustan luonnetta ja sitä, mitä se tarkoittaa pelin suunnittelulle. Barbara Grüter ja Miriam Oks [2007] sanovat mobiilipelien olevan kuin muita pelejä, mutta samalla erilaisia. Mikä on siis mobiiliuden luonne?

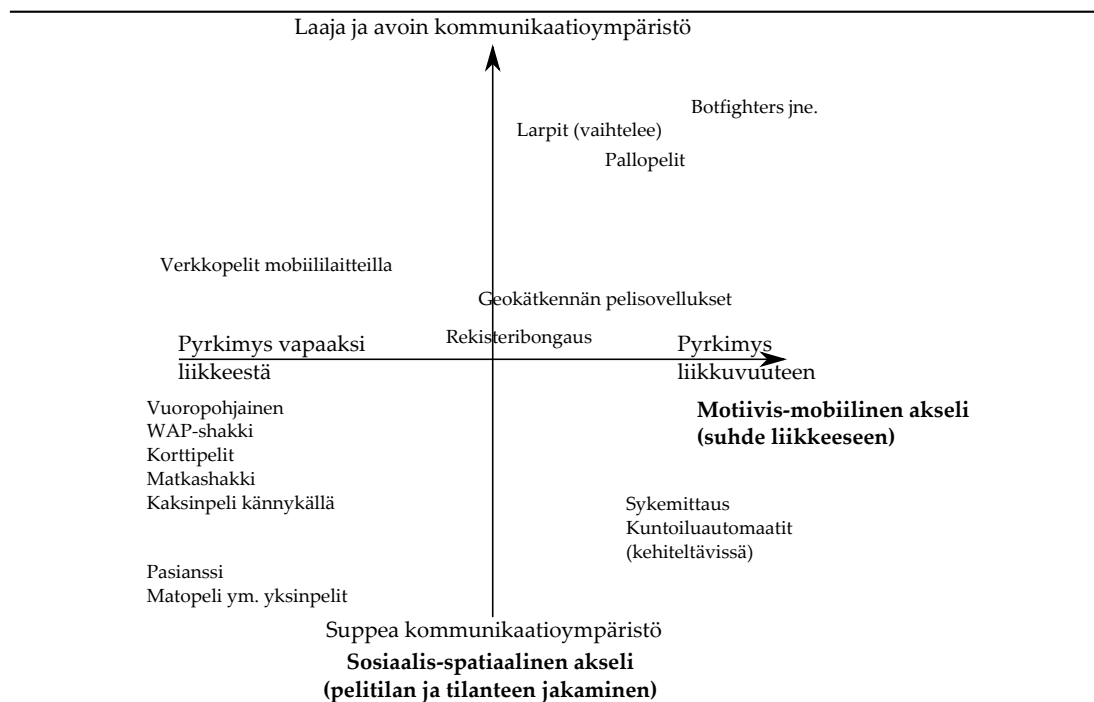
Mobiili tarkoittaa sananmukaisesti liikkuvaa, liikuteltavaa tai siirrettävää. Matkapuhelimet eivät ainakaan vielä liiku itsekseen, eli jonkun täytyy niitä

siirtää tai liikuttaa. Liikkuminen on käyttäjän tehtävä. Joko käyttäjä liikkuu itse, liikuttaa laitetta omin voimin tai käyttää hyväkseen jotain kulkuneuvoa. Aki Järvinen [2002] on pureutunut liikuttamiseen artikkelissaan "Milloin pelistä tulee mobiili?". Järvinen etsii mobiilipeliä "mobiileimmillaan" tarkoittaen todellista mobiilipeliä eikä pelkästään peliä mobiililaitteessa. Järvisen mielestä todellisen mobiilipelin luonne vaatii, että pelaaja liikkuu reaali maailmassa pelataksaan peliä. Hänen painotuksensa on ensimmäisellä sanalla mobiili; pelaaja liikkuu. Järvinen sanoo pelin olevan todella mobiili vasta, kun se sisältää elementtejä, jotka ottavat huomioon matkapuhelimen käyttötarkoitukset ja -tilanteet sekä käyttökulttuurien vivahteet. Palaan näihin seikkoihin tarkemmin kohdassa 2.4 sivulla 18.

Smedin ja Hakosen tietokonepelin määritelmästä mukailtu mobiilipelin määritelmä puolestaan tarkastelee asiaa peliä suorittavan pelilaitteen suunnasta. Järvisen mielestä tällainen laitelähtöinen määrittely ei anna koko kuvaa tilanteesta. Hän jatkaa, että mobiilipelejä on luokiteltu ja määritelty pelivälineen, ohjelmointikielen tai viestintäprotokollan kautta. Esimerkiksi Järvinen antaa määritelmän, jonka mukaan mobiilipeli on peli, jota pelataan mobiililaitteella ja luokituksen pelejä ovat tekstiviesti-, WAP-, GPS- tai GSM-solupaikannus- ja Java-pelit.

Myös Jaakko Suominen [2003] on ottanut mobiiliuden tarkasteluun artikkelissaan "Liikkeelle vai liikkeestä pois? - Mobiilipelejä luokittelemassa". Hänen mielestään liikuteltavuuden, liikkuvuuden ja käyttötilanteen vaihtelu on mobiiliuden ydin. Suominen esittelee toisenkin avainkäsitteen, jonka mukaan mobiilipelejä voidaan tarkastella. Tämä käsite on pelikokemuksen sosiaalisuuden aste. Näiden avainkäsitteiden pohjalta hän muodostaa kaksiakselisen kuvion. Tälle kartalle voidaan hänen mukaansa asemoida mobiilipelit sekä muitakin pelejä, kuten kuvasta 2.1 ilmenee. Vaaka-akselia Suominen nimittää motiivis-mobiiliseksi akseliksi ja pysty-akselia sosiaalis-spatiaaliseksi eli paikan ja sijainnin akseliksi. Motiivis-mobiiliselä akselilla Suominen yrittää tavoittaa Järvisen luonnehtimaa todellista mobiiliutta. Tässä akselissa ääripäinä ovat pyrkimys liikkeeseen pelaamalla sekä liikkeestä vapautumisen pyrkimykset. Liikkeestä vapautuminen on Suominen mukaan joutilaan ajan hetken täyttämistä.

Mobiilipelien markkinoinnissa onkin paljon käytetty aihetta, jossa pelin kerrotaan sopivan täyttämään niin sanotut turhat tai tyhjät ajan hetket pelaajan odotellessa jotain merkittävämpää tapahtumaa. Tätä seikkaa tulisi tutkia tarkemmin, jotta tiedettäisiin paremmin mobiilipelaajien pelisessoiden pituus



Kuva 2.1: Mobiilipelien liikkuvuus ja sosiaalisuus Jaakko Suomisen mukaan

ja konteksti eikä oletettaisi tämän asian automaattisesti olevan näin. Nokian Nielsen Entertainmentillä teettämän tutkimuksen [Nokia, 2006] mukaan keskiverto mobiilipelisessio kestää 28 minuuttia vaihdellen kiinalaisten pelaajien 18,6 minuutista intialaisten 38,9 minuuttiin. Tutkimuksessa oli 300 vastaajaa kustakin kuudesta eri maasta (Kiina, Saksa, Intia, Espanja, Thaimaa ja Yhdysvallat) eli yhteensä 1800 vastaajaa.

Tutkimuksen tulosta vääristää pieni otos maittain sekä se, että vastaajat ovat painottuneita mobiilipelaajiin ja henkilöihin, jotka voisivat ajatella pelaavansa mobiilipelejä. Suuntaa tuloksesta kuitenkin saadaan. Pelisession pituus on yllättävän pitkä verrattuna ajatukseen muutaman minuutin tyhjän hetken kuluttamisesta. Luonnollisesti markkinointi ja todellinen tilanne eivät yleensä suoraan vastaa toisiaan, sillä tulee muistaa, että markkinoinnissa on kyse myös tarpeiden ja halujen luomisesta.

Suomisen mukaan kuvan 2.1 sosiaalis-spatiaalinen akseli selventää mobiilipelin vuorovaikutuksen astetta. Suomisen selittää vuorovaikutuksen asteen kuvaavan sitä, onko interaktio ensisijaisesti vain laitteen ja pelaajan välillä. Hän täydentää, että akseli kuvaa myös sen, onko interaktio välitteistä ja hidadista vai ensisijaista ja välitöntä.

Järvisen mielestä todellinen mobiilipeli ottaa siis huomioon Suomisen kuvaileman motiivis-mobiilisen akselin ja sisältää täten elementtejä, jotka vaativat liikkuvuutta. Tällaiset pelit ovat kartalla oikeassa reunassa. Myös Grüter ja Oks [2007] kertovat mobiilipelien perustuvan pelaajan kehon liikkeelle. Suominen kuitenkin toteaa suurimman osan matkapuhelimiin esiasennetuista tai ladattavista peleistä painottuvan kuvion vasempaan alakulmaan eli alueeseen, jossa ei pyritäkään liikkuvuuteen. Vaarana on, että karttaa lukiessa ajatellaan joiltakin alueilta löytyvien pelien olevan parempia tai edistyneempiä kuin toisella alueella olevien. Suominen kuitenkin painottaa, että hänen tarkoituksensa ei ole kartalla arvottaa tai asettaa mobiilipelejä paremmuusjärjestykseen. Hänen tavoitteenaan on ollut suunnata ajattelua mobiilipelien eri piirteisiin. Tämän tutkielman pääpaino on tutkia sosiaalis-spatiaalista akselia eli pelitilan ja tilanteen jakamista. Suomisen kuvion alalaidassa ovat suppean kommunikointiympäristön pelit ja ylälaidassa laajaa ja avointa kommunikaatioympäristöä tarjoavat pelit.

Palaan hetkeksi laitelähtöiseen pelien luokitteluun, joka oli Järvisen mielestä hieman ontuva. Juul [2005] pohtii tätä samaa asiaa pelien siirtämisestä alustalta toiselle. Alusta eli pelialusta tarkoittaa laitetta, alustaa tai fyysistä mediaa, millä peliä pelataan. Esimerkiksi kortti- tai konsolipelien alustat ovat kortteja tai pelikonsoleita. On kuitenkin muistettava, että peli itsessään voidaan käsitellä myös mediaksi [Wolf, 2002]. Tämä hankaloittaa asian selkeää esittämistä, mutta käytän tässä tutkielmassa alustaa tarkoittamaan sitä mediaa, jolla peliä pelataan.

Juul puhuu alustalta toiselle siirrettävistä peleistä (transmedial games). Tämä tarkoittaa sitä, että myös hän näkee Salenin ja Zimmermanın lailla pelin fyysisen median muuttamisen joksikin toiseksi. Hyvin tunnettu esimerkki on Juulin mukaan shakki, jota voidaan pelata laudalla, tietokoneilla tai jopa sokkona. Viimeinen tarkoittaa pelimuotoa, jossa ei ole laitetta eikä lautaa avustamassa pelaajaa näkemään pelitilanteen, vaan pelaaja pelaa peliä pelkästään päässään. Juul esittelee kaksi tapaa, joilla peli on mahdollista toteuttaa toisella alustalla. Nämä ovat *toteutus* (implementation) ja *sopeuttaminen* (adaptation). Edellä muutamankin kerran mainittu shakki on toteutus siirrettäessä se laudalta tietokoneelle, koska siinä on mahdollista osoittaa jokaisen laudalla esiintyvän pelitilan täsmällinen vastaavuus tietokoneella toteutettuun shakkiin. Juulin mukaan urheilulajien, kuten jalkapallon, tietokonepeliversio on sen sijaan sopeutus, koska kaikkea oikean maailman fysiikkaa ei ole mahdollista mallintaa tietokonepeliin. Hän jatkaa, että käyttöliittymä on myös huo-

mattavasti yksinkertaistettu oikeasta maailmasta tietokoneversioon siirryttäessä. Juul selvittää, että alusta tukee pelejä kahdella eri tavalla:

1. Laskennallisesti, eli miten alusta ylläpitää sääntöjä ja päättää, mitä tapahtuu pelaajan annettua syötteen.
2. Pelitilallisesti, eli miten alusta seuraa pelitilaa.

Juulin mielestä tämä kaksijakoinen selvittäminen on välttämätöntä, jotta eri alustoja voitaisiin tarkastella. Taulukossa 2.1 esitellään Juulin listaamia alustoja ja niiden eroja. Olen lisännyt taulukon loppuun kaksi matkapuhelimella pelattavaa esimerkkiä. Ensimmäinen, korttipeli matkapuhelimessa, kuvaa Järvisen aiemmin esittelemää tilannetta, jossa tietokone- tai tässä tapauksessa korttipeli on vain siirretty pelimedialta toiselle. Siirros on toteutusmuotoinen alustan vaihdos. Taulukon viimeinen rivi esittää tilannetta Järvisen tarkoittamassa todellisessa mobiilipelissä. Esimerkiksi on valittu Suomisen mobiilipelien liikkuvuus ja sosiaalisuus -kartan (ks. kuva 2.1) oikeaan yläkulmaan sijoittuva *Botfighters*. Sen pelitila on osittain pelaajan kehoon liittyvä, koska pelaajan sijaintia paikannetaan. Paikantaminen tapahtuu matkapuhelimen ja -verkon solupaikannusmenetelmää käyttämällä. Pelitilan suhteen paikannus on ongelmakohta. Esitetäänkö asia niin, että pelaajan ja puhelimen sijainti vaikuttaa pelitilaan vai käsitetäänkö paikannus vain yhtenä syötteenä suorittimelle, joka muuttaa keskusmuistissa olevaa pelitilaa?

Historiallisesti tarkasteltuna mobiilipelit matkapuhelimissa ovat vain runsaat kymmenen vuotta vanhoja. Vuonna 1997 Nokia julkisti 6110 matkapuhelimen, jossa oli esiasennettuna matopeli (*Snake*). Ensimmäinen puhtaasti sähköpostiin perustuva peli oli Bandain *Love by Email* vuonna 1999. Samana vuonna tulivat ensimmäiset WAP-pelit, kuten Japanissa pelattu *Tsuri Baka Kibun*. Sen kehittäjä oli japanilainen Dwango. Vuoden 2000 tienoilla ilmestyivät myöskin muut ensimmäisten joukossa verkkoa käyttäneet mobiilipelit, kuten ruotsalaisen PicoFunin *LifeStylers* ja brittiläisen nGamen *Alien Fish Exchange*. Ensimmäinen ympäristöään huomioiva ja samalla episodimainen peli oli myöskin Dwangon vuonna 2000 julkaisema *Samurai Romanesque*. [Tercek, 2007a, Tercek, 2007b]. Siinä oikean ilmakehän säätömiöt vaikuttivat peliin ja esimerkiksi tuuli sekä sade hidastivat liikkumista pelimaailmassa [Järvinen, 2002].

Ensimmäinen paikannusta hyväksikäyttävä kaupallinen peli oli kuvaan 2.1:kin merkitty *Botfighters*. Sen kehittäjä oli ruotsalainen It's Alive ja peliä pelattiin muun muassa Ruotsissa, Suomessa ja Venäjällä. *Botfighters* julkaistiin

Taulukko 2.1: Pelimedioiden eroja

	Säännöt ja laskenta	Pelitila
Korttipelit	Ihmisaivot	Kortit
Lautapelit	Ihmisaivot	Pelinappulat
Sokkoshakki	Ihmisaivot	Ihmisaivot
Urheilu	Fysiikka ja ihmisaivot	Urheilijoiden kehot ja pelivälineet
Tietokonepelit	Suoritin	Keskusmuisti
Kortti- ja lautapelit pelattuna tietokoneella	Suoritin	Keskusmuisti
Urheilupelit tietokoneella	Suoritin	Keskusmuisti
Korttipeli matkapuhelimessa	Suoritin	Keskusmuisti
<i>Botfighters</i>	Suoritin	Pelaajan keho sekä keskusmuisti

vuonna 2001. Ensimmäinen Java-pelien latauspalvelu aloitti Yhdysvalloissa Nextelin operoimana vuonna 2002. Samana vuonna tuli laajasti tarjolle niin Yhdysvalloissa kuin Euroopassakin useita pelien latauspalveluja, joista pelin sai ladattua maksettuaan sen. Myös muut kuin operaattorien omat pelipor- taalit pääsivät kuukausiperustaisen laskutuksen pariin vuonna 2002. Vuonna 2003 oli vuorossa ensimmäinen suuren luokan ladattava 3D-peli 12 dollarin hintaan. [Tercek, 2007a, Tercek, 2007b].

Ensimmäinen reaaliaikainen pelaaja vastaan pelaaja -tyyppinen moninpeli julkaistiin Yhdysvalloissa samana vuonna. Pelin nimi oli *Bejewelled* ja sen julkaisija oli Jamdat Mobile. Vuonna 2003 moninpeleihin tuli myös useita lisäominaisuuksia tai uudistuksia, kuten pelin sisäinen mikromaksu ja ensimmäinen usean operaattorin verkossa toiminut moninpeli (*IHRA Racing*, jonka kehitti Mforma). Samana vuonna kokeiltiin myös kameran käyttöä NECin *Photo Battlerissä* ja Panasonicin *Shakariki Pettossa*. Toinen paikannusta käyttävä peli oli Panasonicin *Fishing Anywhere*, joka käytti GPS-paikannusta. Nokian N-Gage pelipuhelin julkistettiin myöskin 2003. [Tercek, 2007a, Tercek, 2007b].

Minkälaisia mobiilipelejä kuluttajat sitten haluavat pelata? Park Associatesin raportin ”The New Frontier: Portable and Mobile Gaming” [Parks Associates, 2007] mukaan mobiilipelien pelaajat ovat kiinnostuneita pulma- ja

korttipeleistä (55 %) sekä arcade- (33 %) ja sana- (32 %) ja lautapeleistä (23 %). Edellä mainitut Park Associates luokittelee myös ajanvietepeleiksi. Ajanvietepelit on kuitenkin tätä monimutkaisempi ja monimerkityksesempi termi, jota Jussi Kuittinen, Annakaisa Kultima, Johannes Niemelä ja Janne Paavilainen ovat tarkastelleet artikkelissa "Casual games discussion" [Kuittinen *et al.*, 2007]. Park Associates:n raportti luokittelee core-peleiksi urheilu- (6 %), lento- ja ajo- (7 %), action- ja seikkailu- (6 %) sekä FPS-pelit (6 %). Raportissa oli tehty kysely 407 ihmiselle. Kysely oli kuitenkin toteutettu Internet-kyselynä, mistä johtuen tulosta voi kyseenalaistaa siltä osin, että ovatko kyselyyn vastanneet todella kiinnostuneita pelaamaan matkapuhelimella. Tulos heijastelee todennäköisesti myös melko pitkälle verkon ajanvietepelaajien mieltymyksiä. Tästä huolimatta kysely näyttää kiinnostuksen suunnan mobiilipeleihin, sillä Internetin ajanvietepelejä pelaavien on näytetty siirtyvän suosiollisesti myös mobiilipelaajiksi [Olsder, 2007]. Vihisen tutkimuksessa [2007] vastanneiden tärkeimmiksi arvoiksi mobiilipelejä kohtaan nousivat mobiilius, pelin laatu ja hinta. Tutkimuksessa nousi esille yhtenä arvona myös moninpelaamisen mahdollistaminen. Seuraavaksi tarkastelen juuri moninpelaamista.

2.3 Verkkomoninpelit

Pelit ovat aikojen alusta olleet sosiaalisia, koska niihin on osallistunut yleensä useampia pelaajia. Moninpelin voi yksinkertaisesti määritellä olevan peli, johon osallistuu useampia ihmispelaajia. Moninpelien sosiaalinen aspekti kiehtoo pelaajia. Tietokoneen tekoälyvastustajia vastaan pelaaminen, vaikka vastus voisi nykyisin vastatakin osittain ihmistä, ei ole moninpelaamista siinä merkityksessä kuin se tässä tutkielmassa halutaan määrittää. Rajausta on merkittävä siksi, että tekoälyvastustajan tai -vastustajien ei tarvitse sijaita verkkoyhteyden päässä olevilla solmuilla. Tekoäly voi sijaita pelaajan omalla laitteella ja näin ollen verkkoyhteys ei ole tarpeellinen.

Asia voi tuki muuttua tulevaisuudessa. Esimerkiksi matkapuhelimien prosessointiteho tietokoneisiin verrattuna on sen verran vähäinen, että raskasta tekoälyä sisältävissä peleissä tekoäly voisi sijaita palvelimella. Pääasiassa pelaajat ovat kuitenkin kiinnostuneita sosiaalisesta kontaktista. Yksinpelit on voinut nähdä historiassa oikeastaan hieman poikkeavina. Tämä poikkeavuus kuitenkin vahvistui tietokoneiden myötä. Varhaiset videopelilaitteet tosin sisälsivät yleensä kaksi ohjainta mutta tietokoneiden myötä yksinpelit ottivat valta-aseman. Tietokonepelejä voitiin tuki pelata siten, että saman laitteen ääressä

oli useampia pelaajia. Pelaajat saattoivat vuorotella tai he pelasivat samalla tai jaetulla näytöllä.

Pelaajakokoonpanoa käsittelevät Christian Elverdam ja Espen Aarseth artikkelissa "Game Classification and Game Design" [2007]. He esittelevät avoimen pelitypologian, jolla voidaan tunnistaa oleelliset erot pelien välillä ja luokitella pelit tarkasti sekä analyytisesti. Typologian pelaajakokoonpanoa käsittelevä osuus jakaa kokoonpanon kuuteen eri luokkaan: yksi pelaaja, yksi joukkue, kaksi pelaajaa, kaksi joukkuetta, monta pelaajaa ja monta joukkuetta. Huomattavaa tässä on se, että Elverdam ja Aarseth eivät vain luokittele pelaajien lukumäärää, vaan he ovat myös erottaneet pelaajien muodostamat joukkueet. Tämä antaa pelien analyysiin lisää tarkkuutta.

Toinen monienpelien kannalta tärkeä seikka on pelaajien toiminnan ajallisuus: Milloin pelaajat pelaavat peliä ja miten? Tätä jakoa on jäsennetty käsitteillä samanaikainen, vuoropohjainen ja reaaliaikainen. Lisäksi asiaan liittyy synkronisuus ja asynkronisuus. Zagalin *et. al.* [2000] mukaan samanaikaisessa (concurrent) pelissä kaikki pelaajat osallistuvat yhtäaikaaisesti. Heidän mukaansa kaikki pelaajat tietävät, että kaikki pelaajat ovat yhtä aikaa läsnä. Tämä ei kuitenkaan tarkoita, että kaikki pelaajat toimisivat samanaikaisesti. Zagalin *et. al.* jatkavat, että synkronisessa pelissä pelaajat ovat läsnä yhtä aikaa mutta he suorittavat toiminnot vuorollaan eli vuoropohjaisesti. Zagalin *et. al.* mielestä on olemassa vielä yksi vaihtoehto: asynkroninen peli. Siinä pelaajat voivat liittyä peliin ja lähteä pelistä miten haluavat eikä kaikkien pelaajien tarvitse olla läsnä pelissä yhtä aikaa. Voi olla myös tilanteita, jossa pelissä ei ole yhtään pelaajaa jonain tiettyinä hetkenä. Zagalin *et. al.* esittelevät artikkelissaan tenniksen olevan synkroninen peli. Pelaajat lyövät palloa vuorotellen. Jos peliä tarkastellaan vain lyömisen kannalta, niin näin asia yleensä onkin. Pallon ja pelaajien liike on kuitenkin erittäin reaaliaikaista ja yhdenaikaista. Palloton pelaaja vaikuttaa myös liikkumisellaan pallolliseen pelaajaan, vaikka Zagalin *et. al.* mukaan ei ole hänen vuoronsa. Tällaisen varomattomattoman esimerkin käyttö voi tehdä tilanteen lukijalle epäselvemmäksi.

Viimeisenä esitelty vaihtoehto, asynkroninen peli, herättää mielenkiinnon tilanteesta, jolloin pelissä ei ole yhtään pelaajaa. Mikä on pelin tila silloin ja millainen on seuraavalle liittyvälle pelaajalle välitettävä tila? Jos pelimaailma on jatkuva (persistence), seuraava liittyvä pelaaja jatkaa samasta tilasta. Kaikki hänen muutoksensa tiloihin jäävät voimaan ja häntä seuraava pelaaja jatkaa pelaamista tilasta, johon edellinen pelaaja on vaikuttanut. Ei-jatkuva tila on sellainen, jossa pelimaailma tavallaan luodaan uudestaan ja muiden pelaajien

tilaan tekemät muutokset eivät välity seuraavaan pelikertaan. Samanaikainen peli on yleensä ei-jatkuva. Asynkroniset pelit ovat yleensä jatkuvia.

Asynkronisten pelien tilan jatkuvuuden mainitsee myös Ian Bogost artikkelissa [2004], jossa hän pohtii ajanvietteellisten moninpelikokemusten tulevaisuutta. Bogost luettelee asynkronisen moninpelin tyypillisten piirteiden olevan seuraavat:

1. Asynkroninen pelaaminen mahdollistaa monen pelaajan pelaavan peräkkäin.
2. Asynkroninen pelaaminen vaatii jonkinlaista jatkuvaa tilaa, johon kaikki pelaajat vaikuttavat ja joka vaikuttaa takaisin kaikkiin pelaajiin.
3. Asynkronista pelaamista määrittävät tauot pelaajien välissä.
4. Asynkronisen pelaamisen ei tarvitse olla tärkein peliä määrittävä tekijä.

Bogost pohtii, että ajanvietteellisten moninpelien yksi tulevaisuuden vaihtoehto on asynkroninen pelaaminen, koska siinä ei tarvita jatkuvaa ja usean pelaajan yhtäaikaista osallistumista. Täten se voisi sopia mobiilipelienkin moninpelimuodoksi. Sellaisessa pelissä pelaaja osallistuu peliin ehkä lyhyitä aikoja kerrallaan mutta useasti ja erilaisista käyttökonteksteista.

Edellä käsittelin moninpelin määrittelemistä ja eri aspekteja. Miten sitten verkkopeli määritellään? Tässä tutkielmassa käytän verkkomoninpelistä seuraavaa määritelmää: Jos peliä pelataan useilla verkon yli kommunikoivalla laitteella usean pelaajan toimesta, on kyseessä verkkomoninpeli. Tämä määrittely ei tarkoita, että pelaajien tarvitsisi pelata peliä yhtä aikaa eli verkkomoninpeli voi olla myös asynkroninen. Termillä verkkomoninpeli haluan painottaa moninpelin tapahtuvan tietoverkkojen välityksellä. Koska tutkielma käsittelee mobiilipelejä, tietoverkot ovat tällöin lähinnä matkapuhelinverkkoja. Verkkomoninpelissä pelaajien laitteet vaihtavat tietoa verkkojen välityksellä. Verkot ovatkin se kanava, jota hyödyntämällä verkkopeliä voi pelata huomattavasti suurempi pelaajajoukko kuin mitä olisi mahdollista toteuttaa fyysisessä tilassa; pelaajia voi olla tuhansia, jopa kymmeniätuhansia [Siitonen, 2007].

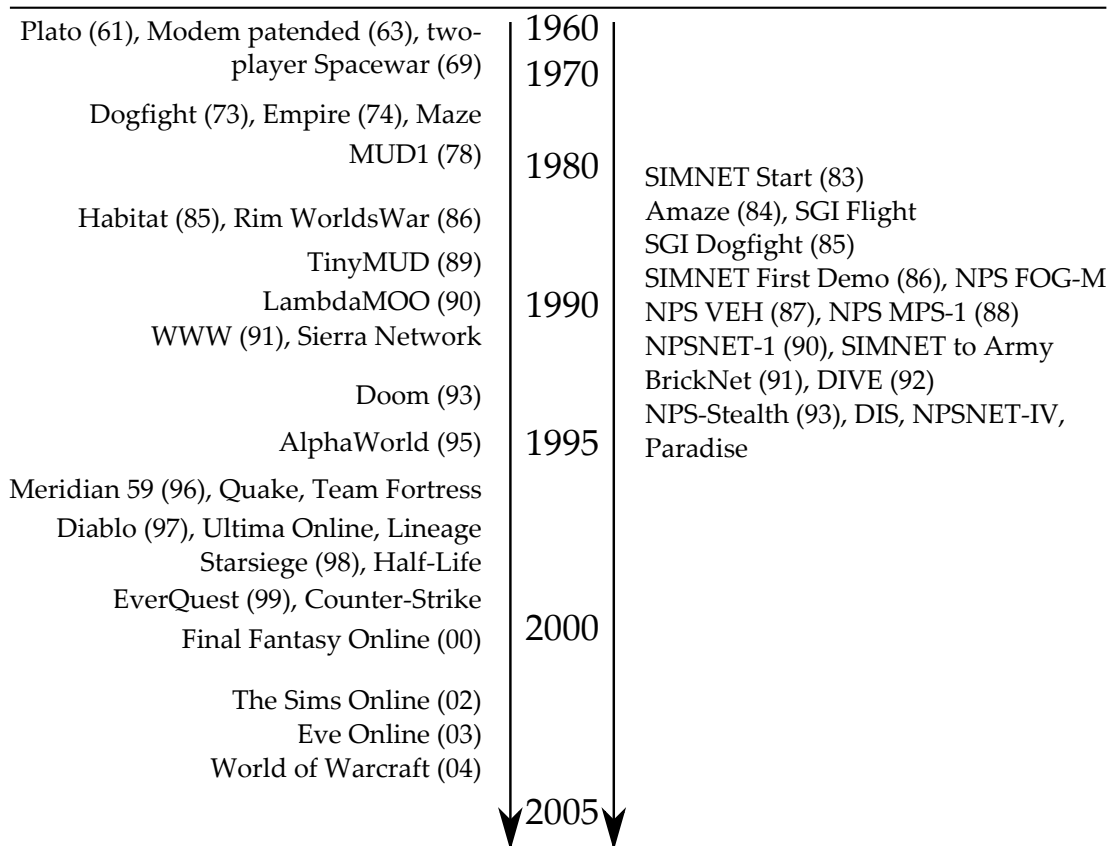
Ensimmäinen suuren suosion saanut useampaa tietokonetta verkon avulla käytävä peli oli vuonna 1993 ilmestynyt *Doom*. Sitä seurasi saman kehittäjän, ID Softwaren, *Quake*-pelisarja. *Quake* oli ensimmäinen peli, joka tarjosi täyden kuuden asteen liikkumisvapauden. Siinä oli siis kolmiulotteinen ympäristö,

missä jokaisen akselin liikkuminen ja pyöriminen olivat vapaita painovoimaj rajoitusten puitteissa. *Quaken* death match -moodi, jossa useampi pelaaja pysyi pelaamaan toisiaan vastaan, oli erittäin suosittu. *Doom* ja *Quake* olivat ensimmäisiä suurta kaupallista suosiota saavuttaneita verkkomoninpelejä.

Friedl [2002] luonnehtii *Doomin* merkittävästi muokanneen käsitystä tietokonepeleistä sekä olleen teknisesti edistyksellinen. Friedel jatkaa *Doomin* mahdollistaneen enimmillään neljän pelaajan moninpelin. Ensimmäiset verkkomoninpelit olivat kuitenkin paljon aikaisemmin ilmestyneet Plato-järjestelmän päällä toimiva *two-player Spacewar* (1969) ja *Multi User Dungeon* eli *MUD* (1978). Plato oli monen käyttäjän järjestelmä ja sitä kehitettiin säännöllisesti. Kehitys alkoi vuonna 1961 ja vuonna 1972 se tuki 1000 yhtäaikaista käyttäjää. Kahden pelaajan *Spacewar* oli alun perin Platon väärinkäyttöä. [Friedl, 2002].

Verkkomoninpelien historiasta on esitetty yhteenveto kuvassa 2.2. Kuvan vasemmalla puolella ovat merkittävät verkkomoninpelit ja oikealla puolella on esitetty yhteenveto virtuaaliympäristöistä. Kuva on koostettu useista eri lähteistä [Mäyrä, 2008, Friedl, 2002, Koster, 2002, Zyda, 2005d, Spohn, 2005]. Listaus ei ole kaikenkattava, mutta antaa yleiskuvauksen eri tuotteista aina verkkomoninpelaamisen alkua ajoilta 1960-luvulta Internetin massiivisen monenpelaajan verkkoroolipeleihin (MMORPG). Mäyrän kirjasta löytyy lisäksi kattavampi katsaus verkkopelien historiaan.

Kuvan oikean laidan verkotetut virtuaaliympäristöt (networked virtual environment) esitellään sen vuoksi, että alan tutkimuksen kannalta ne ovat olleet merkittävässä asemassa. Vasemmalla puolella on niin armeijan tarpeista alkunsa saaneita (*SIMNET*), akatemian puolella kehitettyjä (mm. *NPSNET*, *Paradise* ja *DIVE*) kuin demoja, kuten (*Amaze* ja *SGI Flight*). [Zyda, 2005d]. Tristan Henderson [2003] sanoo verkkopelien olevan verkotettujen virtuaaliympäristöjen osajoukko. Verkkopelit ovat hänen mukaansa suosituin muoto verkotetuista virtuaaliympäristöistä. Hendersonin mukaan varsinkin armeijan tarpeita varten kehitetyistä verkotetuista virtuaaliympäristöistä on käytetty myös nimistystä DIS (distributed interactive simulation). Marshal *et. al.* [2004] esittelevät puolestaan termin DIA eli hajautettu interaktiivinen sovellus. Hendersonin mielestä pelit voivat olla graafisesti näyttävämpiä kuin virtuaaliympäristöt, mutta varsinkin sotilassimulaatiot ovat luotettavampia. Hän jatkaa, että kaista ei ole peleille ongelma mutta viive on. Näitä asioita tarkastelen tarkemmin kohdassa 3.



Kuva 2.2: Verkkopelien ja verkotettujen virtuaaliympäristöjen aikajana

2.4 Mobiililaitteet

Mobiililaitteita on nykyisin monenlaisia lukuisiin eri tarpeisiin. Yleispätevää mobiililaitteen määritelmää on hankalaa antaa mobiililaitteiden valtavan kirjjon vuoksi. Mobiililaitteiksi voidaan sanoa niin kannettavia tietokoneita jopa 20-tuumaisine näyttöineen kuin alle tulitikkuaskin kokoista MP3-soitinta. Mobiililaitteen määritelmää on lähestytty langattomuuden suunnasta, laitteen kookoa ja mukana kuljettavuutta tarkastelemalla sekä käyttäjän kannalta. Langattomuuden käsite puolestaan sisältää sen, että laite kommunikoi digitaalisesti ilman johtoja. Tämä on kuitenkin ongelmallinen määritelmä, koska jopa paikoilleen tarkoitettut pöytätietokoneet tai esimerkiksi televisioon liitettävät mediatoistimet voivat käyttää hyväkseen langattomia verkkoja. Tällaisessa tapauksessa ei kuitenkaan ole kyse mobiililaitteista ja siksi koon määrittelyllä on koitettu rajata laitteet käsiin sopiviksi (*handheld*).

Pieni käteen sopiva muoto ei myöskään pelkästään riitä määrittelemään

mobiililaitteita. Se ei riitä edes langattomuuden kanssa yhdessä käytettynä. Esimerkinä voisi antaa television kaukosäätimen, joka kommunikoi langattomasti ja on käteen mahtuva. Silti se ei ole tarkoittamani mobiililaite. Tommi Mikkonen [2004] kiteyttää määrittelyn vaikeuden seuraavasti: "Vaikka mobiililaitteen käsitteeseen liittyy paljon ominaisuuksia, jotka ovat sisäänrakennettuja oletuksia, ei ole olemassa yksiselittäistä ominaisuus- tai piirrejoukkoa, joka oikeuttaisi kutsumaan jotakin tiettyä yksittäistä laitetta tai järjestelmää mobiiliksi ja joka erottaisi sitä jonkin verran muistuttavat laitteet tai järjestelmät ei-mobiileiksi.". Matkapuhelin on hyvä esimerkki mobiililaitteesta mutta mobiililaitteeksi luetaan myös musiikki- ja videosoittimet, käsipelikonsolit, elektroniset kirjat ja PDA-laitteet.

Perimmiltään mobiliteetti liittyy käyttäjään eikä laitteeseen tai sovellukseen [Ballard, 2007]. Käyttäjä on se, joka on liikkeellä. Mikkosen pragmaattinen lähestymistapa johtaa samankaltaiseen tulkintaan, kun hän toteaa, että mobiililaitteella tarkoitetaan kaikkia tietojärjestelmän sisältäviä laitteita, joita niiden käyttäjä kuljettaa mukanaan. Eli käyttäjän on voitava kuljettaa laitetta mukanaan. Englannin kielen sana *mobile* tarkoittaaakin siirrettävää, liikuteltavaa tai vaihtelevaa. Se tarkoittaa myöskin matkapuhelinta tai kännykkää.

Mobiilipelilaitteiksi luetaan muun muassa käsikonsolit, kuten Nintendon Game Boy ja DS -sarjojen tuotteet sekä Sonyn Playstation Portable. Suurempien valmistajien ohella on myös hakkerihenkisempiä laitteita kuten GP2X. Aiemmin suosittuja olivat elektroniikkapelit, jotka sisälsivät vain yhden pelin. Tällaisia olivat esimerkiksi Nintendon Game & Watch -sarjan elektroniikkapelit. Kuten jo johdannossa totesin, nykyisin määrällisesti suurin mobiililaitteiden ryhmä on matkapuhelimet. Matkapuhelimet ovat laitteina niin ominaisuuksiltaan kuin käyttötarkoituksiltaan laaja ja monimuotoinen kenttä. Yhteinen nimittäjä niille kuitenkin löytyy: puhelinominaisuus. Matkapuhelin on ensisijaisesti puheyhteyksiin tarkoitettu matkaviestin. Tämä yhteydenpito-mahdollisuus ja sen luonnollisesti tarvitsema verkkoyhteys ovat matkapuhelimien tärkeimpiä ja muista laitteista erottavia tekijöitä [Turpeinen *et al.*, 2005].

Matkapuhelimet ovat siis ensisijaisesti kommunikointivälineitä. Käsikonsolit ovat ensisijaisesti pelilaitteita, vaikka niistäkin voi nykyisin löytyä tukea langattomille verkoille ja kommunikointiin. Matkapuhelinmalleja löytyy hintavista ja ominaisuuksiltaan monipuolisista älypuhelimista aina edullisiin peruspuhelimiin, jotka sisältävät minimissään vain puhelin- ja tekstiviestio-minaisuudet. Matkapuhelimen määritelmä kuuluu, että laitteella pitää voida kommunikoida puhumalla ja sen tulee olla langaton. Lataamiseen johtoa toki

vielä tarvitaan mutta itse käyttö tapahtuu pääsääntöisesti ilman johtoa.

Matkapuhelimia voidaan luokitella myös niiden käyttöjärjestelmän avulla. Valmistajilla on usein omia suljettuja käyttöjärjestelmiä, mutta myös josain määrin avoimia käyttöjärjestelmiä löytyy. Täysin avointa Linuxia on myös muokattu matkapuhelimiin sopivaksi. Sellainen löytyy esimerkiksi mobiililaitteeksi luokiteltavan Nokian Internet Tabletin sisältä. Malleja ovat 770, N800 ja uusin julkaisu N810. Näissä laitteissa ei kuitenkaan ole perinteisiä matkapuhelinominaisuuksia eikä tukea matkapuhelinverkoille. Niillä pystyy silti kommunikoidaan Skypeen kaltaisilla ohjelmilla WLAN-verkkoja käyttäen.

Pelien kannalta kiinnostavimpia matkapuhelinkäyttöjärjestelmiä ovat ne, jotka tarjoavat mahdollisuuden ajaa kolmansien osapuolten ohjelmia. Avoi- met ohjelmointirajapinnat ja kehitysympäristöt on nähty myös erottavana tekijänä konsolien ja matkapuhelimien suhteen [Turpeinen *et al.*, 2005]. Yleisin ohjelmointikieli, jota matkapuhelimien käyttöjärjestelmät tukevat, on Java ja sen Micro Edition (Java ME). Javan ”käännä kerran, aja kaikkialla” -periaate toteutetaan virtuaalikoneen avulla.

Matkapuhelimita virtuaalikone on kutistettu pienemmäksi Kilo Virtual Machineksi (KVM). Kun KVM käännetään toimimaan eri valmistajien matkapuhelinkäyttöjärjestelmillä, ohjelmoijan ei tarvitse enää huolehtia ohjelman- sa yhteensopivuudesta. Asia on näin vain periaatteessa - käytännössä melkein jokaiselle matkapuhelinmallille joudutaan tekemään oma versionsa SKU (stock keeping unit) pelistä. Näin menetellään sen takia, että matkapuhelimita on esimerkiksi eri kokoisia näyttöjä, eri tehoisia prosessoreita ja näppäimistö- töt eroavat toisistaan. Peleissä grafiikka on tärkeä elementti, ja se tulee skaalata jokaisen matkapuhelimen näytölle oikeaan kokoon. Pelien automaattisesta kääntämisestä jokaiselle puhelimelle sopivaksi versioksi riittäisi tutkimista omaksi tutkielmakseen. Tämän tutkielman tarkoituksiin riittää ymmärtää matkapuhelimien olevan erittäin heterogeeninen laitealusta ohjelmointikehityksen näkökulmasta.

Symbian OS on yleisesti älypuhelimissa toimiva käyttöjärjestelmä ja Series 60-alusta on sen laajimmalle levinnyt versio [Rapeli, 2008]. Symbianille kehitys tehdään C++:lla, vaikka myös sille löytyy MIDP eli Java ME -tuki. Toi- nen yleinen matkapuhelimien käyttöjärjestelmä, varsinkin Yhdysvalloissa, on BREW. Sillekin natiiviohjelmistokehitys tehdään C++:lla ja siitäkin löytyy Java ME -tuki. Muita mobiilipelin kehityksen kannalta merkittäviä alustoja ovat FlashLite, Mophun ja Python for S60. Mikään näistä ei ole oma käyttöjärjestelmänsä, vaan ne ovat ohjelmointikieliä ja kehitysympäristöjä. Mikkonen [2004]

ei puhu käyttöjärjestelmistä vaan käyttää termiä mobiilialusta tarkoittamaan tätä kirjoa, jonka varaan mobiililaitteiden ohjelmistot toteutetaan. Mikkonen listaa mobiilialustoiksi edellä mainittujen lisäksi vielä Palm OS:n ja Microsoft Smartphonon. Hän myös pitää Symbianin OS:ää ja Javaa keskeisimpänä mobiilialustoina.

Mielenkiintoisia tulevaisuuden mobiilialustoja voivat Open Handset Alliancen Android ja Applen iPhone. iPhone on jo myynnissä ja sen kehitysympäristökin on saatavilla. iPhonessa mielenkiintoista on suuri kosketusnäyttö. Pitkälti Googlen alulle panema Android on käyttöjärjestelmän, väliohjelmiston (middleware) ja tärkeimmät ohjelmat sisältävä avoin ja ilmainen mobiilialusta [Google, 2008, Open Handset Alliance, 2007]. Sitä tukevia matkapuhelimia ei ole vielä markkinoilla.

Tutkielmassani tarkoitan mobiililaitteella ensisijaisesti matkapuhelinta eli kännykkää. Tutkielmani kannalta matkapuhelimen tekee relevantiksi niiden suuri määrä ja levinneisyys. Määritelmänsä mukaan niissä on myös lähes kaikkialla käytettävissä oleva verkkoyhteys matkapuhelinverkon kautta. Matkapuhelinverkon kautta matkapuhelimiin voidaan myös ladata ohjelmia ja täten myöskin pelejä. Matkapuhelinten, niiden sisältöjen ja käyttäjien välillä on myös erityislaatuinen positiivinen suhde [Vincent, 2005]. Koska matkapuhelimet ovat kommunikoinnin välineitä, soveltuvat ne hyvin moninpelien alustaksi. Kuten kohdassa 2.3 mainitsin, ovat moninpelit ensisijaisesti sosiaalista pelaamista.

Matkapuhelimen käyttökonteksteista puhuttaessa tarkoitetaan käyttötilanteen aikaa ja paikkaa eli tilannetta, jossa käyttäjä kulloinkin käyttöhetkellä on [Järvinen, 2002]. Mobiliteetin yhteydessä on hoettu mantraa ”missä vain, milloin vain”. Totta onkin, että matkapuhelimet ovat lähes aina mukana olevina laitteina käytettävissä silloin, kun käyttäjä sitä haluaa. Arkielämässä mobiilitilanteet voivat olla silti melko suhteellisia. Eija-Liisa Kasesniemi ja Pirjo Rautiainen käsittelevät näitä mobiliteetin liikkuvuuden tasoja kirjassaan *Kännyssä piilevät sanomat - Nuoret, väline ja viesti* [2001]. Heidän mielestään liikkuvuus on jaettavissa osittaiseen liikkuvuuteen ja täydelliseen liikkuvuuteen. Täydellinen liikkuvuus toteutuu Kasesniemen ja Rautiaisen mukaan esimerkiksi pyörällä ajettaessa, jolloin matkapuhelimen käyttäjä on keskellä fyysistä liikettä eikä voi keskittää koko huomiokykyään matkapuhelimeen. Osittaisella liikkuvuudella he tarkoittavat tilanteita esimerkiksi junissa tai jopa täysin ympäristöltään liikkumattomissa paikoissa kuten toimistossa. Osittain tämä jako on linjassa kohdassa 2.2 tarkasteltuun Suomisen esittämään karttaan ja

sen mobiilis-motiivilliseen akseliin.

Kasesniemen ja Rautiaisen mukaan niin aikuiset kuin nuoretkin korostavat sitä, että matkapuhelin kulkee aina mukana. Markkinoinnin kuva matkalla tai liikkeessä käytetystä matkapuhelimesta on siis mennyt hyvin perille. Kasesniemi ja Rautiainen yllättävät kuitenkin tutkimustuloksellaan, jossa matkapuhelimen määrällisesti suurin käyttöosuus ei olekaan liikkeessä vaan paikallaan. He jatkavat, että ”täyden liikkuvuuden korostus osoittautuu osin harhaksi”. Kasesniemen ja Rautiaisen mukaan nuoret käyttävät matkapuhelinta määrällisesti paljon kotonaan: omassa huoneessa, vuoteessa. Usein vielä iltamyöhään tai jopa yöllä. Kasesniemen ja Rautiaisen mukaan kyse on pysähtyneestä käytöstä.

Tämä ei toisaalta ole yllättävää, sillä suurin osa ajasta ollaan kotona ja jäljelle jäävästä ajasta valtaosa ollaan töissä tai koulussa. Näiden paikkojen välillä siirtymiseen käytetty aika on huomattavasti pienempi kuin paikoissa vietetty aika. Näin ollen todellinen mobiilius matkapuhelimen käytössä on suhteellista. Järvinen vertaa, että konsoli ja tietokonepelaaminen sen sijaan tapahtuu lähes poikkeuksetta kotona, joko pelaajan kodissa tai ystävän luona. Verrattaessa konsoli- ja tietokonepeleihin matkapuhelimessa toimivilla mobiilipeleillä on mahdollista rikkoa kodin seinät pelaamisen vankilana ja murtautua mobiilipeleiden kanssa minne vain, milloin vain. Toinen vaihtoehto on, että lähdettäessä kotoa muutenkin jonkin asian takia matkapuhelin, aina mukana olevana laitteena, lähtee luonnollisesti mukaan.

Nokian tutkimuksen [Nokia, 2006] mukaan matkapuhelimilla pelataan kyllä liikkeellä ollessakin (61 % vastaajista) ja odotellessa (56 %). Nokian tutkimus vahvistaa edellä esitettyjä tuloksia mobiiliuden suhteellisuudesta, sillä sen mukaan kotona pelataan kuitenkin eniten (62 %). Nokian tutkimus kuitenkin osoittaa, että puhelin laitteena on läsnä niin kotona kuin kodin ulkopuolellakin. Kontekstitietoisuutta on tutkittu aikaisemminkin ja esimerkiksi Chenin ja Kotzin raportti nimeltä ”A Survey of Context-Aware Mobile Computing Research” [2000] antaa lisätietoa asiasta.

Kuten todettua, matkapuhelin on kaikkialla mukana kulkeva laite. Jotta matkapuhelin voisi muodostaa yhteyksiä, tulee sillä olla pääsy verkkoihin. Seuraavassa luvussa tarkastelen tarkemmin yhteyden mahdollistavia langattomia verkkoja.

3 LANGATTOMAT VERKOT

Verkot ovat välttämättömiä mobiileille verkkomoninpeleille. Verkot edustavat fyysistä tasoa, jolla solmut ovat yhteydessä toisiinsa. Tämän fyysisen verkkokerroksen päälle rakennetaan looginen kerros, joka määrittää viestien virtaa verkossa. Looginen taso määrittää myös käytettävät arkkitehtuurit kommunikoinnille, datalle ja hallinnoinnille. Vasta loogisen tason varaan kehitetään itse verkkoa käyttävä sovellus. Tässä tutkielmassa tarkastelun kohteena sovellusalueen osalta ovat monen pelaajan mobiilit verkkomoninpelit. Seuraavaksi tarkastelen lyhyesti eri langattomia verkkotekniikoita ja keskityn fyysisen kerroksen osalta matkaviestinverkkoihin. Tämä on tarpeen, sillä verkkojen fyysiseltä tasolta kumpuaa rajoituksia. Rajoitukset liittyvät erilaisiin resurssirajoitteisiin kuten fyysisiin, teknisiin ja laskennallisiin rajoitteisiin. Nämä asettavat rajoja muun muassa sille, mitä ylipäätään on mahdollista toteuttaa fyysisen kerroksen päälle muille tasoille. Resurssirajoituksia matkaviestinverkkojen kannalta tarkastelen tarkemmin kohdassa 3.3.

3.1 Verkkoteknologiat

Matkapuhelimet ovat luonnollisesti yhteydessä matkapuhelinverkkoihin. Matkapuhelinverkot ovat laajan alueen radioaaltoliikenteeseen pohjautuvia verkkoja ja perustuvat riittävän tiheään sekä kattavaan tukiasemaverkostoon. Tukiasemaverkoston avulla saavutetaan palvelulle riittävä peitto. Lyhyemmän kantaman langaton verkkotekniikka on esimerkiksi WLAN. Se ei ole tiukasti matkapuhelmiin sidottu verkko. WLAN on enemmänkin yleistynyt kannettavien tietokoneiden myötä, mutta nykyisin tuki WLAN-verkoille on myös osassa älypuhelimia. Joistakin matkapuhelimista löytyy myös lyhyemmän kantaman Bluetooth-tuki. Sen kantama on noin kymmenen metriä, joten pelien kannalta pelaajien tulee olla melko lailla samassa tilassa, jotta Bluetooth-yhteyttä voitaisiin käyttää.

Puhelinverkot, sekä kiinteät langalliset että langattomat, on kehitetty alunperin puheliikennettä varten. Vasta myöhemmin on puhelinverkoille huomattu muutakin hyötykäyttöä mahdollisuutena välittää kaikenlaista dataa. Datavälittämisestä on tullut nykyään suurempi ja tärkeämpi osa tietoliikennettä kuin pelkän puheen välittämisestä. Ahmad [2005] kertoo kirjassaan *Wireless and Mobile Data Networks* kolme tärkeää ominaisuutta äänisignaaleille: kiinteä vähimmäiskaistanleveys, väljä laadun määrittely ja viive. Ahmadin mu-

kaan näistä ehdottomin on tiukka viiveettömyyden vaatimus. Ihmisten välisessä puheessa ei saa olla korkean interaktiivisen luonteensa takia yli 250 millisekunnin viiveitä. Tämä on myös ITU:n maksimisuositus viiveelle. Viiveen vaihtelun eli *huojunnan* (jitter) tulisi olla Ahmadin mukaan kertaluokkaa pienempää. Tällöin puheliikenteen suurin sallittu huojunta olisi 25 millisekuntia. Näiden ominaisuuksien perusteella puheliikenne on perinteisesti välitetty *piirikytkentäisessä* (*circuit switched*) verkossa. Näin on toimittu myös langattomien verkkojen kanssa. [Ahmad, 2005]. Piirikytkentäisessä verkossa kahden solmun välille muodostetaan koko ajan avoinna oleva yhteys. Pakettivälitteisessä (*packet switched*) verkossa solmujen välillä kulkeva liikenne lähetetään pienemmissä osissa, toisin sanoen paketteina. Solmujen välille ei muodosteta koko ajan avoinna olevaa yhteyttä vaan paketit kulkevat yhteydettömästi. Kaikki paketit eivät välttämättä kulje edes samaa reittiä solmujen välillä. Uusimman sukupolven matkapuhelinverkot ovat siirtyneet pakettivälitteiseen ratkaisuun sekä puheen että datan osalta.

Koska tutkielmani keskittyy mobiileihin moninpeleihin, en käsittele kiinteitä langallisia verkkoja. En käsittele myöskään Ahmadin [2005] kuvailemia kiinteitä langattomia verkkoja, koska ne eivät ole mobiileja. Langattomista verkoista puhuttaessa tarkoitetaan *matkaviestinverkkoja* (mobile network, cellular network), *langattomia lähiverkkoja* (wireless local area network, WLAN), *lähikantaman verkkoja* (short range network) sekä *käyttäjäverkkoja* (personal area network, PAN), *kaupunkiverkkoja* (wireless metropolitan area networks), *spontaaniverkkoja* (*ad hoc networks*) ja satelliittiyhteyksiä [Boukerche, 2006, Ahmad, 2005]. Spontaaniverkot toimivat ilman tukiasemaa. Spontaaniverkko voidaan muodostaa nimensä mukaan spontaanisti kahden tai useamman laitteen välille. Spontaaniverkossa ei ole myöskään reitittimiä ohjaamassa liikennettä, vaan jokainen laite ohjaa liikennettä osaltaan. Verkot voidaan luokitella niiden kantaman perusteella pienemmästä kantamasta suurempaan: PAN, LAN, MAN ja WAN. Seuraavaksi käyn lyhyesti läpi edellä luetellut verkkotyypit kantamien järjestyksessä.

Käyttäjäverkkojen kantama on muutamissa metreissä käyttäjän ympärillä. Maksimikantama on yleensä noin kymmenen metriä. Käyttäjäverkoissa käytettyjä tekniikoita ovat muun muassa infrapuna (IrDA) ja Bluetooth. [The RFID Centre, 2006]. Infrapuna on oikeastaan sarjakaapelia vastaava yhteys mutta kaapeli on korvattu infrapunalähettimellä ja -vastaanottimella. Infrapunaan perustuvat laitteet tarvitsevat yhteyttä varten suoran näköyhteyden ja kantama on korkeintaan noin metrin luokkaa. IrDa on luonteeltaan spontaani-

verkko ja sen kaistanleveys on 9600 bitistä sekunnissa (bps) jopa 16 megabittiin sekunnissa (Mbps) saakka. [SIG, 2008]. Se ei ole radioaaltoihin perustuva tekniikka vaan pohjaa nimensä mukaisesti valoon. Radioaaltoihin perustuva Bluetooth-tekniikka on jo pääasiassa syrjäyttänyt infrapunayhteyden.

Bluetooth-tuki löytyy jo yli miljardista laitteesta mutta yleistymisen hiipumisen merkkejä on jo havaittu [MarketResearch.com, 2008]. Bluetooth on standardoitu radiotekniikka ja kommunikointiprotokolla. Tekniikan nimenä käytetään myös IEEE 802.15.1 -standardia. Tavoitteena on ollut vähän virtaa kulluttavan ja avoimia radiotaajuuksia käyttävän tekniikan kehittäminen. Bluetooth käyttää avointa 2,4 GHz radiotaajuuksien aluetta [SIG, 2008]. Bluetoothin kantama määräytyy teholuokan mukaan. Alin teholuokka (Class 3) kantaa noin metrin päähän, yleisimmin matkapuhelimissa käytetty toinen luokka (Class 2) kantaa 10 metriä ja suurin teholuokka (Class 1) kantaa jopa 100 metrin päähän. [The RFID Centre, 2006]. Bluetoothin maksimi kaistanleveys käyttäen EDR-tekniikkaa on 3 Mbps. Bluetoothin ensimmäisen version kaistanleveys on 1 Mbps. [SIG, 2008].

Bluetoothin seuraajan nimenä oli pitkään Wibree mutta se on nyt nimetty ULP Bluetoothiksi. Tekniikan tavoitteena on entisestään radikaalisti pudottaa virran kulutusta ja täten mahdollistaa esimerkiksi pienen pariston virran riittäminen yhteyksiin jopa vuodeksi. [Wibree, 2007]. Bluetoothia käyttäen on toteutettu useita mobiileja moninpelejä. Myös Bluetooth toimii spontaaniverkon periaatteella. Bluetooth-laitteiden muodostamaa verkkoa kutsutaan *pikoverkoksi* (piconet) ja siihen voi olla liittyneenä korkeintaan kahdeksan laitetta kerrallaan [Mikkonen, 2004].

Koska käyttäjäverkkojen kantama on lyhyt, edellyttää se pelaajien olemista samassa tilassa - käytännössä samassa huoneessa. Tämä ei kuitenkaan ole mobiilipelien todellisen luonteen mukaista. Pelaajien olisi kokoonnuttava samaan tilaan samaan aikaan pelatakseen peliä. He eivät voi pelata peliä sieltä, missä sattuvat olemaan silloin, kun he haluavat verkon kantaman ja spontaaniverkon luonteen vuoksi. Ajanvietepelien ja mobiilipelien kohdalla samaan fyysiseen tilaan pääsy voi olla pelisessiolle suorastaan ylitsepääsemätön este. Toisaalta monet seikat, esimerkiksi pelaajien välinen kommunikointi, helpottuvat pelaajien ollessa samassa tilassa. Edellä esitettyjen huomioiden vuoksi en tässä tutkielmassa tarkastele käyttäjäverkkoja tämän tarkemmin.

Siirryttäessä käyttäjä- eli PAN-verkkojen kantamasta hieman laajemmalle tullaan lähi- eli LAN-verkkojen alueelle. Standardiperhe langattomille lähiverkoille (WLAN) on IEEE 802.11. Standardiperheen versiot merkitään kirjaimel-

la numerosarjan perään. Tällä hetkellä julkaistuna ovat standardit a:sta j:hin. Valmistelussa on stardardeja z:aan saakka. Tämän hetken suosituimmat standardit kuluttajamarkkinoilla ovat 802.11b ja 802.11g. Ensimmäisen kaistanleveys on 11 Mbps ja jälkimmäisen 54 Mbps. [Kerry, 2008]. WLAN:n kauppanimenä kuluttajille käytetään toisinaan termiä Wi-Fi¹. LAN- ja WAN-verkkojen kantaman väliin on kehitetty MAN-verkot. MAN-lyhenne muodostuu sanoista *metropolitan area network* ja tarkoittaa siis kaupunkiverkkoja. Langattoman kaupunkiverkon kauppanimenä on WiMAX. WiMAX pohjautuu IEEE 802.16-standardiin. WiMAX-tekniikan on ajateltu tulevan myös osaksi suunnitteilla olevaa neljännen sukupolven matkapuhelinverkkoja. WiMAX-tekniikan teoreettinen kaistanleveys on 75 Mbps mutta mitattu kapasiteetti jää 2 Mbps:iin latauksessa ja 100 kbps - 1 Mbps lähetyksessä [Mäkeläinen, 2006]. WiMAX:in teoreettista maksimikaistaa lähentelevät nopeudet voidaan saavuttaa erittäin pienellä etäisyydellä [Wikipedia, 2008b].

Vielä MAN-verkkoja suurempi kantama saavutetaan laajan alueen WAN-verkoilla. Matkaviestinverkot luetaan kuuluviksi laajan alueen verkkoihin. Matkaviestinverkoissa on jo käytössä niin sanottu kolmas sukupolvi, 3G. Matkaviestinverkkojen historia lähtee kuitenkin liikkeelle jo jälkeempään 0G:ksi nimitetystä analogisesta ARP-verkosta. 1980-luvulla käyttöön otettu analoginen NMT-matkapuhelinverkko (1G) suljettiin Suomessa vuoden 2002 lopussa. Ruotsissa NMT-verkot suljettiin vuoden 2007 lopussa [Laurio, 2007]. Yhdysvalloissa vastaava ensimmäisen sukupolven AMPS-matkapuhelinverkon alasajo sallittiin helmikuussa 2008 ja operaattorit alkoivat sammutella verkkojaan [Linnake, 2008]. Nykyään vallitseva matkapuhelinverkkotekniikka on toisen sukupolven (2G) GSM. GSM on puheen osalta piirikytkentäinen. GPRS on pakettikytkentäinen mobiilidatapalvelu GSM-verkon päälle. GSM:a GPRS:n kanssa kutsutaan toisinaan 2,5G:ksi. [GSM Association, 2007]. Tänä vuonna GSM-verkkoa käyttäviä matkapuhelimia on kaikista laitteista noin 85 prosenttia, mikä tarkoittaa, että käyttäjiä on jo yli 2,6 miljardia. [MarketResearch.com, 2008]. Tähän lukuun voidaan lisäksi laskea, että lähes kaikki uudempaa kolmannen sukupolven matkaviestinverkkoja tukevat laitteet tukevat vielä myös GSM:a. GSM:n tiedonsiirtoon löytyy useampia päivityksiä, kuten EDGE ja HSCSD. Jälkimmäisen ollessa nimensä mukaisesti nopeuspäivitys GSM:n alkueraiseen piirikytkentäiseen datan välitykseen.

¹ Yleisesti luullaan Wi-Fi:n muodostuvan sanoista "wireless fidelity". Tämä on kuitenkin virheellinen käsitys termin ollessa vain mitään tarkoittamaton sana. [Fleishman, 2005].

Kolmannen sukupolven (3G) UMTS-verkot ovat alkamassa valloittaa markkinoita 2G:lta. Vielä ei kuitenkaan ole näkyvissä päivämäärää 2G-verkkojen sulkemiselle. 3G:in tärkein uudistus 2G:iin nähden on sen kyky tukea useampia käyttäjiä. Tämä on tarpeen etenkin kaupungeissa. Nopeudetkin toki kasvavat 3G:in myötä. UMTS:ssä käytettyjä radiostandardeja ovat muun muassa WCDMA ja CDMA2000. UMTS:iin on useita päivityksiä, joilla nopeutta nostetaan. Nimeltään ne ovat muun muassa HSPA ja UMTS-TDD.

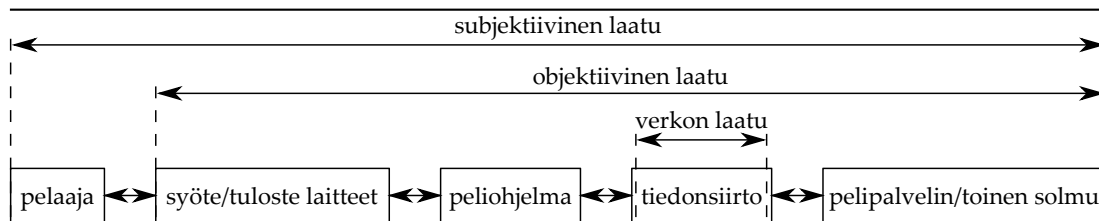
3G:in seuraajaa 4G:a suunnitellaan jo. 4G:sta ei olla vielä päästy sellaiseen yksimielisyyteen, että sen määritelmä tiedettäisiin. 4G:in suunnittelun lähtökohtana on pidetty sitä, että se on täysin IP-pohjainen järjestelmä. Tavoitteena on luonnollisesti myös nopeuden kasvattaminen. Todennäköisesti 4G on yhdistelmä, virtuaaliverkko, erilaisia nykyisistä kehitettyjä ja uusia verkkotekniikoita. [Wikipedia, 2008a].

Boukerche [2006] odottaa langattomien verkkojen tulevaisuuden olevan kirjava. Hänen mukaansa tulevaisuudessa tulee todennäköisesti olemaan yhtä aikaa useita osittain päällekkäisiä teknologioita, jotka tarjoavat langattoman nopean Internet-yhteyden. Hän jatkaa myös, että jokaiselle teknologialle löytyy oma pääsegmentti mainiten esimerkiksi langattoman lähiverkon, joka todennäköisesti tulee olemaan standardiverkko aktiivialueilla kannettavissa tietokoneissa ja kämmenmikroissa (PDA). Aktiivialueita kutsutaan myös liittytäpisteiksi (hotspot). Näillä alueilla sijaitsee WLAN-tukiasema, jonka kautta verkkoyhteys voidaan muodostaa. Matkapuhelinten osalta Boukerche kertoo useiden teknologioiden integraation samaan laitteeseen olevan myös todennäköistä. Boukerchen visioima kehitys on jo toteutunut, sillä nykyisin on vaikeaa löytää uutta kannettavaa tietokonetta ilman langatonta lähiverkkoa ja työpuhelinmenikin sisältää tuen sekä 2G- (GSM 900 / 1800 / 1900) että 3G- (WCDMA 2100) matkaviestinverkoille, lähikantaman verkoille (Bluetooth ja IrDa) ja langattomalle lähiverkolle (WLAN 802.11b/g). Verkkojen heterogeenisyys on tarpeen, koska ei ole olemassa yksittäistä teknologiaa, joka palvelisi kaikkia tarpeita. Kaikille ratkaisuille on oma optimaalinen sijaintinsa ja tarkoituksensa. [Akkawi *et al.*, 2004].

3.2 Pelien erityisvaatimukset verkoille

Pelin laatua voidaan tarkastella usealla tasolla. Tärkein tekijä on pelaajan pelistä saama pelikokemus (game experience). Schaefer *et al.* [2002] luokittelevat pelin laatua seuraaviin osa-alueisiin: subjektiiviseen, objektiiviseen ja ver-

kon laatuun. Subjektiiivinen laatu on osa pelikokemusta, mutta pelikokemus on myös paljon laajempi kenttä kuin pelkkään pelaamisen hetkeen ja vuoro-vaikutukseen liittyvä tekijä [Kuittinen *et al.*, 2007]. Kuvassa 3.1 on Schaeferin *et al.* näkemys pelin laadun osa-alueista. Tässä luvussa keskityn verkon laatuun.



Kuva 3.1: Pelin laadun osa-alueet

Verkkoja voidaan luonnehtia niiden ominaisuuksien mukaan. Yleensä puhutaan vain verkon *kaistanleveydestä* tai *kaistasta* (*bandwidth*). Internet-liittymiä kaupataan nykyisin laajakaistoina, jolloin niiden tiedonsiirtonopeuden sanotaan olevan esimerkiksi 2 Mbit/s. Kaistaa pidetään yleensä ainoana tärkeänä ominaisuutena, vaikka se määrittää oikeastaan vain yhteyden teoreettisen nopeuden. Toinen tärkeä verkkoja määrittävä ominaisuus on *viive* (*latency, delay*). Peleissä juuri viive näyttelee yleensä tiedonsiirtokaistaa tärkeämpää roolia. Siirrettävät datamäärät ovat suhteellisen pieniä, joten kohtuullinen kaista riittää hyvin. Käyttäjän painalluksen ja täten toiminnon, kuten hahmon liikkeen tulisi näkyä kuitenkin välittömästi pelaajalle itselleen mutta myös muille pelaajille. Tiedon välittymisen pelaajalta A pelaajalle B kuuluu teoriassa vähintään yhtälön $\frac{\text{matka}}{\text{nopeus}} = \text{aika}$ kuvaama aika. Matka on seuraavassa esimerkissä pelaajien välinen fyysinen etäisyys ja nopeus on valon nopeus tyhjiössä. Näin ollen A:n ollessa Helsingissä ja B:n Rovaniemellä heidän viestiensä viive on vähintään $\frac{815000\text{m}}{299792458\frac{\text{m}}{\text{s}}} = 0,0027\text{s}$ eli 2,7 ms. Tiedon välittymisen nopeudelle on siis teoreettinen minimi. Teoreettinen minimi on kuitenkin kaukana käytännöstä. Tietoverkot ja niihin kytketyt laitteet aiheuttavat lisää viiveitä. Tyypillinen Ethernet-paikallisverkon viive on yleensä alle 0,3 ms ja vaihtelee enemmän WAN-ympäristössä [Moon *et al.*, 2006]. Atlantin ylitys, tai paremminkin alitus, tapahtuu valon nopeuden ja kaapelissa tapahtuvan elektronisen signaalin hidastumisen takia vähintäänkin 25-30 ms viivellä [Smed and Hakonen, 2006]. Reititys, jonotus ja pakettien käsittely lisäävät jokaisessa reitin solmussa viivettä entisestään. Langattomien matkapuhelinverkkojen suuret, jopa sekun-

tien, viiveet aiheuttavat todellisia ongelmia reaaliaikaa tavoitteleville peleille.

Viive voi vaihdella eikä siis ole vakio koko ajan. Tätä vaihtelua nimitetään *huojunnaksi (jitter)*. Välillä viive voi olla siedettävä pelin kannalta mutta hetken kuluttua viive on voinut kasvaa liian suureksi. Viive ja kaista voivat olla eri suuruksia eri suuntiin. Matkapuhelinten kohdalla laitteesta lähettämisen (*upload*) viive voi olla huomattavasti suurempi kuin pakettien vastaanottamisen (*download*) viive. *Kiertoaika (RTT, round-trip time, ping)* kuvaa aikaa, joka paketilta menee lähettävältä laitteelta vastaanottavalle ja takaisin. Paketteja voi myös kadota matkalle. Tätä hukkumista kutsutaan pakettien *häviöksi (loss)*. Häviö liittyy verkon luotettavuuteen (*reliability*).

Paketit voivat myös saapua perille siten, että niiden sisältö on muuttunut. Sisällön muuttumista kutsutaan korruptioksi ja se on yhdessä häviön kanssa luotettavuutta laskeva tekijä. Tyypillisesti Internetissä vallitsevasta viiveestä ja häviöstä pelaajat kokevat viiveen tärkeämpänä tekijänä, joka vaikuttaa heidän pelikokemukseensa ja pelisuoritukseensa [Zander and Armitage, 2004]. Tämä on oikeastaan aika odotettu tulos, sillä pelaajalla on pelatessaan aika hankalaa erottaa häviön ja viiveen vaikutukset toisistaan. Luotettavuus kertoo siis, kuinka paljon dataa häviää verkossa siirron aikana. Paketin hävitessä tai ollessa korruptoitunut se yksinkertaisesti unohdetaan tai sitten se lähetetään uudelleen. Jos paketti joudutaan lähettämään uudelleen, viive vähintäänkin tuplaantuu kyseiselle paketille. Yleensä viive kasvaa enemmänkin, koska hävinnyttä pakettia ei heti tiedetä hävinneeksi. Kadonneet ja unohdetut paketit eivät sinällään kasvata viivettä, mutta pelaaja saattaa huomata ne pelin outona käytöksenä. Kohteet, esimerkiksi hahmot, voivat pomppia oudosti näytöllä, kun niiden jotkut sijaintitietopakettit ovat kadonneet matkalla. Paketin uudelleenlähetys ja unohtaminen on kiinni käytetystä protokollasta. Mobiilipelien käytetyimmät tiedonsiirtoprotokollat ovat SMS ja MMS, WAP, TCP, UDP ja SIP. Tutkielman kannalta tärkeimmät ovat kolme viimeistä, koska niillä päästään edes yrittämään reaaliaikaista moninpeliä. Protokollat käyn lyhyesti tarkemmin läpi alaluvussa 4.2.1 sivulla 49.

Minkälaisia kaistavaatimuksia peleillä on? Asiakkaan lähettämät paketit palvelimelle ovat noin 60 tavun kokoisia ja palvelimen vastausviestit ovat noin 130-tavuisia kahdessa eri FPS-pelissä [Jehaes *et al.*, 2003]. Toisessa tutkimuksessa kahdelle eri tyyppiselle pelille saatiin pakettien kooksi 120 ja 160 tavua. Pakettien lähetystiheydessä oli selviä eroja, toisen pelin lähettäessä paketteja tasaisesti 10-300 ms välillä, toisen liikenne ruuhkautui välillä nopeaan frekvenssiin. Noin 30 prosenttia paketeista tuli selvissä purskeissa. Toisessa pelis-

sä palvelimen lähettämät viestit olivat selvästi isompia kuin asiakkaan paketit. Palvelimen paketeista 10 prosenttia oli yli 500-tavuisia. Tämäkin tavumäärä on kuitenkin selvästi alempi kuin Internetin muun liikenteen keskimääräinen pakettien koko, joka on yli 400 tavua ja mediaani on noin 1500 tavussa. [Claypool *et al.*, 2003].

Pakettien kooksi valitaan 160 tavua eli 1280 bittiä. Tämän voi pyöristää vielä ylöspäin 1,5 kilobittiin. Viestien lähetystaajuuden pyöristän alaspäin eli tiukemmaksi 50 ms:iin. Tällöin viestejä lähetetään 20 sekunnissa. Kaistan tarpeeksi saadaan näillä arvoilla noin 30 kbps. Kahdella tutkitulla pelillä keskiarvo kaistan tarpeelle oli 5,2 - 29,2 kbps [Claypool *et al.*, 2003]. Sami Mäkeläinen [2006] arvioi pelien kaistan tarpeen olevan alle 64 kbps. Mäkeläinen ennustaa, että maksimikaistan tarve FPS-peleille olisi tulevaisuudessa 128 kbps. Mäkeläinen jatkaa, että kaistavaatimuksen alaraja riippuu erittäin paljon peleistä ja pelityypeistä. Hänen käyttämänsä ylärajat ovat mielestäni korkeita nykyisten pelienkin valossa. Taulukossa 3.1 on listattuna pelien mainostamia kaistanleveyksiä, joita ne vaativat toimiakseen [Henderson, 2003]. Ottaen huomioon tutkimuksen ajankohdan, pelien kaistojen tarpeet on ilmoitettu vallitsevien verkkotekniikoiden mukaisesti. Näitä olivat modeemi- ja ISDN -yhteydet. Tästä voidaan päätellä, että pelien kaistantarve ei aivan yltänyt mainittuihin arvoihin saakka vaan pelit toimivat hyvin alle taulukossa mainittujen arvojen.

Taulukko 3.1: Pelien vaatimia kaistanleveyksiä

Peli	Mainostettu kaistan tarve
<i>The Sims</i>	28,8 kbps
<i>Half-Life: Generations</i>	28,8 kbps
<i>Medal Of Honour</i>	33,6 kbps
<i>Dungeon Siege</i>	56 kbps
<i>FIFA 2002 World Cup</i>	56 kbps

Kaistan lisäksi toinen tärkeä seikka on pelien viiveettömyysvaatimukset. Intensiivisessä FPS-tyyppisessä peleissä on mitattu, että tarkkuutta vaativissa tilanteissa, kuten amunnassa, jo vaatimaton (75-100 ms) viive vaikeuttaa osumista huomattavasti. Edellä mainitun kokoinen viive normaalitilanteeseen verrattuna huonontaa tarkkuutta 50 prosentilla. Samassa tutkimuksessa on todettu, että pelin subjektiivinen laatu heikkenee ja viive on havaittavissa noin 100 millisekunnissa. 200 ms viive on jo ärsyttävä. [Beigbeder *et al.*, 2004].

FPS-pelit ovat viiveettömyysvaatimusten tiukimmassa luokassa, urheilupelit keskellä ja reaaliaikaiset strategiapelit löyhimmässä luokassa. Urheilupeli, kuten amerikkalainen jalkapallo, näyttää sietävän hyvin viiveitä jopa 500 ms:n saakka ilman, että pelaajien suoritukset siitä paljoa kärsivät. Yli 500 ms:n viivet alkavat sitten laskea suoritusta jo 30 prosentilla. [Nichols and Claypool, 2004]. Ajopeleissä viiveen tulisi olla alle 100 ms, vaikka 200 ms:n viivellä auto voi olla vielä hallittavissa. 200 ms:ssa pelaaja näkee jo viiveen vaikutuksen selvästi. [Pantel and Wolf, 2002a]. Reaaliaikaisten strategiapeliin viiveen sieto on korkea. Vielä 500 ms:n viiveetkään eivät pilaa pelikokemusta. Pelaaja alkaa kärsiä viiveistä toden teolla vasta 800 ms:n viiveen paikkeilla. Sinällään tulos ei ole yllättävä, sillä strateginen ajattelu ei vaadi sellaista nopeutta kuin nopea toiminta FPS-peleissä. Strategioiden muodostaminen ja ennenkaikkea toteuttaminen, olkoonkin ne reaaliaikaisia strategioita, vie pelaajalta sekunneista jopa minuutteihin. Tällöin alle puolen sekunnin viiveet eivät ole enää häiritseviä. [Sheldon *et al.*, 2003].

Smedin ja Hakosen [2006] mielestä hyvä perussääntö on, että viive väliltä 0,1-1,0 sekuntia eli 100-1000ms on hyväksyttävissä reaaliaikaisille järjestelmille kuten tietokonepeleille. Heidän mukaansa DIS-standardi sotilassimulaatioille määrittelee, että viiveen tulisi olla alle 100 ms. Smed ja Hakonen jatkavat, että jatkuvan sulavan liikkeen hallinta on mahdollista alle 200 ms viiveellä. Smed ja Hakonen vahvistavat, että reaaliaikastrategiapeleissä jopa 500 ms viiveet ovat hyväksyttäviä, kunhan huojunta pysyy vähäisenä. Samaten heidän mielestään FPS-pelien viiveen sieto on vähäisempi ja viiveen tulisi olla 100 ms paikkeilla. Smed ja Hakonen jatkavat kertomalla, että usea tutkimus on linjannut samansuuntaisia havaintoja kollaboratiivisissa virtuaaliympäristössä (collaborative virtual environments). Taulukkoon 3.2 on tiivistetty eri pelityyppien viivevaatimukset lisättynä Mäkeläisen [2006] arviolla vuoropohjaisten pelien viivevaatimuksista.

3.3 Matkapuhelinverkot

Tämän tutkielman kannalta keskeisimmät verkot ovat matkaviestinverkot. Ne tarjoavat pääsyn Internetiin lähes kaikkialta. Matkaviestinverkot palvelevat täten parhaiten mobiilipelien pelaajien tärkeintä arvoa, eli kohdassa 2.2 mainittua mobiiliutta. Vaikka tutkielman rajaus koskettaa matkapuhelinverkkoja, voitaisiin rajaus tehdä vielä tiukemmin koskemaan 2G-verkkoja pienimpänä yhteisenä nimittäjänä kaistan ja viiveen suhteen. Viestintäviraston telepalve-

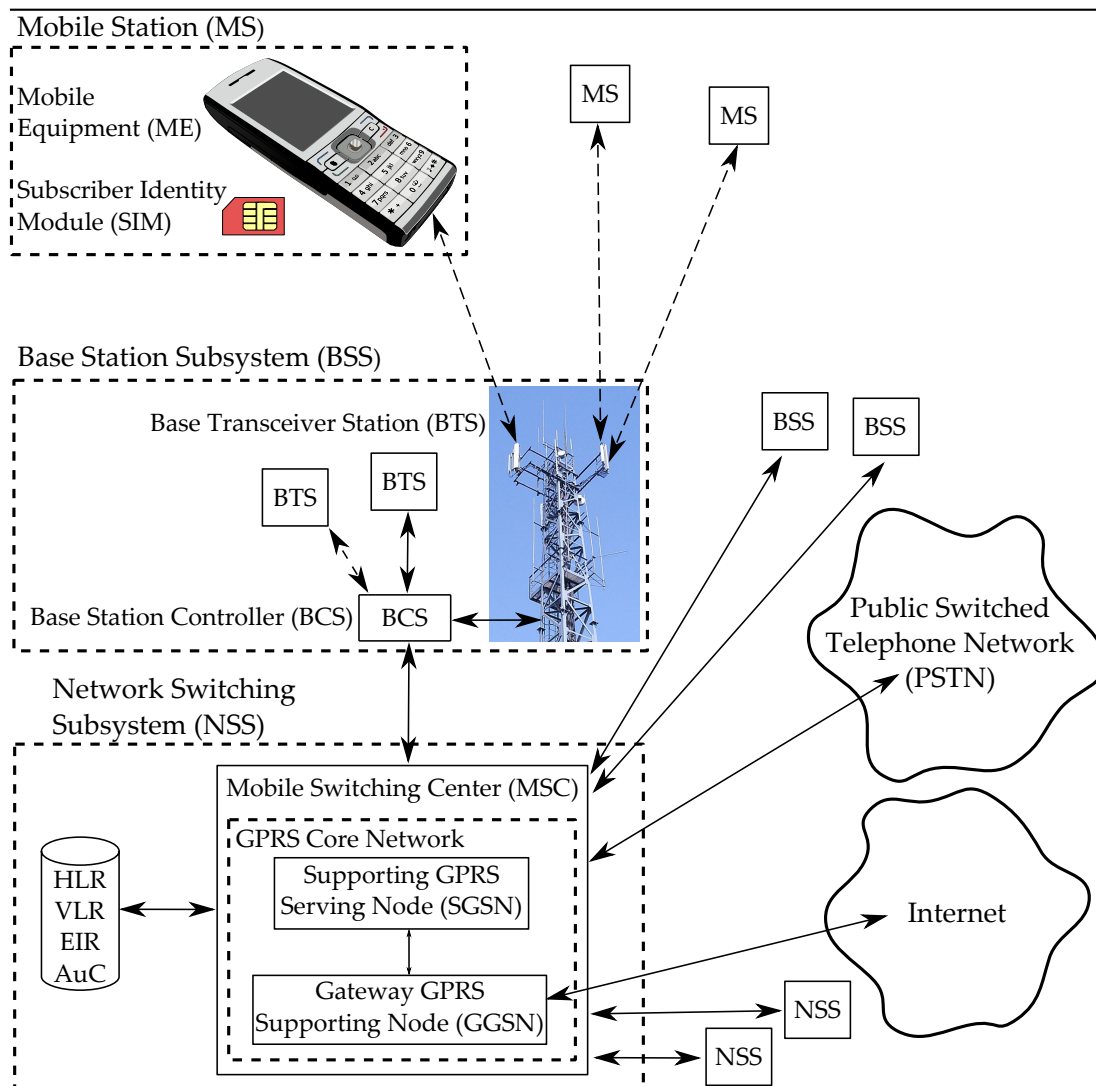
Taulukko 3.2: Pelityyppien viivevaatimukset

Pelityyppi	Maksimi viive
Sotilassimulaatio	100 ms
FPS	100 ms
Ajo	150 ms
Urheilu	500 ms
Reaaliaika strategia	600 ms
Vuoropohjainen	5000 ms

luiden käyttötutkimuksen [Lagerström, 2008] mukaan vasta noin 23 prosentilla kotitalouksista on 3G-matkapuhelin. Tätä lukua tukee edellä esittelemäni maailmanlaajuinen arvio, jossa GSM:n levinneisyyden arvioitiin olevan noin 85 prosenttia.

Viestintäviraston telepalveluiden käyttötutkimuksen mukaan suurimmalla osalla 3G-puhelimen hankinta ei vaikuta mobiilipalveluiden käyttöön eikä mobiilipalveluiden käytön uskota juurikaan muuttuvan lähitulevaisuudessa. Langattomien lähiverkkojen ja niiden liityntäpisteiden määrä on noussut, mutta nykyisin on jo alettu epäillä liityntäpisteille käyvän samoin kuten puhelinkopeille. Ne katoavat, koska uudemmat matkapuhelinverkot tekevät ne tarpeettomiksi [Pietarinen, 2008]. Tulevat kaupunkiverkot voivat sen sijaan tarjota riittävän nopeuden ja riittävän peiton, jotta ne olisivat kaupunkilaispelaajille riittävän kattavia. Tällä hetkellä tarjolla olevista WiMAX- tai 450-verkon langattomista palveluista on kuitenkin tietoinen vasta alle joka kymmenes kansalainen [Lagerström, 2008].

Kuvassa 3.2 on esitetty tyypillinen GSM-verkon rakenne GPRS-komponenttien kera. Matkapuhelin (ME) ja siihen liitetty SIM-kortti muodostavat MS:in. Tukiasema (BTS) voi olla langattomasti radiosiirtotietä käyttäen yhteydessä useaan matkapuhelimeen. Tukiasemaa ohjaa tukiasemaohjain (BCS), jolla voi olla ohjattavanaan useita tukiasemia. Tukiasemaohjain välittää liikenteen edelleen NSS:in MSC:lle. MSC voi hallinnoida useita BSS:tä. MSC:in osana on myös GPRS toiminnallisuus, jonka kautta matkapuhelimella on lopulta pääsy Internetiin. MSC voi olla yhteydessä muiden NSS:en MSC:in. NSS sisältää lisäksi erilaisia rekisteitä: home location register (HLR), visitor location register (VLR), equipment identification register (EIR) ja authentication center (AuC). Kuvaa on yksinkertaistettu rekisterien osalta huomattavasti. Rekisterit eivät



Kuva 3.2: GSM-verkon rakenne

aina ole kuvan mukaisesti yhdessä. MSC hoitaa yhteydet myös esimerkiksi kiinteään puhelinverkkoon (PSTN). Esimerkkikuva on yksinkertaistettu ja muutoksia rakenteeseen tuovat operaattoreiden erilaiset valinnat heidän verkonsa toteutuksesta [Ahmad, 2005, Pulse, 2006, Wikipedia, 2006].

3.3.1 Kaista

Taulukkoon 3.3 on koottu eri matkapuhelinverkkotekniikoiden teoreettisia maksimilähetys- ja vastaanottoaistan leveyksiä eri lähteistä [Kreher and Rüdibusch, 2007, Mäkeläinen, 2006, GSM Association, 2007, Hentunen, 2007, Wi-

kipedia, 2008b]. Taulukossa on myös joitakin tyypillisiä mitattuja sekä ennakoituja vastaanottoaikaisten leveyksiä. Taulukossa on mallin vuoksi myös muutamia kiinteiden ja muiden langattomien verkkojen tietoja. Samankin verkko-tekniikan hieman erilaiset toteutukset, eri kokoonpanot ja asetukset sekä olosuhteet aiheuttavat heittoa lukuihin [Mäkeläinen, 2006]. Lukujen epätarkkuutta lisää se, että toiset lähteet mainitsevat teoreettisen huippukaistanleveyden ja toiset ylläpidettävissä olevan nopeuden. Luvut näyttävät kuitenkin yleisen suunnan verkon kapasiteetista.

Taulukko 3.3: Verkkojen kaistanleveys

Verkon tyyppi	Lähetys	Vastaanotto	Mitattu vastaanotto
GPRS	14-40 kbps	28-171,2 kbps	30-40 kbps
GSM CSD	9,6-14 kbps	9,6-14 kbps	-
HSCSD	28-43 kbps	28-58 kbps	-
EDGE	59 kbps	200-473,6 kbps	128 kbps
EDGE Evolution	0,9 Mbps	1,9 Mbps	-
UMTS	64 kbps	384 kbps	300 kbps
WCDMA	384 kbps	384 kbps	128-384 kbps
HSDPA + HSUPA	1,8-7,2 Mbps	3,6-14,4 Mbps	1 - 2 Mbps
HSPA+	11,5 Mbps	42 Mbps	-
CDMAone (IS-95B)	-	115,2 kbps	10-40 kbps
CDMA2000 1x	144 kbps	144-614 kbps	80-100 kbps
CDMA2000 1x EV-DO Rev 0	150 kbps	2480 kbps	600-1000 kbps
Modeemi	56 kbps	56 kbps	-
ISDN vakio	64 kbps	64 kbps	-
ADSL	256 kbps	512 kbps	-
Laajakaista	1-100 Mbps	1-100 Mbps	-
WiMAX	75 Mbps	75 Mbps	2 Mbps
WLAN	11-54 Mbps	11-54 Mbps	-
Bluetooth	1-3 Mbps	1-3 Mbps	-

Vertailtaessa taulukossa esitettyjä mitattuja tai ennakoituja kaistanleveyksiä teoreettisiin maksimiarvoihin huomataan, että todelliset suoritusarvot ovat kaukana jäljessä teoriasta. Mäkeläinen [2006] esittelee yleissäännön tälle poik-

keammalle todetessaan, että todellinen teho mobiileille dataverkoille on vain noin 10-30 prosenttia ilmoitetusta teoreettisesta maksimista. Syiksi sille, että teoreettista maksimia ei saavuteta käytännössä, Mäkeläinen luettelee seuraavat seikat: teoreettinen maksimi olettaa vain yhden käyttäjän solua kohti, teoreettinen maksimi olettaa ideaalit radio-olosuhteet tukiaseman ja päätelaitteen välille, erilaisia standardisoituja koodausmalleja ei välttämättä ikinä toteuteta käytännössä ja teoreettinen maksimi olettaa jonkun tietyn konfiguraation. Teoreettinen maksimi lataukselle ja lähetykselle ei välttämättä ole saavutettavissa samalla konfiguraatiolla. Lisäksi yhteyden vaikuttaa se, onko matkapuhelin paikallaan vai liikkeessä. Myös liikkeen nopeudella on merkitystä.

Verrattaessa taulukon 3.3 lukuja kohdassa 3.2 mainittuihin pelien kaistan tarpeisiin huomataan, että kaistan suhteen peleillä ei ole ongelmia. Lisäksi, kun huomioidaan aiemmin mainittu viestien koon asymmetrisyys – eli palvelimelta tulevat viestit ovat isompia kuin asiakkaan lähettämät – huomataan, että tämäkin sopii matkapuhelinverkoille hyvin. Matkapuhelinverkoissahan kaistaa on yleensä enemmän lataus- kuin lähetyssuuntaan.

3.3.2 Viive, huojunta ja häviö

Taulukko 3.4: Kiinteiden verkkojen viiveitä

Verkon tyyppi	Viive
Ethernet	0,3 ms
Analoginen modeemi	100 - 200 ms
ISDN	15 - 30 ms
DSL/Cable	10 - 60 ms
Paikallaan oleva satelliitti	500 ms
DS1/T1	2 - 5 ms
Valon nopeus valokaapelissa aikavyöhykkeittäin	8,25 ms

Kohdassa 3.2 tarkastelin pelien viiveettömyysvaatimuksia. Matkapuhelinverkkojen viiveistä puhuttaessa on hyvä verrata niitä kiinteiden verkkojen viiveisiin. Taulukossa 3.4 on esitettyä tyypillisiä viiveitä erilaisille kiinteille verkoille. Taulukko on peräisin Zydalta [2005c] ja siitä löytyy myös yksi langattoman verkon viive: Paikallaan olevan satelliitin viive on noin 500 ms. Sen viive johtuu pääasiassa suuresta välimatkasta eli kiertoradan korkeudesta. Kun tau-

lukon 3.4 arvoja verrataan matkapuhelinverkkojen viiveet tiivistävään tauluk-
koon 3.5, huomataan, että matkapuhelinverkkojen viiveet ovat kertaluokkaa
suurempia kuin kiinteiden verkkojen viiveet. Taulukko 3.5 on koostettu kah-
desta lähteestä [Kreher and Rüdebusch, 2007, Holma *et al.*, 2006, Mäkeläinen,
2006, Derksen *et al.*, 2006, Lee, 2006]. Tämän vuoksi keskityn tutkielmassani
nimenomaan matkapuhelinverkkoihin. Jos pelin saa toimimaan riittävän no-
peasti matkapuhelinverkolla, saa sen toimimaan erittäin todennäköisesti myös
muilla verkoilla. Tällöinkin on pelisuunnittelussa huomioitava se seikka, että
nopeamman verkon avulla pelaava pelaaja ei saa saada pelin tasapainoa pilaa-
vaa hyötyä nopeamman verkon käytöstä.

Taulukko 3.5: Matkapuhelinverkkojen viiveitä

Verkon tyyppi	Viive min	Viive max	Tyypillinen mitattu kiertoaika
GPRS	400 ms	1000 ms	700-1100 ms
EDGE	300 ms	700 ms	500-600 ms
CDMA2000 1x	-	-	500-800 ms
CDMA2000 1x EV-DO Rev 0	140 ms	1700 ms	400-600 ms
UMTS	170 ms	300 ms	-
WCDMA	100 ms	200 ms	250 ms
HSDPA	75 ms	100 ms	70-95 ms
HSUPA	40 ms	80 ms	-

Huojunta voi olla huomattavaa matkaviestinverkoissa. Huojuntaa on tut-
kittu kuitenkin huomattavasti vähemmän kuin viivettä. Kritiikkiä on myös esi-
tetty joitakin tutkimuksia kohtaan siitä, että Internetissä olevien oikeiden peli-
palvelimien logeja tarkastelemalla ei saada selvää kuvaa viiveen ja huojunnan
erottamiseksi [Armitage and Stewart, 2004]. Matkapuhelinverkon huojuntaa
lisää radiotien muutoksien lisäksi muun muassa matkapuhelimen siirtyminen
(handover) tukiasemalta toiselle. Matkapuhelimella ei myöskään ole kiinteän
verkon tapaan koko ajan avoinna olevaa datayhteyttä, vaan se pitää käynnis-
tää aina, kun tietoja halutaan alkaa siirtää. GPRS-yhteys ei siis ole muodos-
tettu valmiiksi silloin, kun matkapuhelin käynnistetään. Osittain tukiasemal-
ta toiselle siirtyvä matkapuhelin kärsii tästä samasta viiveestä, koska yhteys
pitää muodostaa uuden tukiaseman kanssa. Mäkeläinen [2006] sanoo, että tä-
mä GPRS:n alussa muodostama PDP-konteksti GPRS:n ydinverkon kanssa vie

useita sekunteja. Omasta kokemuksesta sanoisin GPRS-yhteyden muodostamisen vievän puolesta minuutista minuuttiin. UMTS-yhteyden muodostaminen on huomattavasti nopeampaa.

Beigbeder *et al.* [2004] tutkimuksessa todetaan, että häviöllä ei ole niin suurta merkitystä FPS-tyyppiselle pelille. Jopa 5 prosentin häviö jäi pelaajilta huomaamatta. Mobiilipelaajien voisi olettaa huomaavan häviön vaikutukset helpommin, koska viive on matkapuhelinverkoissa huomattavaa. Pakettien uudelleen lähettäminen häviön tapahtuessa kärsii pitkistä viiveistä paljon. Tämä on yksi syy, miksi mobiilipelien ei pitäisi käyttää protokollia, joissa paketit lähetetään uudelleen niiden hävitessä. Myöskään mobiilipelien ei pitäisi perustua ensinnäkään ajatukselle, että kaikkien viestien on tultava läpi.

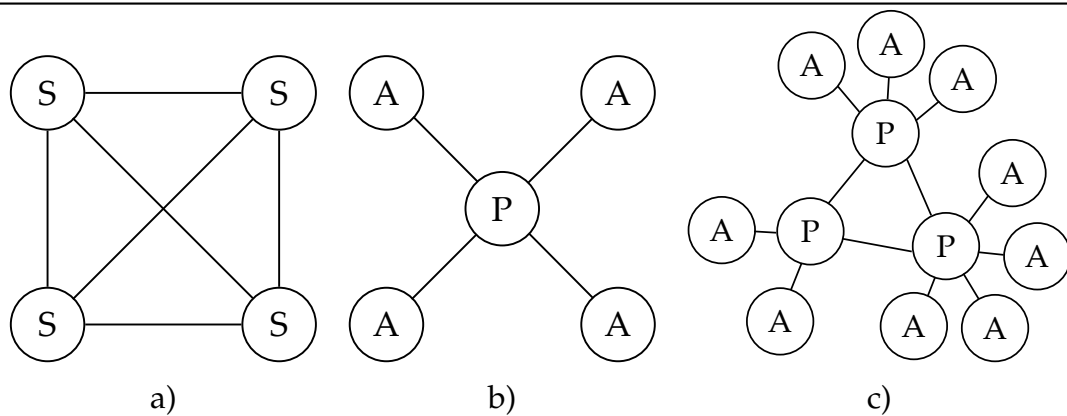
Matkapuhelimien heterogeenisyys samoin kuin verkkojen heterogeenisyys asettavat pelaajat eriarvoiseen asemaan sekä koetun pelikokemuksen että myös pelisuorituksen suhteen. Erityisesti viive koetaan eriarvoisuutta lisääväksi tekijäksi. Tämän vuoksi onkin esitetty, että pelin pitäisi ottaa tämä huomioon ja sisältää esimerkiksi automaattisen viiveen tasapainotusmekanismin, joka takaisi pelaajille tasavertaiset mahdollisuudet pelata [Zander *et al.*, 2005]. Tasapainotusmekanismin toteuttaminen lisää kuitenkin entisestään kehittäjien työtä, jota on jo muutenkin huomattavissa määrin juuri laitekannan heterogeenisyyden vuoksi. Ongelmaa voisi yrittää ratkaista myös pelisuunnittelullisesti, jolloin ohjelmiston kompleksisuutta saataisiin pienemmäksi.

3.4 Mobiilipelien verkkoarkkitehtuurit

Loogisen tason ratkaisut määräävät mobiilipelin kommunikointiarkkitehtuurin. Data- ja hallinnointiarkkitehtuurit kuuluvat myös loogiselle tasolle mutta niihin palaan kohdassa 4.1 sivulla 41, jossa käsitellään aiheeseen tiukasti sidoksissa olevia konsistenssin, vasteen ja skaalautuvuuden kysymyksiä. Verkkoarkkitehtuurille löytyy useita malleja. Yksinkertaisimmillaan kyseessä on vain yksi solmu ilman yhteyksiä mutta moninpelien kannalta se ei anna kovinkaan paljon mahdollisuuksia. Historiakatsauksesta muistetaan toki jaetun ruudun ja hotseat-tyyppiset moninpelit mutta ne eivät tue mobiliteettiä. Matkapuhelimilla toteutetut moninpelit perustuvat siihen, että jokainen pelaaja käyttää omaa matkapuhelinta. Täten kommunikointiarkkitehtuuri käsittää useita solmuja. Jos solmut kommunikoivat suoraan keskenään, on kyseessä *vertaisverkko* (peer-to-peer, P2P). Vertaisverkkoarkkitehtuuri on kuvassa 3.3 a). Verkosta voidaan myös nostaa yksi solmu keskeisemmäksi ja nimittää sitä pal-

velimeksi. Tällöin kyseessä on asiakaspalvelinarkkitehtuuri.

Monenpelattavat verkkopelit olivat alkuaikoina vertaisverkon varaan rakennettuja. Selitys tälle oli se, että vertaisverkkoratkaisu oli melko suoraviivaista toteuttaa yhdellä solmulla toimivasta yksinpelistä. Vertaisverkossa tasa-arvoiset solmut kommunikoivat toistensa kanssa. Jokainen solmu voi välittää viestejä suoraan mille tahansa toiselle solmulle. Vertaisverkkoarkkitehtuuri muodostuu ongelmalliseksi solmujen määrän noustessa, jolloin kaistan tarve kasvaa lineaarisesti. Vertaisverkkoarkkitehtuuri soveltuukin peleille, joissa pelaajien määrä on pieni tai verkko on nopea, kuten LAN-toteutuksissa. [Smed and Hakonen, 2006].



Kuva 3.3: Verkkoarkkitehtuurit: a) vertaisverkko, b) asiakas-palvelin ja c) palvelinverkko

Vertaisverkkojen skaalautuvuusongelmaa voidaan yrittää kiertää *asiakas-palvelin-arkkitehtuurilla* (client-server), jossa yksittäinen solmu kommunikoi vain palvelimen kanssa. Tämä on esitetty kuvassa 3.3 b). Palvelin on nostettu erikoistettuun rooliin verrattaessa muihin solmuihin. Muut solmut ovat tällöin asiakkaita, jotka ovat keskenään tasa-arvoisia. Palvelin välittää viestit muille solmuille eli asiakkaille. Koska kaikki viestit kulkevat palvelimen kautta, saadaan viestien virtaa hallinnoitua. Tämä avaa mahdollisuuksia optimointiin esimerkiksi siten, että palvelin ei välitäkään kaikkia viestejä kaikille solmuille. Toisaalta optimoinnille on myös tarvetta, koska solmujen välisen kommunikoinnin on kierrettävä aina palvelimen kautta. Tämä lisää viestien viivettä. Yksi palvelimen tarjoamista eduista on hallinnointiominaisuudet, joita kaikkien viestien kulkeminen sen kautta tarjoaa. Peleissä tämä voi tarkoittaa esimerkiksi mahdollisuutta kuukausiperusteiseen laskuttamiseen tai laittomilla oh-

jelmakopioilla pelaavien pelaajien pelaamisen estämiseen. [Smed and Hakonen, 2006]. Huijauksen estäminen on myös yksi palvelimen tarjoamista eduista [Majewski *et al.*, 2006]. Toisaalta asiakaspalvelin-arkkitehtuurin ongelmakohta on myös palvelin, jonka kaistan tarve kasvaa asiakkaiden määrän lisääntyessä sekä palvelimen muodostama yhden kohteen vikaantumisen riski. Palvelimen vikaantuessa koko palvelu on poissa käytöstä. [Smed and Hakonen, 2006]. Palvelimen tarjoama etu on myös pelitilan tai -maailman mahdollinen jatkuvuus. Pelitilan jatkuvuuden määrittelyn kohdassa 2.3. Jatkuvan pelimaailman toteuttaminen vertaisverkkoarkkitehtuurilla on erittäin haastavaa, koska tilaa ei säilytetä missään keskitetysti.

Asiakaspalvelin- ja vertaisverkkoarkkitehtuurien välimuotoja on useita. Vertaisverkkoon voidaan sisällyttää palvelimen kaltainen komponentti – *keskusvälittäjä* (central arbiter). Solmut kommunikoivat suoraan toistensa kanssa mutta välittävät tiedon myös keskusvälittäjälle. [Pellegrino and Dovrolis, 2003]. Asiakaspalvelin-arkkitehtuurin palvelin voidaan edelleen hajauttaa tai monistaa useammaksi solmuksi. Kyseessä on tällöin *palvelinverkko*, jossa asiakas ottaa yhteyden johonkin palvelinverkon palvelimista. Palvelinverkossa yksittäisen palvelimen kuorma saadaan jaettua useammalle palvelimelle. Palvelimien välinen kommunikointiarkkitehtuuri voi olla vertaisverkko tai jokin hierarkkisempi malli. Palvelinverkko on kuvattu kuvassa 3.3 c).

Palvelinta päästään tuskin koskaan asentamaan operaattorin verkkojen sisälle. Tämän estävät muun muassa turvallisuussyt. Otettaessa huomioon matkapuhelinverkon rakenne (ks. kuva 3.2), palvelimen tulisi tällöin olla mahdollisimman lähellä GGSN:a. Tämä on tarpeellista, ettei GGSN:n ja Internetissä sijaitsevan palvelimen viive, vaikkakin se on todennäköisesti vähäisempi kuin radiotiestä johtuva viive, lisäisi kokonaisviivettä tarpeettomasti. Pelaajat voivat olla useiden eri operaattorien asiakkaita. Tällöin pelipalvelin tulisi saada mahdollisimman lähelle jokaisen operaattorin GGSN:a. Tämä implikoisi, että paras arkkitehtuuriratkaisu olisi jonkinlainen palvelinverkko. Näin jokainen pelaaja olisi topologisesti lähellä oman operaattorin lähetyvillä olevaa palvelinta. Pelipalvelimet kommunikoisivat keskenään nopeita kiinteitä verkkoja käyttäen. Tässä ajatuksessa on tietenkin oletettu, että peli halutaan mahdollistaa usean operaattorin asiakkaille mahdollisimman laajan asiakaspeiton saamiseksi.

4 PELIN TILAN SYNKRONOINTI

Matkapuhelin on laitteena huomattavan rajoittunut resurssien suhteen. Kuten jo kohdassa 2.4 totesin, matkapuhelimien suorittimet ovat vaatimattomia, muistia on niukasti puhumattakaan käytettävyyssrajoitteista. Kohdassa 3 kerroin verkkojen rajoituksista. Smed ja Hakonen sanovat kirjassaan *Algorithms and Networking for Computer Games* [2006], että verkkojen käyttö on aina riippuvaista resurssirajoituksista. Smedin ja Hakosen kirja on tuore ja kattava verkkopelien algoritmeja, arkkitehtuureja ja kompensatiotekniikoita käsittelevä teos. Joitakin muita teoksia löytyy myös mutta niissä pääpaino on Smedin ja Hakosen kirjaa enemmän pelien sijasta virtuaaliympäristöissä. Smed ja Hakonen luettelevat Singhalin vuonna 1996 ilmestyneen väitöskirjan perusteella kolme resurssirajoitusta, jotka verkkoa hyödyntävä ohjelmisto kohtaa: kaistanleveys, viive ja solmujen prosessointiteho verkkoliikenteelle. Näistä lähtökohdista ajateltuna verkkoa käyttävät mobiilipelit ovat todella ahtaalla käytettävissä olevien resursien suhteen.

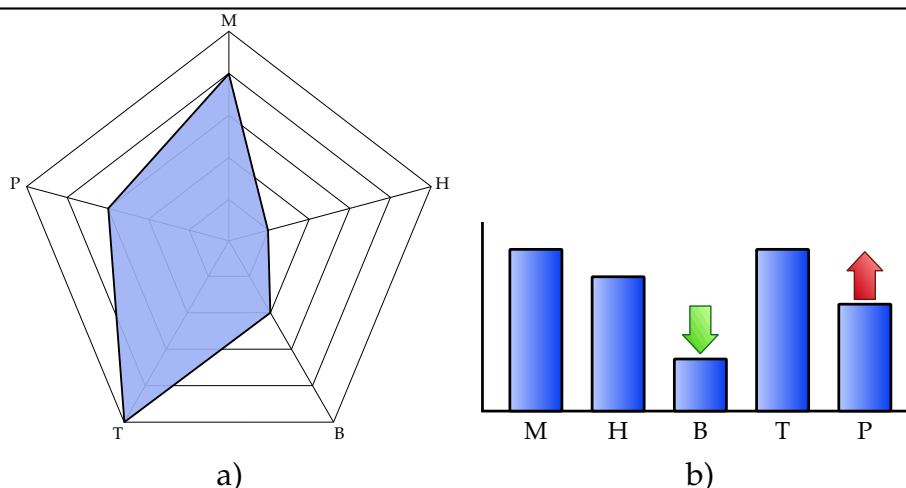
Tarvittavien resurssien määrä verkkopelille on suoraan verrannollinen siihen, kuinka paljon tietoa on lähetettävä ja vastaanotettava jokaisella solmulla ja kuinka nopeasti verkon on se välitettävä [Smed and Hakonen, 2006]. Tätä lakia Smed ja Hakonen nimittävät informaation periaatteeksi. Smed ja Hakonen viittaavat Singhalin ja Zydan vuonna 1999 ilmestyneeseen kirjaan *Networked Virtual Environments: Design and Implementation* esitellessään informaation periaatteen ja sitä konkretisoivan yhtälön. Informaation periaatteen yhtälö (*information principle equation*) auttaa tarkastelemaan rajoitettuja resursseja ja havainnollistaa, mitä parannukset joillakin osa-alueilla aiheuttavat muualla. Yhtälö on seuraavanlainen:

$$\text{Resurssit} = M \times H \times B \times T \times P. \quad (4.1)$$

Yhtälössä M on lähetettyjen viestien määrä (Messages), H on keskimääräinen vastaanottavien solmujen lukumäärä (Hosts), B on keskimääräinen kaistan määrä viestille jokaiselle vastaanottavalle solmulle (Bandwidth), T on ajallisuus (Timeliness) ja P on tarvittava prosessorin jaksojen lukumäärä viestien vastaanottamiseen sekä käsittelyyn (Processor). Ajallisuus on aika, jona verkon on välitettävä viesti jokaiselle solmulle. Suuret T arvot kuvaavat tilannetta, jossa viiveen tulee olla pieni ja toisin päin. [Smed and Hakonen, 2006].

Kaavan 4.1 muuttujien arvoja voidaan havainnollistaa kahdella tavalla. Smedin ja Hakosen esittelevät kirjassa tutkakuvion ja Smed käyttää Multiplay-

er Computer Games -kurssin kalvoissaan [Smed, 2005] lisäksi pylväsdiagrammia. Tutkakuvion esimerkki näkyy kuvassa 4.1 a) ja pylväsdiagrammilla havainnollistaminen kuvassa 4.1 b).



Kuva 4.1: a) Tutkakuvion esimerkki ja b) Viestien tiivistäminen vaikutus

Yhtälöstä huomataan, että mikäli käytettävissä olevien resurssien määrä tai resurssien määrän vaatimus on kiinnitetty eli vakio, verkkomobiilipelissä on tietyn tasoisia ominaisuuksia. Ominaisuuksia ovat esimerkiksi vaste ja skaalautuvuus. Kaavalla 4.1 on useita mahdollisia ratkaisuja resurssien ollessa vakio. Tällaisessa tilanteessa suunnittelija pystyy käyttämään yhtälöä työkaluna tasapainottaessaan toteutuksen vaatimuksia ja rajoitteita. Yhtä muuttujaa muutettaessa se täytyy kompensoida muuttamalla myös jonkin tai joidenkin muiden muuttujien arvoja. Tästä on seurauksena se, että jos jotain muuttujan arvoa halutaan pienentää, joku toinen tai useat muuttujat kasvavat arvoltaan. Näin ollen pelin jonkun osa-alueen parantuessa jokin toinen heikkenee tai pelikokemus heikentyy. Kuvassa 4.1 b) havainnollistetaan protokollan optimointitilannetta, jossa viestejä on tiivistetty pienemmäksi. *B* eli kaistan tarve vähenee mutta samalla prosessointiin vaaditaan enemmän resursseja. [Smed and Hakonen, 2006]. Prosessointi lisääntyy, koska viestien tiivistäminen tai pakkaaminen ei tapahdu ilman ylimääräistä prosessointia.

4.1 Konsistenssi, vaste ja skaalautuvuus

Hajautetut interaktiiviset sovellukset, kuten pelit, ovat ongelmallisia hajautetun tilan hallinnan ja toisaalta riittävän nopean vasteen suhteen. Hajautettua

tilaa käytettäessä pelaajan ja hänen laitteensa hallinnoiman pelitilan välinen vaste pienenee. Näin tapahtuu, koska pelitilaa muuttavia tietoja ei tarvitse välittää verkon yli johonkin keskitettyyn pelitilaan ja täten verkon aiheuttama viive eliminoiduu. Ongelmaksi muodostuu kuitenkin hajautetun tilan hallinta ja yhden pelaajan pelitilan muutoksien välittäminen muillekin pelaajille. Jos tilat ovat kaikilla pelaajilla yhtenevät eikä eroja ole, tilat ovat *konsistenssit*. Smed ja Hakonen [2006] sanovat konsistenssin (*consistency*) tarkoittavan yhtäläisyyttä datan näkymään, joka verkon solmuilla on. He toteavat myös, että absoluuttinen konsistenssi kuvaa tilannetta, jossa jokaisella solmulla on yhtenevä data. Jotta absoluuttinen konsistenssi saataisiin ylläpidettyä, on taattava, että päivitys on siirretty jokaiselle solmulle ja vasta sitten voidaan jatkaa ajoa.

Smed ja Hakonen määrittävät vasteen tarkoittamaan viivettä, jonka päivityksen saattaminen jokaisen solmun tietoon vie. Smedin ja Hakosen mukaan riittävän nopean vasteeseen pääsemiseksi on joskus jatkettava suoritusta ennen kuin kaikki osapuolet ovat saaneet päivityksen. Näin ollen hajautetulla tilalla voidaan olla hetkellisesti tilanteissa, että eri laitteilla olevat pelitilat ovat epäkonsistenssit. Hetkellinen epäkonsistenssi auttaa siihen, että vaste pienenee mutta siitä aiheutuu mahdollisesti tilan hallinnoimiseen liittyviä ongelmia. Tilan ollessa keskitetty, esimerkiksi palvelimelle, ongelman aiheuttaa liian hidas vaste. Vaste syntyy pääasiassa verkon viiveestä. Keskitetyllä tilalla ei kuitenkaan ole epäkonsistenssiongelmaa. Konsistenssi ja vaste eivät siten ole toisistaan riippumattomia tekijöitä. Tietokantatutkimuksessa vaste on perinteisesti aina alistettu konsistenssille, Smed ja Hakonen toteavat. He lisäävät, että verkkopeleissä reaaliaikaisen interaktion vuoksi vaste tulee kuitenkin tärkeämmäksi ja tämän vuoksi konsistenssista joudutaan välillä tinkimään.

Informaation periaatteen yhtälöä tarkasteltaessa on kaksi asiaa, jotka kompensatiotekniikoissa on huomioitava:

1. konsistenssin ja vasteen tasapaino, ja
2. skaalautuvuus.

Yhtälön kannalta konsistenssi ja vaste koskettavat ajallisuutta T ja sitä, miten sitä voidaan pienentää. Skaalautuvuus kohdistuu muuttujaan H ja siihen, miten se saadaan mahdollisimman pieneksi. Muuttuja H :han tarkoitti keskimääräistä vastaanottavien solmujen lukumäärä. Smed ja Hakonen kirjoittavat, että

todellisuudessa verkkoarkkitehtuuri ei voi saavuttaa sekä korkeaa konsistenssia että korkeaa vastetta yhtä aikaa. He jatkavat, että arkkitehtuurin valinta on perimmiltään kompromissi konsistenssin ja vasteen väliltä. Tätä kompromissien skaalaa on havainnollistanut Michael Zyda Networked Games - Design & Implementation Fall 2005 -kurssin kalvoissaan [Zyda, 2005b] ja se esitetään taulukossa 4.1.

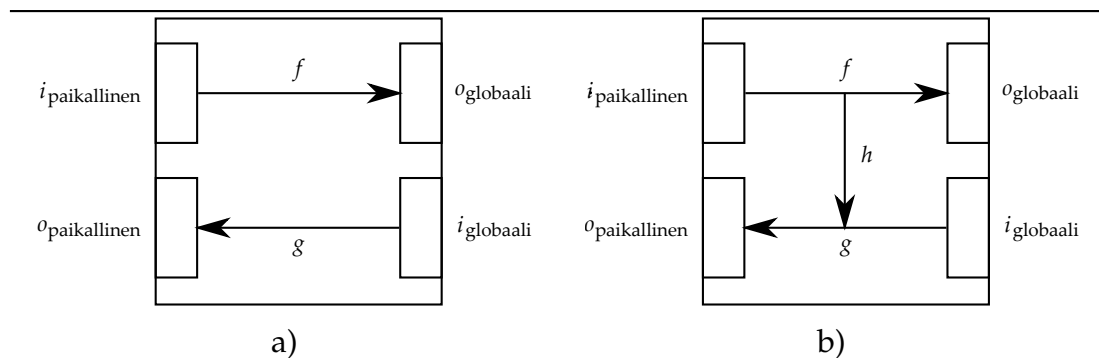
Taulukko 4.1: Konsistenssin ja vasteen kompromissien skaala

Järjestelmän ominaisuus	Absoluuttinen konsistenssi	Korkea vaste
Näkymän konsistenssi	Identtinen kaikilla solmuilla	Ratkaistaan saapuneen datan perusteella jokaisella solmulla
Dynaaminen datan tuki	Matala, konsistenssin hallintamekanismi ja protokolla rajoittavat	Korkea, ainoastaan saatava kaista rajoittaa
Verkko infrastruktuurin vaatimukset	Matala latenssi, korkea luotettavuus, rajoitettu vaihtelevuus	Heterogeeniset verkot mahdollisia
Osallistujien määrä	Pieni	Mahdollisesti suuri

Smed ja Hakonen valaisevat asiaa lisää erittelemällä kolme osaa data- ja kontrolliarkkitehtuureista. Näitä ovat paikallinen solmu, verkko ja linkki (*relay*), joka yhdistää solmun ja verkon. Solmun halutessa lähettää tietoja, se lähettää linkille kontrolliviestejä ja vastaanottaa linkiltä dataviestejä. Linkki kommunikoi muiden solmujen linkkien kanssa verkon välityksellä. Linkki on looginen käsite, joka selventää sitä, miten kontrolli vaikuttaa dataan. Linkki on siis solmun ja verkon välissä oleva välittäjä. Linkin rakenne määrittää sen, kuinka hyvä vaste ja konsistenssi arkkitehtuurilla voi olla. Kuvassa 4.2 a) on linkin yksinkertaisin muoto: kaksisuuntainen linkki (*two-way relay*). Siinä on merkitty paikallinen ja globaali syöte sekä tuloste. Tietovirta kulkee paikallisesta syötteestä globaaliin tulosteeseen ja myös globaalista syötteestä paikalliseen tulosteeseen. Kuvassa on merkitty *f*:llä ja *g*:llä ne operaatiot, joita linkki tekee tietovirroille. Operaatiot voivat olla esimerkiksi pakkaus ja purku tai salaus ja salauksen purku.

Kaksisuuntainen linkki on malli, jota yleisesti käytetään muun muassa ha-

jautettujen tietokantojen ja keskitettyjen järjestelmien yhteydessä. Kaksisuuntainen linkki mahdollistaa korkean konsistenssin saavuttamisen, koska siinä paikallisen solmun uudet viestit linkitetään verkkoon ja ne eivät ilmaannu edes lähettävälle paikalliselle solmulle ennen kuin linkki on saanut verkosta vastauksen. Verkosta tuleva vastaus voi tulla keskitetyltä palvelimelta tai vaikka muilta solmuilta. Joka tapauksessa vahvistus odotetaan ja vasta sitten jatketaan toimintaa. Tämä takaa korkean konsistenssin, mahdollisesti jopa absoluuttisen. [Smed and Hakonen, 2006].



Kuva 4.2: Linkkien rakenteet: a) Kaksisuuntainen linkki ja b) Oikosuljettu linkki

Tästä samaisesta asiasta kuitenkin johtuu myös se, että kaksisuuntainen linkki ei voi taata korkeaa vastetta. Takaus ei onnistu, koska linkki on riippuvainen saatavilla olevista verkkoresursseista. Jos vastetta halutaan parantaa, tulee linkki oikosulkea (*short-circuit relay*). Oikosulkeminen tarkoittaa, että linkin kaksi tietovirtaa yhdistetään. Kuvassa 4.2 b) yhdistäminen on kuvattu operaatiolla h . Operaatio f säilyy ennallaan, mutta $o_{paikallinen} = g(i_{gloobaali}) \times h(i_{paikallinen})$. Paikallisen solmun lähettämät viestit palautetaan siis välittömästi myös paikalliseen tulosteeseen linkissä. Linkin ei siis tarvitse enää odotella viestiä takaisin verkosta, vaan paikallinen solmu saa tiedon heti käyttöönsä. Tämä parantaa selvästi vastetta, mutta ongelmaton se ei ole. Ongelmia voi tulla siitä, että paikallisen solmun data voi nyt olla epäkonsistenssia muiden solmujen tai keskitetyn palvelimen dataan suhteutettuna. Epäkonsistenssin mahdollisuus pakottaa kehittämään jonkinlaisia palautus- tai neuvottelumekanismeja, joilla epäkonsistenssi voidaan ratkaista ja konsistenssi palauttaa. [Smed and Hakonen, 2006].

Korkean konsistenssin arkkitehtuuri vaatii siis kaksisuuntaisen linkin, ja korkean vasteen arkkitehtuuri puolestaan tarvitsee oikosuljetun linkin. Mitä

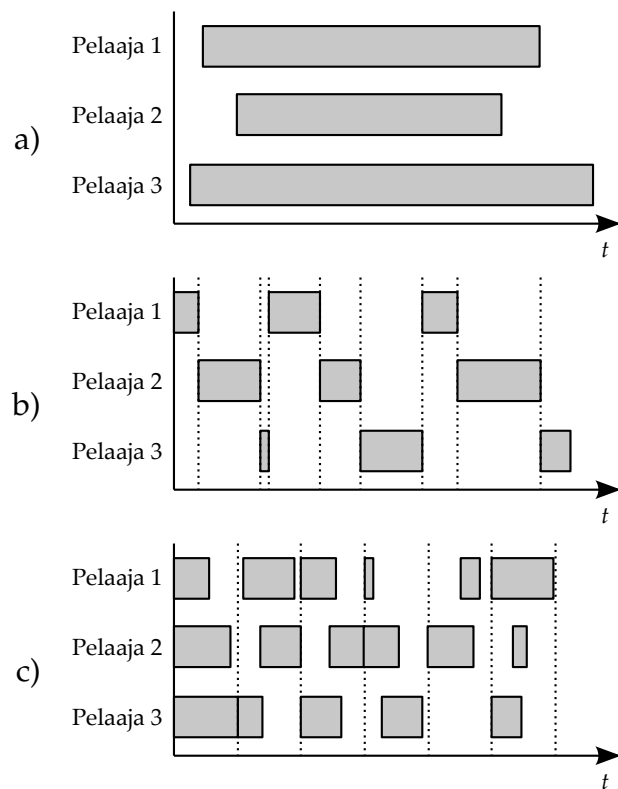
nämä kaksi arkkitehtuuria sitten tarkoittavat keskitetyn, hajautetun ja monistetussa verkkoarkkitehtuurien suhteen? Keskitetyssä arkkitehtuurissa linkin tehtävänä on pääasiassa siirtää paikallinen kontrolli verkkoon ja vastaanottaa dataa verkosta takaisin. Hajautetussa arkkitehtuurissa asia on käännetty toisinpäin. Monistetussa arkkitehtuurissa paikallinen syöte ja tuloste on sekoitus kontrolli- ja dataviestejä. Jokaisella arkkitehtuurilla on myös tyypilliset ongelmansa: keskitetyssä arkkitehtuurissa dataan pääsy voi olla hidasta, hajautetussa datan palasten hajautus täytyy hoitaa kunnolla, ja monistetussa datan päivittäminen jokaiseen toisintoon voi olla hankalaa. [Smed and Hakonen, 2006].

Skaalautuvuudella tarkoitetaan kykyä mukautua resurssien muuttumiseen. Tietokonepelien kohdalla tämä tarkoittaa muun muassa sitä, miten tehdä pelipalvelin niin, että se dynaamisesti mukautuu muuttuvaan pelaaja määrään. Skaalautuvuutta tarkasteltaessa ajaudutaan tarkastelemaan sarjallista ja rinnakkaista suorittamista. Verkko-ohjelman sarjallisesti suoritettavia osuuksia ei voida nopeuttaa rinnakkaisella laskennalla, mistä seuraa, että teoreettinen nopeutus S optimaalisesti rinnakkaistetulle järjestelmälle voidaan laskea *Am-dahlin* laista [Smed and Hakonen, 2006]:

$$S(n) = \frac{T_s + T_p}{T_s + T_p/n} = \frac{1}{\alpha + (1 - \alpha)/n} \leq \frac{1}{\alpha'}$$

missä T_s on sarjallinen aika, T_p rinnakkainen aika ja $T_s + T_p = 1$ sekä $\alpha = T_s / (T_s + T_p)$. Laista seuraa, että jos esimerkiksi viisi prosenttia ($\alpha = 0,05$) ohjelmasta pitää suorittaa sarjallisesti, saavutettava maksiminopeutus on tällöin 20. Ideaalitulanteessa sarjallista osuutta ei siis tulisi olla ollenkaan ja kaikki ohjelmakoodi olisi mahdollista laskea rinnakkain. Tällaisessa ideaalitulanteessa ei kuitenkaan ole enää mahdollisuutta solmujen yhteistyöhön. Ääritilanteina ovat siis tilanteet, joissa monen pelaajan pelissä kaikki pelaajat pelaavat riippumatta toisistaan ja toisessa ääripäässä tilanne, jossa pelissä ei ole minikäänlaista rinnakkaisuutta. Rinnakkaisuuden puuttuminen, eli täysin sarjallinen peli, ei ole pelimaailmassa vieras moninpelin muoto. Esimerkiksi lauta- ja korttipelit sekä niiden sopeutukset digitaalisiksi peleiksi ovat nimittäin vuoropohjaisia. Ääripäiden välissä ovat pelit, jotka sisältävät reaaliaikaista interaktiota pelaajien välillä. Tämän vuoksi ne sisältävät sekä rinnakkaista että sarjallista laskentaa. Reaaliaikaisuutta ja vuoropohjaisuutta on havainnollistettu kuvassa 4.3 [Smed and Hakonen, 2006].

Kuvassa 4.3 a) toisistaan erilliset reaaliaikaiset pelit toimivat rinnakkaisesti mutta ilman interaktiota pelien välillä. Saman kuvan kohdassa b) vuoropoh-



Kuva 4.3: Rinnakkaista ja sarjallista suoritusta tietokonepeleissä: a) Toisistaan erilliset reaaliaikaiset pelit, b) Vuoropohjaiset pelit ja c) Interaktiiviset reaaliaikaiset pelit

jaiset pelit ovat sarjallistettuja ja interaktiivisia mutta eivät reaaliaikaisia. Ku-
vassa 4.3 c) interaktiivinen reaaliaikainen peli sisältää sekä sarjallista että rin-
nokkaista suoritusta.

Solmujen on sovittava tapahtumien järjestys sarjallisille osuuksille. Hel-
poinn se saavutetaan käyttämällä asiakaspalvelinarkkitehtuuria. Palvelin
pystyy hallinnoimaan kommunikointia välittämällä, suodattamalla ja muok-
kaamalla viestejä. Smed ja Hakonen huomauttavat, että vertaisverkkoarkki-
tehtuurissa verkko itsessään toimii kuten palvelin. Solmut jakavat saman sar-
jallisen kommunikointikanavan. Poikkeuksena on tilanne, jossa solmut on yh-
distetty toisiinsa suoraan fyysisellä kaapelilla tai solmut käyttävät hyväkseen
joukkolevitystä (multicast). Sarjalliset osuudet vaativat siis kommunikointia.
Tällöin skaalautuvuutta rajoittaa valitun verkkoarkkitehtuurin kommunikoin-
tikapasiteetti. Asiakaspalvelinarkkitehtuurille käyttäen täsmälähetystä (uni-
cast) seuraava ehto on voimassa:

$$d \cdot f \cdot n \leq C,$$

missä d on viestin bittien määrä, f on lähetysfrekvenssi, n on täsmälähetyssyhteyksien lukumäärä ja C on kommunikointi kanavan maksimikapasiteetti. Smed ja Hakonen havainnollistavat tilannetta seuraavilla arvoilla 10 Mbps ethernet verkossa: $f = 5$, $C = 10^7$ ja $d = 752 + 32 = 784$. Ensimmäinen luku kuvaa IPv6-protokollan kehyksen kokoa ($68 \cdot 8 + 26 \cdot 8 = 752$ bittiä eli 94 tavua). Jälkimmäinen luku on varattu varsinaiselle datalle, eli tässä tapauksessa tarkoituksena on lähettää yksi 32-bittinen kokonaisluku. Näillä arvoilla laskettuna asiakaspalvelinarkkitehtuurin palvelin voi tarjota sarjallistusta korkeintaan 2551 asiakkaalle.

Edellisessä laskuesimerkissä on valittu melko pienet arvot f :lle ja d :lle. Todellisissa tilanteissa kumpainenkin olisi todennäköisesti suurempi. Täten yli 2500 asiakkaan palveleminen on ylioptimistista, koska kommunikointi kuluttaa lisäksi laskentatehoa palvelimelta. Asynkronisessa viestinnässä asiakkaat saavat lähettää viestejä milloin haluavat. Tällöin pahin tilanne on se, kun kaikki asiakkaat yrittävät kommunikoida yhtä aikaa. Verkkoarkkitehtuurin tulee selviytyä myös tästä tilanteesta. [Smed and Hakonen, 2006]. Matkapuhelimille sovellettuna huomioon täytyy ottaa kommunikointikanavan kapasiteetti asiakkaan päässä sekä lähetysten frekvenssi. Kapasiteetti rajoittaa pakettien kokoa ja frekvenssiä pienentää viive. Periaatteessa palvelimen verkkoyhteyden kais-tanleveys ei ole rajoittunut samaan tapaan kuin asiakkaan, koska palvelin ei toimi matkapuhelinverkkojen varassa (katso kuva 3.2).

Taulukossa 4.2 on listattuna kommunikointikapasiteetin vaatimusten suuruusluokat eri verkkoarkkitehtuureille Smedin ja Hakosen mukaan. Yksittäisen solmun kapasiteettitarve on nolla, sillä solmun ei tarvitse kommunikoida. Yksittäinen solmu on harhaanjohtava nimitys, sillä solmua ei esiinny ilman verkkoa. Itse nimittäisin tilannetta yksittäiseksi suorittajaksi, sillä asiakaspalvelinarkkitehtuurissa voi esiintyä tilanne, jossa palvelin palvelee vain yhtä asiakasta. Tällöin kapasiteettivaatimus ei enää ole nolla. Vertaisverkon tilanteessa solmun kommunikoinnin resurssitarve on $O(n)$, kun kaikkien solmujen välillä on suora yhteys tai kun kommunikointiin käytetään joukkolevitystä. Resurssitarve $O(n^2)$ saavutetaan vertaisverkossa silloin, kun solmut kommunikoivat täsmälähetystä käyttäen. Asiakaspalvelin-arkkitehtuurin palvelinpään kapasiteettitarve on $O(n)$, koska jokaisella solmulla on yhteys palvelimeen. Palvelinverkkoarkkitehtuureissa palvelimia on m kappaletta ja asiak-

kaat ovat jakautuneet tasaisesti palvelimille. Jos palvelimet ovat vertaisverkossa keskenään, niiden kommunikointitarve keskenään on $O(m) \dots O(m^2)$ ja lisäksi tulee asiakkaiden kommunikointi, joka on $O(m/n)$. Hierarkisessa palvelinverkossa palvelimet on järjestetty esimerkiksi puu-muotoon. Tällöin juuresa oleva palvelin on pullonkaula vaatien $O(n)$ kapasiteetin. [Smed and Hakonen, 2006].

Taulukko 4.2: Kommunikointikapasiteettivaatimukset eri verkkoarkkitehtuurille solmujen määrällä n ja palvelimien määrällä m

Verkkoarkkitehtuuri	Kapasiteettivaatimus
Yksittäinen solmu	0
Vertaisverkko	$O(n) \dots O(n^2)$
Asiakas-palvelin	$O(n)$
P2P palvelinverkko	$O(n/m + m) \dots O(n/m + m^2)$
Hierarkinen palvelinverkko	$O(n)$

Smedin ja Hakosen mielestä edellinen laskuharjoitus ja sen tulos 2551 tuetua asiakasta osoittaa, että käytännössä lineaarinen kapasiteettivaatimus on liian iso. Tämän vuoksi skaalautuvuuden ydin on saavuttaa kommunikointi, jonka kapasiteettivaatimus ei nouse edes lineaarisesti. Sublineaarinen kapasiteettivaatimus tarkoittaa käytännössä sitä, että asiakas ei voi olla koko ajan tietoinen kaikista muista asiakkaista ja että palvelimien välistä kommunikointia joudutaan rajoittamaan. Sublineaarisen kommunikaation saavuttaminen palvelinverkossa tarkoittaa sitä, että palvelimien välistä kommunikointia tulee rajoittaa. Jos palvelinverkko on hierarkinen k -puu, ja palvelin lähettää sen vanhemmalle korkeintaan $1/k$ osan sen lapsien viesteistä, kapasiteettivaatimus on logaritminen. Juuripalvelimen kommunikointi on silloin $O(\log(n))$. Ongelmaksi muodostuu täten se, miten toteuttaa viestien vähennys seuraaville tasoille. Tähän ongelmaan kompensointitekniikat tarjoavat vastauksen: lapsen viestejä voidaan pakata ja koota yhteen, jos ne takaavat $1/k$ vähennyksen joka tasolla. [Smed and Hakonen, 2006].

Näin suurta vähennystä Smed ja Hakonen pitävät melko epätodennäköisenä. Heidän mielestään käyttökelpoisempi ratkaisu on ensin käyttää kiinnostuksen alueen hallintaa (*interest management*) ja sitten valita yksi viesti, jonka palvelin välittää eteenpäin. Solmut voivat arvioida dataa esimerkiksi käyttämällä *liikkeen ennustamista* kaikille pidätetyille viesteille. Muita mahdollisia

Smedin ja Hakosen listaamia kompensointitekniikoita ovat *kiinnostuksen alueen mukainen suodatus*, *protokollan optimointi*, *paikallinen havainnointisuodatus* ja *synkronoitu simulaatio*. Tarkastelen tekniikoita seuraavissa alaluvuissa hieman tarkemmin.

4.2 Resurssirajoitusten kompensointitekniikat

Resurssirajoitusten kompensointitekniikoita on lukuisia. Tekniikat voidaan luokitella esimerkiksi Jiung-yao Huangin [2006] esittämällä tavalla. Hän luokittelee tekniikat seuraavasti: protokollan optimointi, datavirran ja datan näkyvyyden hallinta, ihmisen hahmottamiskyvyn rajoitusten hyödyntäminen ja järjestelmän arkkitehtuurin parantaminen. Hänen mukaansa voimme valita, mitä informaation periaatteen yhtälön muuttujaa haluamme pienentää. Muuttujan valinta riippuu siitä, minkälainen kyseinen peli on, jossa tekniikkaa käytetään ja myöskin siitä mitkä resurssit ovat pahiten pullonkaulana.

Huangin karkean luokittelun mukaan käyn seuraavaksi läpi protokollan optimoinnin. Datavirran ja datan näkyvyyden hallinnasta esimerkkinä on kiinnostuksen alueen mukainen suodatus. Ihmisen hahmottamiskyvyn rajoituksia hyödynnetään paikallisessa havainnointisuodatuksessa. Samaan luokkaan voitaisiin laskea kuuluvaksi myös liikkeen ennustaminen. Toisaalta Huang mainitsee, että liikkeen ennustamisen voidaan nähdä kuuluvan myös protokollan optimointiin. Tarkemmin sanottuna viestien ulkoiseen tiivistämiseen, koska siinä yleensä lähetetään delta- eli muutostietoja. Viimeisenä resurssirajoitusten kompensointitekniikkana esittelen synkronoidun simulaation, joka kuuluu Huangin luokittelussa järjestelmän arkkitehtuuriin. Muita tähän luokkaan kuuluvia tekniikoita käsittelemme jo kohdassa 3.4, jossa kävin läpi mobiilipelien verkkoarkkitehtuureja.

4.2.1 Protokollan optimointi

Protokollan optimoinnilla (*protocol optimization*) tavoitellaan Smedin ja Hakosen mukaan resurssien säästämistä kahdella tavalla. Ensimmäinen vaihtoehto on pienentää lähetettävien viestien kokoa ja toinen tapa on vähentää viestien määrää. Viestien koon pienentäminen toteutetaan pakkaamalla eli tiivistämällä tietoa. Pakkaustekniikoita on kahden tyyppisiä: häviöttömiä (*lossless*) ja häviöllisiä (*lossy*). Häviötön pakkaustapa tiivistää tiedon jollakin valitulla algoritmilla ja vastaanottavassa päässä purkaa viestin. Purettu viesti on täsmälleen

sama kuin viesti ennen tiivistämistä. Häviötön pakkaus ei kuitenkaan yleensä saavuta niin suurta pakkaussuhdetta kuin häviöllinen pakkaus. Häviöllisessä pakkaustavassa nimensä mukaisesti hävitetään ja hukataan tietoa. Tietoa ei kuitenkaan hävitetä satunnaisesti vaan se yritetään kohdistaa paikkoihin, jotka ennallistamisen jälkeen jäävät ihmiseltä huomaamatta [Smed and Hakonen, 2006]. Häviöllisessä pakkauksessa käytetään yleensä hyväksi ihmisen aistien tiedettyjä rajoituksia ja täten kohdistetaan hukkaaminen alueille, jotka aisteilta jäävät havaitsematta. Näin toimii esimerkiksi suosittu häviöllinen musiikin ja äänen pakkausmenetelmä MP3 sekä kuvien tiedostokoon pienentämisessä auttava JPEG-pakkaus.

Smed ja Hakonen kertovat pakkaustekniikoiden olevan joko sisäisiä tai ulkoisia. Niiden ero on siinä, että kun tarkastellaan kommunikointia sarjana viestejä, sisäinen pakkaus keskittyy tiivistämään informaatiota pelkästään yhdessä viestissä. Sisäinen pakkaustapa ei siis ota mitenkään huomioon aikaisemmin lähetettyjä viestejä. Tämän takia Smed ja Hakonen sanovat sen soveltuvan hyvin epäluotettavien protokollien, kuten UDP:n käytön yhteyteen. Ulkoinen pakkaustapa ottaa siis huomioon jo aiemmin lähetetyt paketit ja tiedot.

Smed ja Hakonen antavat esimerkin, jossa ulkoisella pakkauksella ei lähetetäkään objektin tarkkaa sijaintitietoa vaan esimerkiksi delta, eli edelliseen tilanteeseen verrattu sijainnin muutos. Tämän pitäisi sisältää vähemmän bittejä kuin joka kerta lähetettävä absoluuttinen sijaintitieto. Ulkoisessa pakkauksessa voidaan antaa myös viitteitä edellisiin viesteihin, jos sama viesti toistuu useammin. Smedin ja Hakosen mukaan ulkoinen pakkaus voi huomioida suuren määrän dataa kerralla ja siten havainnoida päällekkäisyyksiä ja toistuvuutta tiedoissa sekä viestisarjassa. Täten ulkoinen pakkaus voi saavuttaa paremman pakkaussuhteen kuin sisäinen pakkaus. Smed ja Hakonen lisäävät, että koska ulkoinen pakkaus voi viitata aikaisempiin tietoihin, se vaatii luotettavaa viestinvälitysprotokollaa, kuten TCP. Taulukossa 4.3 on tiivistettyinä pakkaustekniikkakategoriat Smedin ja Hakosen mukaan.

Viestien määrän vähentäminen toteutetaan yhdistämällä usean viestin tiedot samaan viestiin. Huangin [2006] mukaan viestien yhdistämisellä voidaan saavuttaa jopa 50 prosentin vähennys kaistan tarpeessa. Smedin ja Hakosen mukaan on kaksi tapaa päättää, koska viestejä on yhdistetty riittävästi. Ensimmäinen kriteeri käyttää aikakatkaisua (*timeout-based*), jossa kaikki tietyn ajanjakson aikana tuotetut paketit yhdistetään yhteen lähetettävään pakettiin. Lähestymistapa takaa ylärajan viiveelle, jonka yhdistämisen eli aggregoinnin aikakatkaisun odottaminen aiheuttaa. Tässä lähestymistavassa kaistan säästö

Taulukko 4.3: Pakkaustekniikkakategoriat

	Häviötön	Häviöllinen
Sisäinen	Koodaa viestin tehokkaampaan muotoon ja poistaa päällekkäiset tiedot	Suodattaa epäolennaisen tiedon tai vähentää tiedon yksityiskohtaisuutta yhdessä viestissä
Ulkoinen	Välttää lähettämästä tietoa, joka on identtinen jo aikaisemmin lähetetyn tiedon kanssa	Välttää lähettämästä tietoa, joka on liian samankaltainen jo lähetetyn tiedon kanssa.

määräytyy siitä, mikä on viestien synnyttämisen nopeus ja kuinka monta viestiä yhdistetään aikarajan puitteissa yhteen lähetettävään pakettiin. Pahimmillaan säästöä ei synny, jos ajanjakson sisällä ei synnytetä yhtään tai korkeintaan yksi viesti. Tällöin viestien aggregointi lisää vain viivettä mutta ei vähennä kaistaa.

Toinen kriteeri on käyttää kiintiöpohjaista ratkaisua (*quorum-based*). Siinä ennalta sovittu määrä viestejä yhdistetään aina yhteen. Smed ja Hakonen toteavat, että koska tässä ratkaisussa odotetaan aina tietty määrä syntyviä viestejä yhdistettävään viestiin, viiveelle ei ole mitään takuita. Kiintiöperustainen ratkaisu takaa kuitenkin kaistan säästön ennustettavuuden mutta peli voi kärsiä pitkistä viiveistä, kun järjestelmä odottaa riittävää määrää paketteja synnytyksi ennen niiden yhdistämistä ja aggregaatin lähettämistä. Smedin ja Hakosen mielestä näiden kahden tavan rajoituksia voidaan kiertää kehittämällä niistä hybridiratkaisu, jossa kumpikin kriteeri yhdistetään. Tässä hybridissä aggregaattiviesti koostetaan ja lähetetään aina, kun jompi kumpi ehdoista täyttyy. Joko tapahtuu aikakatkaisu tai sitten aikarajan sisällä on syntynyt riittävä määrä viestejä ja aggregaatti muodostetaan ja lähetetään jo ennen aikakatkaisua.

Viestien pakkaaminen kohdistuu yhtälössä 4.1 muuttujaan B eli keskimääräistä pakettien kokoa yritetään pienentää. Tätä ei kuitenkaan saavuteta ilman pakkaamisen ja purkamisen vaatimaa laskentaa, joten muuttuja P kasvaa. Viestien yhdistämisessä viestien määrä M pienenee mutta samalla pienenee myös ajallisuus T . Vaikka keskimääräinen pakettien koko B kasvaakin, niin kokonaiskaistan tarve pienenee mutta prosessointi tehoa P tarvitaan lisää.

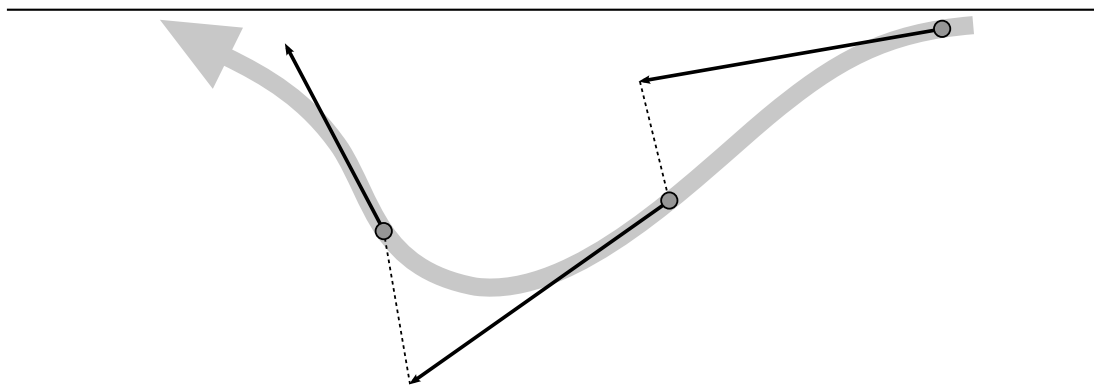
Protokollan optimointi kaistan vähentämiseksi on toisilla sovellusalueilla tavoiteltavaa. Suurin ongelma verkkomobiilipelien kohdalla ei kuitenkaan ole kaistan riittävyys vaan viive. Samankaltaisia ongelmia kohtaavat Internetissä soitetut puhelut. Ongelmaan on ratkaisuksi kehitetty menetelmä, jossa pakettiin sisällytetäänkin enemmän tietoa kuin on tarpeellista. Lisätieto auttaa pienentämään latenssia tapauksissa, joissa paketteja hukkuu. Näin kaistaa voidaan käyttää jossain määrin hyväkseen vähentämään latenssia. [Evensen *et al.*, 2008]. Tämä ratkaisu ei todennäköisesti kuitenkaan ole käyttökelpoinen mobiilipelien kohdalla, koska se perustuu TCP-protokollaan, jota ei yleisesti suositella verkkopeleille.

4.2.2 Liikkeen ennustaminen

Liikkeen ennustamistekniikat (*dead reckoning*) saivat alkunsa merenkulusta. Laivojen kulkua haluttiin seurata, ja kun ei ollut mitään varmaa tapaa määrittää sijaintia, päädyttiin käyttämään liikkeen estimointia. Siinä aluksen nykyinen sijainti selvitetään tiedetyn lähtösijainnin, aluksen nopeuden ja kuluneen ajan avulla. Verkkopeleissä liikkeen ennustamista käytetään hyväksi, jotta kaistan tarvetta saataisiin pienemmäksi lähetettävien pakettien frekvenssiä harventamalla. Harventuneiden päivitystietojen välissä objektien paikkoja lasketaan liikkeen ennustamistekniikoilla. Yksinkertaisimmillaan edellisiä tietoja käytetään hyväksi ja niistä ekstrapoloidaan nykyinen tilanne.

Kuvassa 4.4 on esitetty kappaleen todellinen reitti harmaalla nuolella ja ennustettu reitti mustilla nuolilla sekä päivityspakettien sisältämät sijaintitiedot, jotka näkyvät harmaina ympyröinä. Liikkeen ennustamisen menetelmä koostuu ennustamis- ja konvergenssiosuuksista. Konvergenssia tarvitaan, jos ja kun ennustettu tilanne ja päivityksessä saatu tieto ovat liian kaukana toisistaan. Tällöin tilojen ero tulee yhdistää eli konvergoida tuoreempaan tietoon. Kuvassa 4.4 katkonainen viiva edustaa kappaleen paikan vaihdosta uuden päivitystiedon saapuessa. Tämä ennustetun tiedon ja uuden todellisen tiedon välinen siirtymä käsitellään konvergenssialgoritmilla [Smed and Hakonen, 2006]. Tarkastelen kuitenkin ensin ennustamista.

Smed ja Hakonen sanovat, että yleisin ennustamistekniikka on käyttää derivoituvia polynomeja (derivative polynomial). Jos tilainformaatio esittää paikkaa p , tällöin kaksi ensimmäistä derivaattaa saa luonnolliset tulkinnat nopeutena v ja kiihtyvyytenä a . Nollas derivaatta sisältää vain paikan tiedon, johon ei ole sisällytetty mitään ennustamiseen liittyvää tietoa. Ensimmäisen derivaatan



Kuva 4.4: Oikea liike ja ennustettu liike: ennustettu liike (mustat nuolet) eroavat oikeasta liikkeestä (harmaa nuoli)

tapauksessa välitetään lisäksi nopeus:

$$p(t) = p(0) + v(0)t$$

Ennustuksen tarkkuutta pystytään nostamaan lisäämällä kiihtyvyysetieto päivityksiin. Tällöin kyseessä on toinen derivaatta:

$$p(t) = p(0) + v(0)t + \frac{1}{2}a(0)t^2$$

Smedin ja Hakosen mielestä tällainen toinen derivaatta mahdollistaa aika tarkan kulkuneuvon liikkeen mallintamisen. Heidän mielestään ensimmäinen derivaatta sopii kuitenkin paremmin vähemmän säännöllisesti liikkuvien objektien, kuten ihmishahmojen, liikkeen ennustamiseen. Syy tälle on siinä, että korkeamman asteen polynomit ovat herkempiä virheille, koska derivaatan arvojen tulee olla silloin tarkemmat. Tätä korkeamman asteen polynomeja ei kannata myöskään käyttää koska silloin joudutaan lähettämään enemmän tietoja, mikä puolestaan kuluttaa kaistaa. Jokainen lisätermi lisää myös laskentaa muutamilla operaatioilla.

Derivoituvat polynomit voidaan myös unohtaa ja käyttää pelkästään kappaleen historiatietoja, joista ekstrapoloidaan ennuste. Smed ja Hakonen kuvailevat tällaisen paikan historiatietoihin perustuvan liikkeen ennustamistekniikan lähettävän vain kappaleen absoluuttiset sijaintitiedot. Objektin nopeus ja kiihtyvyys arvioidaan historiasta. Tämä metodi voi halutessaan valita dynaamisesti käyttääkö se ensimmäisen vai toisen asteen polynomeja. Valintansa se tekee kolmesta uusimmasta päivityksestä muodostetun ennusteen perusteella. Jos näistä tiedoista päätellen kiihtyvyys on pientä tai suurta, käytetään en-

simmäisen asteen polynomeja. Muissa tapauksissa käytetään toisen asteen polynomeja. Tällä pyritään pienentämään kiihtyvyystermistä johtuvia epätarkkuuksia.

Smed ja Hakonen kertovat lisäksi, että ennustaminen voi ottaa huomioon paremmin kyseessä olevan kappaleen. Derivoituvat polynomit eivät ota millään tapaa huomioon sitä, mitä objekti on juuri nyt tekemässä, mitä objekti pystyy tekemään tai kuka kontrolloi objektia. Smed ja Hakonen antavat esimerkiksi autot ja lentokoneet, jotka noudattavat samoja fysiikan lakeja. Esimerkeille kuitenkin voisi olla varsin erilaiset ennustusalgoritmit, jotka ottaisivat huomioon niiden erilaiset käyttäytymiset. Lopuksi Smed ja Hakonen muistuttavat, että tällainen mallintaminen voi kuitenkin lisätä tarvittavaa laskentaa huomattavasti ja eri käyttäytymisalgoritmien ylläpito sekä kehittäminen vievät resursseja lisää. Pantel ja Wolf [2002b] ovat tosin osoittaneet, että eri pelityypeille kannattaa suunnitella erilaisia ennustamisalgoritmeja. Heidän mitaustensa mukaan kaikkein monimutkaisimmat ennustamismenetelmät eivät antaneet parhaita tuloksia.

Aggarwal *et al.* [2004] kehottavat käyttämään yhteistä synkronoitua aikaa solmuille sekä lisäämään aikaleiman jokaiseen pakettiin, joka sisältää paikkatiedon päivityksiä. Tällä yhteisen synkronoidun ajan käyttämisellä ja aikatie-doilla paketissa päästään ennustuksissa ja varsinkin objektien piirtämisessä suurempaan tarkkuuteen. Aggarwal *et al.* [2004] mukaan heidän menetelmälään saadaan parannusta tarkkuuteen aina 100 ms:n viiveistä lähtien.

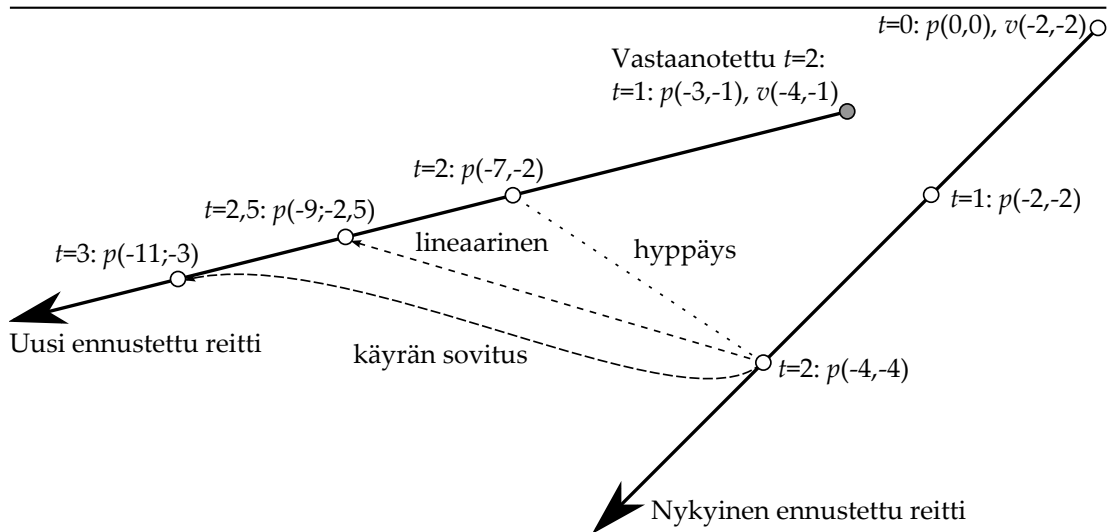
Koska solmut tietävät, mitä ennustusmenetelmää vastaanottava puoli käyttää, lähettämässä voidaan käyttää hyväksi tätä tietoa. Tällä voidaan pienentää lähetettävien pakettien lähetystiheyttä eli lähettää paketteja harvemmin. Smed ja Hakonen kertovat tällaisen tilanteen olevan mahdollista määrittämällä virhekynnys tai -raja. Lähettävä solmu voi käyttää samaa ennustamisalgoritmia ja laskea koska todellinen tilanne ja ennustus ovat virhekynnystä suuremmat. Vasta tällöin lähetetään seuraava päivitys, joka korjaa tilanteen. Smed ja Hakonen mainitsevat menetelmiä, joissa lähettävä solmu voi dynaamisesti säätää virherajaa käytettävissä olevien verkkoresurssiensa puitteissa. Tällöin kaistan tarpeen ollessa suuri, virhekynnystä nostetaan riippuen objektien välisistä etäisyyksistä tai päivityspaketin voimassaoloaikaa kasvatetaan.

Miten käsitellään tilanne, jossa todellinen tilanne ja ennustus ovat huomattavan kaukana toisistaan? Solmu, joka käyttää liikkeen ennustamista, kohtaa tämän tilanteen lähes jokaisen päivityksen saapuessa. Ennustetun ja todellisen tilan välistä siirtymää ohjaamaan on kehitetty konvergenssitekniikoita. Sme-

din ja Hakosen mielestä yksinkertaisin konvergenssitekniikka on niin sanottu nolla-tason konvergenssi. Tässä tekniikassa siirtymää ei yritetä tehdä tasaisesti, vaan objektin paikkatieto päivitetään kertarysäyksellä uuteen tietoon. Yhtäkkäinen siirtymä voi kuitenkin näyttää pelaajan mielestä epämiellyttävältä, kun esimerkiksi vastapelaaja hypähtää odottamatta uuteen sijaintiin. Joissakin tilanteissa tällainen käyttäytyminen voi näyttää jopa mahdottomalta, kun jokin objekti näyttää siirtyvän suoraan esimerkiksi seinän toiselta puolelta toiselle. Monien verkkopelien yhteydessä tästä ongelmasta puhutaan warppauksena (*warpping*). Warppauksen ainoa syy ei ole konvergenssi. Liikkeen ennustaminenkin voi aiheuttaa sitä, jos esimerkiksi liikerata ennustetaan kulkemaan seinän läpi. Tähän auttaisi liikeradan ennustamisessa käytettävä törmäyslaskenta. Smedin ja Hakosen mielestä hyvä konvergointimenetelmä pyrkii oikaisemaan ennusteen todelliseen tilanteeseen nopeasti mutta mahdollisimman huomaamattomasti pelaajalle.

Konvergointi toteutetaan valitsemalla aika, jonka puitteissa virhe oikaistaan. Objektin ennustetun liikeradan ja todellisen liikeradan välille lasketaan siirtymä. Siirtymä voi olla jo edellä mainittu yhtäkkäinen pomppaus radalta toiselle mutta se voi olla myös tasaisempi siirtymä. Kuvassa 4.5 on esitetty nykyinen ennustettu reitti ja päivituksen jälkeen laskettu uusi ennustus. Kuvassa näkyy myös nollatason konvergenssi, joka on kuvattu harvalla katkoviivalla. Linearisessa siirtymässä lasketaan piste uudelta ennustetulta kulkureitiltä, jossa objekti on valitun konvergenssiajan päästä. Tämän jälkeen lasketaan suora siirtymä nykyisestä pisteestä konvergenssipisteeseen. Virheen oikaisuun varatun ajan jälkeen objekti on konvergenssipisteessä, eli oikeassa positiossaan ja jatkaa liikettään uuden ennustetun kulkuradan mukaisesti. Tämä lineaarinen siirtymä on myös esitetty kuvassa 4.5. Uutta ennustettua kulkurataa seurataan niin pitkään, että uusi päivitys saapuu ja konvergointiprosessi suoritetaan uudestaan. Lisäksi Smed ja Hakonen kertovat monimutkaisempiakin konvergointitapoja. He sanovat, että vaikka lineaarinen tapa on huomattavasti parempi kuin yhtäkkäinen, voi lineaarinen konvergointi silti näyttää pelaajan mielestä epäluonnolliselta liikkeeltä. Erityisesti ongelmia voi esiintyä erityisesti konvergoinnin aloitus- ja lopetuspisteissä, joissa objektin pyöriminen kulkureiteille voi olla luonnottoman näköistä.

Konvergoinnin liikerata vanhan ja uuden ennustetun reitin välissä voidaan tehdä sujuvammaksi käyttämällä käyrän sovitusta (*curve-fitting*). Smedin ja Hakosen mukaan konvergoinnin aloitus- ja lopetuspisteiden lisäksi valitaan kaksi muuta pistettä. Toinen piste valitaan vanhalta ennustetulta liikeradalta



Kuva 4.5: Konvergointitapoja

eli objekti on jo kulkenut tämän pisteen kautta. Viimeinen piste valitaan uudelta ennustetulta liikeradalta konvergointipisteen jälkeen. Käyrä sovitetaan kulkemaan näiden neljän pisteen kautta käyttäen esimerkiksi kolmannen asteen kuutiollista käyrää. Smed ja Hakonen toteavat, että näin konvergoinnin liikeradasta tulee sulava ja siirtymä vanhalta liikeradalta uudelle on jouheva ilman jyrkkiä käännöksiä. He jatkavat, että vaikka siirtymä onkin nyt huomaamattomampi, vaatii sen toteuttaminen kuitenkin enemmän laskentaa. Tämä tilanne on myös hahmoteltu kuvaan 4.5.

Informaation periaatteen yhtälöstä (4.1) liikkeen ennustaminen vähentää viestien määrää M ja ajallisuutta T . Tätä kompensoidakseen solmujen on kuitenkin laskettava liikkeen ennustuksia, jotka kasvattavat P :tä. Mahdollinen konvergenssitilanne vaatii entisestään lisää laskentaa. Mitä hienommalla ja monimutkaisemmalla algoritmilla siirtymäpolku lasketaan, sitä enemmän se vaatii laskentatehoa.

4.2.3 Paikallinen havainnointisuodatus

Liikkeen ennustamisen ja erityisesti konvergenssialgoritmien yhteydessä kirjoitin siitä, miten konvergointi tulisi toteuttaa mahdollisimman huomaamattomasti pelaajaa ajatellen. Tätä pelaajan havainnointikykyä käytetään hyväksi myös resurssien kompensointitekniikassa, jonka nimi on paikallinen havainnointisuodatus (*local perception filters*). Smed ja Hakonen kirjoittavat, että pai-

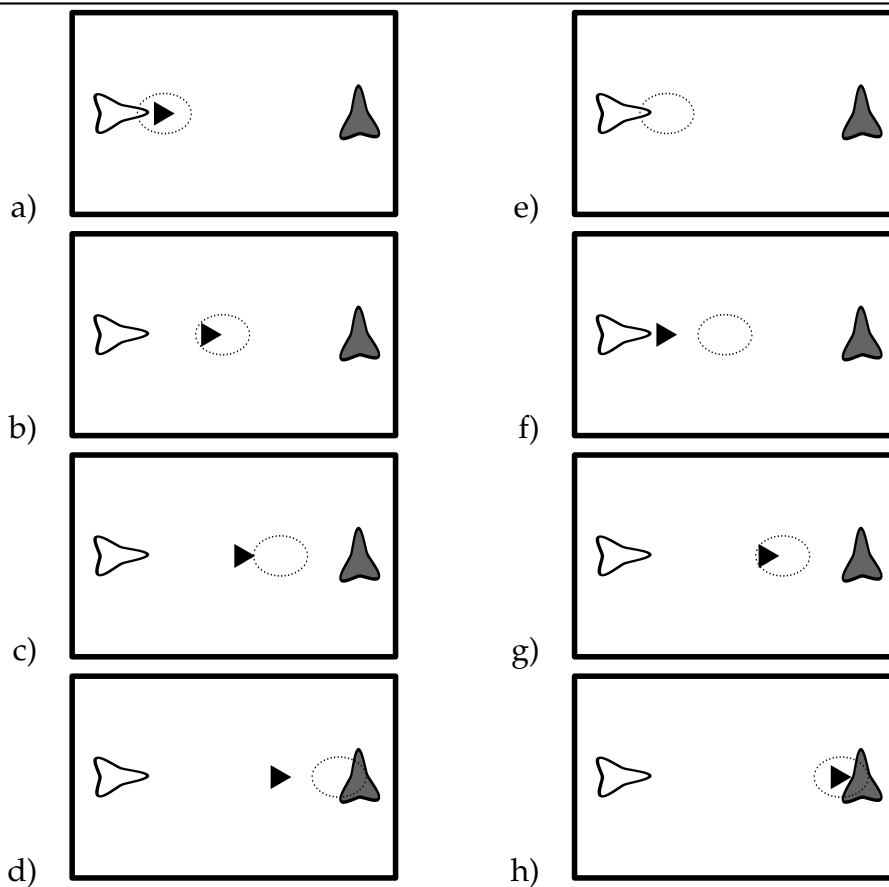
kallinen havainnointisuodatus kätkee viiveen käyttämällä hyväkseen ihmisen hahmottamisen rajoituksia. Liikkeen ennustamisen tekniikassa pyrittiin pitämään konsistenssinäkymä yllä ennustamalla liikettä. Paikallisen havainnoinnin suodatuksessa näkymän ajalliset vääristymät sallitaan, piirtämällä objektit hieman vääriin paikkoihin. Vähän väärään paikkaan piirtäminen ei kuitenkaan ole mitenkään satunnaista vaan se tehdään tarkasti solmujen välisen viiveen pohjalta. Vääristymien ei tässäkään tekniikassa kuitenkaan haluta näkyvän pelaajille, vaan se yritetään kätkeä pelaajan havainnointirajoitusten puitteisiin. Smed ja Hakonen korostavat, että vaikka he määrittelevät paikallisen havainnoinnin suodatuksen visuaalisuuden kautta, sen kantava idea on yleisempi ja täten sovellettavissa muuhunkin käyttöön.

Tässäkin tekniikassa, niin kuin synkronoidussa simulaatiossa (4.2.5) Smed ja Hakonen tuovat esiin deterministiset ja epädeterministiset objektit ja tapahtumat. Smedin ja Hakosen mukaan deterministisiä objekteja tai tapahtumia ovat passiiviset objektit tai objektit, jotka ovat järjestelmän hallinnassa. Determinististen objektien käyttäytyminen noudattaa jotain mallinnettua kaavaa tai lakia, esimerkiksi fysiikan lakeja. Ne ovat täten ennustettavia käyttäytymiseltään. Epädeterministiset objektit liittyvät usein ihmisiin. Ihmispelaajan tekemiset eivät ole ennustettavissa ja tekemiset ovat arvaamattomia. Smed ja Hakonen jaottelevat pelaajat paikallisiin ja etäpelaajiin viiveen perusteella.

Paikallinen havainnointisuodatus kohtaa viiveen aiheuttamat ongelmat erottamalla todellisen tilan pelaajalle näytetystä tilasta. Pelaajalle piirretyn näkymän ei tarvitse näyttää todellista tilannetta vaan se voi sisältää jotain vanhentunutta tietoa. Smed ja Hakonen kertovat, että tämän ajallisen vääristymän ero on helppoa määrittää pelaajille. Paikalliset pelaajat piirretään ajantasaisista tietoa käyttämällä ja verkon kautta yhteydessä olevat etäpelaajat piirretään käyttäen d sekuntia vanhaa tietoa. Viive määrittää, minkä arvon d saa. Syyseuraus -suhteen eli kausaliteetin säilyttäminen on pelaajille tärkeää. Tämän vuoksi ajallinen vääristymä passiivisille objekteille vaihtuu dynaamisesti. Mitä lähempänä passiivinen objekti on paikallista pelaajaa, sitä lähemmäksi oikeaa tilaa se tulee piirtää. Tämä on tarpeellista, koska pelaaja voi haluta olla vuorovaikutuksessa objektin kanssa. Vastaavasti, kun passiivinen objekti lähestyy etäpelaajaa, täytyy se vähitellen piirtää lähemmäksi etäpelaajan aikaa. Tämä tulee tehdä sen takia, että jos etäpelaajan ja passiivisen objektin välillä on interaktiota, piirretään se paikalliselle pelaajalle vasta viiveen suuruisen ajan jälkeen. Smed ja Hakonen selventävät ajatusta seuraavasti: Paikalliselle pelaajalle piirretty kuva etäällä tapahtuvasta interaktiosta on jo tapahtunut, vaikka

interaktio itsessään tapahtui reaaliaikaisesti. Silloin kun paikallinen pelaaja on itse interaktiossa, täytyy se piirtää reaaliaikaisesti, koska se tapahtuu juuri silloin kyseiselle pelaajalle.

Ideaa havainnollistaa kuva 4.6, jossa kuvat a-d esittävät tilanteen vaaleaa aluksen pelaajalle. Kuvat e-h esittävät saman tilanteen tumman aluksen pelaajalle. Kuvasarjassa katkoviivainen soikio kuvaa ammuksen todellisen sijainnin ja musta kolmio kuvaa ammuksen pelaajalle näytetyn sijainnin.



Kuva 4.6: Paikallisen havainnoinnin suodatuksen esimerkkikuvasarja: Kuvat a-d esittävät tilanteen vaalean aluksen pelaajalle ja kuvat e-h kuvaavat saman tilanteen tumman aluksen pelaajalle.

Smed ja Hakonen jatkavat kertomalla jokaisen pelaajan omaavan oman näkymänsä pelimaailmaan. Jokaisella objektilla on tuossa maailmassa paikka-koordinaatit (x, y, z) mutta niihin on myös liitetty viive (t) . Täten koordinaatisto on kolme ja puoli ulotteinen. Paikallisen pelaajan viive on nolla eli $t = 0$. Etäpelaajien viive vaihtelee ja sen mukana myös niille asetettu t -arvo. Jokai-

selle pelaajalle voidaan muodostaa oma ajallinen poikkeuskäyrästä (*temporal contour*) pelimaailmaan, kun viive ja t -arvot ovat selvillä. Smed ja Hakonen kutsuvat ajallista poikkeuskäyrästä myös kausaaliseksi- eli syy-seuraus-tasoksi (*causal surface*). Ajallinen poikkeuskäyrästä kuvaa t :n arvon missä tahansa pelimaailman pisteessä. Jos passiivinen objekti on esimerkiksi ammus, jonka paikallinen pelaaja ampuu, sen t -arvo on ensin nolla. Ammuksen liikkuessa etäämmälle paikallisesta pelaajasta ja lähestyessään jotain etäpelaajaa, ammuksen t -arvo kasvaa lineaarisesti saavuttaen saman arvon kuin etäpelaaja ollessaan sen kohdalla. Smed ja Hakonen täydentävät, että passiivisen objektin liikkeen tulisi näyttää pelaajalle tasaiselta. Tämän vuoksi on hyvä käyttää tasaisen lineaarisesti kasvavaa ajallista poikkeuskäyrästä kohteiden välillä. Kaiken interaktion tulisi olla myös realistista ja johdonmukaista, eli tapahtumien kausaalisuus pitäisi pysyä oikeana. Smed ja Hakonen summaavatkin kausaalisen tason vaatimukset seuraavaan kolmeen sääntöön:

- Pelaajan tulee pystyä vuorovaikuttamaan reaaliaikaisesti lähellä olevien objektien kanssa.
- Pelaajan tulee pystyä seuraamaan etäällä tapahtuvaa vuorovaikutusta reaaliaikaisesti, vaikkakin se voi olla tapahtunut jo aikaisemmin.
- Ajallisten vääristymien havaitseminen tulisi olla mahdollisimman huomaamaton.

Smedin ja Hakosen mukaan ensimmäisestä säännöstä seuraa paikallisen havainnoinnin suodatuksen suurin rajoitus: Pelaaja ei voi suoraan vuorovaikuttaa etäpelaajan kanssa. Pelaajat voivat vaihtaa keskenään passiivia objekteja, kuten luoteja, nuolia tai ohjuksia, mutta he eivät voi esimerkiksi nyrkkeillä. Smed ja Hakonen selventävät, että kun pelaajat lähestyvät toisiaan, niin ajallinen vääristymä tulee aina vain ilmeisemmäksi. Lopulta pelaajat saavuttavat kriittisen rajan, jolloin he ovat liian lähellä toisiaan ja vuorovaikutus passiivistenkin objektien kautta on mahdotonta. Zydan [2005a] mukaan paikallinen havainnointisuodatus vaikuttaa informaation periaatteen yhtälöön seuraavasti: Ajallisuus T pienenee ja prosessointitehoa tarvitaan lisää, eli P kasvaa.

Paikallinen havainnointisuodatus olettaa, että tarkat viiveet solmujen välillä ovat tiedossa. Todellisuudessa viive ja verkon kuorma vaihtelevat ajan mittaan. Tämä aiheuttaa sen, että kausaalitason huippujen on muututtava viiveen

vaihteluiden mukaan. Smedin ja Hakosen mukaan suuri huojunta on ongelmallista. Huojunta saattaa aiheuttaa sen, että kausaalitasot päivitetään ja huippujen arvot vaihtelevat. Tällöin matkalla oleva passiivinen objekti voi pelaajan mielestä näyttää liikkuvan edestakaisin. Tämä johtuu siitä, että sen aika-arvoja muutetaan kausaalitason muuttuessa ja objektin t -arvo pomppii edestakaisin menettäen tasaisen lineaarisen ajallisen siirtymän. Smed ja Hakonen antavat toisen hankaluuksia aiheuttavan esimerkin kertomalla etäpelaajan yllättävää poistumisesta pelistä. Tällöin passiivinen objekti pomppaa eteenpäin ajassa heijastellen syy-seuraustason päivitystä.

4.2.4 Kiinnostuksen alueen mukainen suodatus

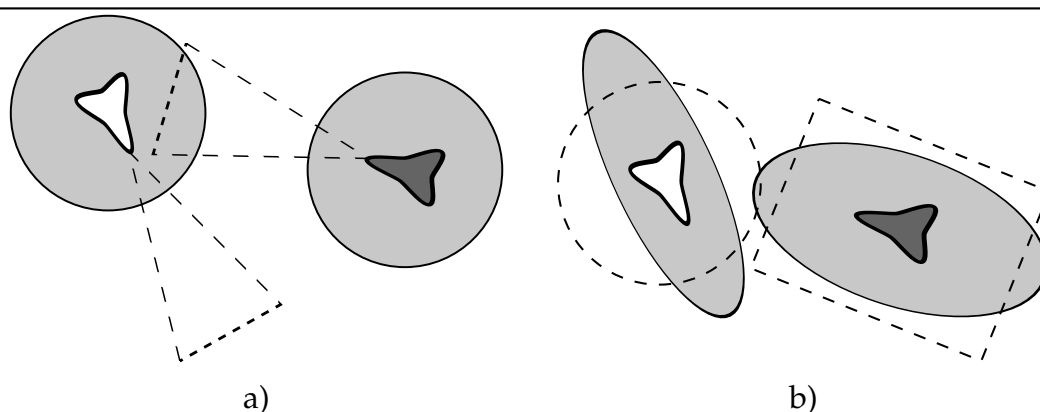
Miten voitaisiin vähentää lähetettävien pakettien määrää tilanteessa, jossa pelimaailma on esimerkiksi erittäin laaja? Tällaisessa ympäristössä kaukana toisistaan olevat objektit eivät välttämättä ole kiinnostuneita toistensa tiloista. Smedin ja Hakosen mukaan luonnollinen menetelmä on tällöin välittää päivitykset vain niille solmuille, jotka ovat kiinnostuneita lähettävän objektin tilan muutoksista. Tällaista tietojen lähettämistä vain niistä kiinnostuneille kutsutaan suodatukseksi kiinnostuksen alueen mukaan (*area-of-interest filtering*). Smed ja Hakonen sanovat, että solmulla on mahdollisuus ilmaista kiinnostuksensa vain osaan saatavilla olevista tiedoista.

Smed ja Hakonen nimittävät kiinnostusta johonkin tietoon kiinnostuksen alueeksi, eli auraksi. Heidän mukaansa aura on yksinkertaisesti alueen tai tilan osa, missä interaktio tapahtuu. Kun objektin aura ei leikkaa minkään muun objektin auraa, päivityksiä ei tapahdu. Kahden objektin aurojen leikatessa kumpainenkin lähettää ja vastaanottaa päivityksiä toisiltaan. Smedin ja Hakosen mukaansa aura yleensä mukaillee mallinnetun objektin havaintokykyä. Objektilla voikin olla erilliset aurat eri aisteille, esimerkiksi visuaalisuuteen tai kuuloon perustuvat aistit. Aura voi olla muotonsa puolesta minkäläinen vain eikä sen Smedin ja Hakosen mukaan tarvitse olla pelimaailman geometrian mukainen.

Smed ja Hakonen jatkavat esittelemällä kolme tapaa auran määrittämiseksi. Näitä ovat kaavaan perustuva menetelmä, rajauslaatikkomenetelmä ja solupohjainen menetelmä. Ensiksi mainittu perustuu matemaattisesti esitettyyn kaavaan, joka rajaa alueen. Kaavan määrittämä alue voi olla erittäin tarkka mutta samalla se lisää toteutuksen kompleksisuutta, jolloin vaadittava laskenta monimutkaistuu. Muodostamalla auran ympärille rajauslaatikon (*boun-*

ding box), laskentaa saadaan huomattavasti yksinkertaistettua kaavaan perustuvaan ratkaisuun verrattuna. Soluperustainen suodatus perustuu pelimaailman jakamiseen soluihin. Siinä vaihtoehdossa objekti on kiinnostunut tiedoista, jotka tapahtuvat niissä soluissa, joita sen aura leikkaa. Smed ja Hakonen sanovat soluperustaisen ratkaisun olevan myös helpompi ja kevyempi toteuttaa kuin kaavaan perustuvan auran. Huonoksi puoleksi he kertovat suodatuksen tarkkuuden olevan huonompi. Rajauslaatikkotapaa he kuvailevat suodatukseltaan paremmaksi kuin suuriin soluihin perustuvaa soluperustaista menetelmää.

Viestien suodatus on aina symmetristä. Jos aurat leikkaavat, molemmat osapuolet saavat toistensa päivitykset. Smed ja Hakonen esittelevät auran jaon fokukseen ja nimbukseen. Fokus tarkoittaa objektin omaa havainnointikyvyn aluetta, eli havainnointialuetta. Nimbus on vastaavasti alue, jolta muut objektit voivat tehdä tästä objektista havaintoja. Nimbusta voisi kutsua objektin havaittavaksi alueeksi. Tällä jaolla viestien määrää saadaan entisestään vähennettyä, koska havainnoitava objekti lähettää päivityksiä vain niille havainnoijille, joiden fokus leikkaa havainnoitavan objektin nimbuksen. Tilanne on havainnollistettu kuvassa 4.7 a), jossa tumman aluksen fokus leikkaa vaalean aluksen nimbuksen ja täten vaalea alus lähettää päivitystietonsa tummalle alukselle. Tumma alus ei kuitenkaan lähetä tietojaan vaalealle, koska vaalean aluksen fokus ei leikkaa tumman aluksen nimbusta. Kuva 4.7 b) näyttää, että fokukset ja nimbukset voivat olla eri muotoisia. Vaalea alus saa päivityksiä tummalta kuvan tilanteessa.



Kuva 4.7: Aurat: fokus (katkoviivalla rajattu) ja nimbus (harmaa alue)

Suodatus kiinnostuksen alueen perusteella on Smedin ja Hakosen mukaan

joko sisäistä tai ulkoista suodatusta. Edellä kuvatut tavat ovat sisäistä suodatusta, koska ne käyttävät hyväkseen ohjelmiston sisäistä dataa päättäessään mille solmuille päivitys lähetetään. Näin päästään hienojakoiseen tiedon välittämiseen mutta rangaistuksena on jälleen kerran lisäprosessointiin vaadittava aika. Tarkasteltaessa yhtälöä 4.1 huomataan, että suodatus kiinnostuksen alueen perusteella vähentää viestien lukumäärää M ja kaistaa viesteille B . Sen sijaan solmujen välinen lisääntynyt organisointi kasvattaa kuitenkin prosessin määrää P .

Ulkoiseksi suodatukseksi Smed ja Hakonen nimittävät verkon ominaisuuksien mukaan tehtyä viestien välitystä. Tällainen voi olla esimerkiksi verkkoosoitteen perusteella tehty suodatus. Joukkolähetys on yksi tapa suorittaa ulkoista suodatusta, jonka tarjoaa verkko itsessään. Ulkoinen suodatus on myös nopeampaa suorittaa kuin sisäinen. Smedin ja Hakosen mukaan joukkolähetysten ryhmät voivat kuitenkin olla haastavia muodostaa. Solun pitää liittyä niihin ryhmiin, joiden tiedot sitä kiinnostavat. Ryhmien muodostamiseen on Smedin ja Hakosen mukaan käytännössä kaksi tapaa. Ensimmäinen tapa muodostaa ryhmän jokaista objektia kohden, kun taas toinen tapa on hieman soluperustaisen menetelmän kaltainen, jossa maailma on jaettu alueisiin ja yhtä aluetta kohden muodostetaan yksi joukkolähetysryhmä.

Kritiikkiä kiinnostuksen alueen mukaiseen suodatukseen esittää Huang [2006]. Hänen mielestään aurojen määrittäminen onnistuu pienelle joukolle objekteja. Huang jatkaa, että aurojen käyttö ei skaalaudu hyvin suurelle objektimäärälle.

4.2.5 Synkronoitu simulaatio

Synkronoidussa simulaatiossa (*synchronized simulation*), eli samaan aikaan tahdistetussa simulaatiossa, kaikilla solmuilla on absoluuttinen konsistenssi. Se saavutetaan monistettua verkkoarkkitehtuuria käyttämällä. Tässä arkkitehtuurissa jokaisella solmulla on oma kopio pelintilasta ja tilat ovat yhtenevät. Yhtenevyys saavutetaan pitämällä yllä absoluuttinen konsistenssi. Smed ja Hakonen selvittävät, että viestien määrän vähentämiseksi solmujen väliltä tilat pyritään pitämään synkronoituina. He jatkavat, että pelimaailman tapahtumat voidaan jakaa kahteen luokkaan: simulaation synnyttämiin ja pelaajan aloittees- ta syntyneisiin tapahtumiin. Simulaation synnyttämiä tapahtumia nimitetään deterministisiksi. Arvaamattomamman pelaajan aloitteet eivät ole deterministisiä – ainakaan simulaation kannalta. Jokaisen solmun ajaessa samaa simulaa-

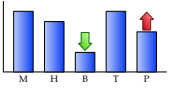
tion kopiotaan deterministisiä tapahtumia ei tarvitse välittää muille solmuille. Deterministiset tapahtumat syntyvät jokaisessa monistetussa simulaatiossa kuitenkin. Näin saadaan vähennettyä päivityspakettien määrää. Jäljellä jäävä kommunikaatio käsittää pelaajien antamat käskyt simulaatiolle.

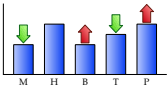
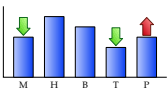
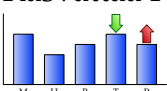
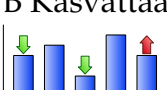
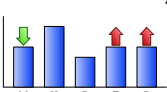
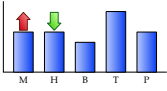
Smed ja Hakonen lisäävät synkronoidun simulaation toteutuksen tarvitsevan konsistenssin tarkastus- ja toipumismekanismeja. Tätä tarvitaan varsinkin tilanteissa, joissa osa viesteistä häviää matkalla. Luonnollisena vaihtoehtona olisi tällöin käyttää viestintään TCP-protokollaa. Synkronoitu simulaatio takaa tai oikeastaan perustuu absoluuttiseen konsistenssiin. Sen avulla informaation periaatteen yhtälön (4.1) muuttujista saadaan arvoltaan pienennettyä M :ää. Viestejä tarvitsee lähettää vähemmän, koska deterministisiä viestejä ei tarvitse välittää. Vastapainoksi laskentaa tarvitaan enemmän eli P nousee. Myöskin ajallisuus eli T huononee, koska absoluuttinen konsistenssi on ylläpidettävä ja toipumismekanismi voi hetkellisesti lisätä viivettä, kun tilat saattetaan synkrooniin.

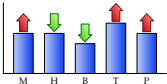
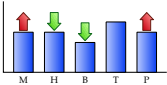
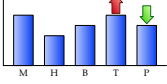
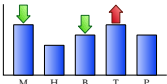
4.3 Kompensaatiotekniikoiden yhteenveto

Taulukossa 4.4 on tiivistetty resurssirajoitusten kompensaatiotekniikoiden tietoja. Taulukko on koostettu edellä esitellyistä sekä Zydan [Zyda, 2005a] esittelemistä tekniikoista. Taulukon viimeiseen sarakkeeseen on koottu huomiota ja kommentteja menetelmien soveltuvuudesta mobiilipeleille.

Taulukko 4.4: Yhteenveto kompensointitekniikoista

Kompensaatiotekniikka	Vaikutus informaation periaatteen yhtälöön	Selitys	Soveltuvuus mobiilipeleille
Viestien pakkaaminen	Pienentää B Kasvattaa P 	Viestit pakataan ja puretaan.	Laskentaresurssien puitteissa soveltuva.

Kompensaatio- tekniikka	Vaikutus informaation periaatteen yhtälöön	Selitys	Soveltuvuus mobiilipe- leille
Viestien yhdistäminen	Pienentää M, T Kasvattaa B, P 	Sulauttaa useita viestejä yhteen pakettiin käyttäen aikakatkaisua, pakettien määrää tai hybridiä	Riippuu vii- vevaatimuk- sista ja prosessori- tehosta.
Liikkeen ennustaminen	Pienentää M, T Kasvattaa P 	Objektien liikkeitä ennustetaan käyttäen edellisiä sijaintitietoja.	Riippuu vii- vevaatimuk- sista ja prosessori- tehosta.
Paikallinen havainnointi- suodatus	Pienentää T Kasvattaa P 	Kauempana olevat kohteet näytetään solmujen välisten viiveiden pohjalta muodostetun ajallisen poikkeutuskäyrästä perusteella.	Herkkä huojunnalle, joten sopivuus mobiiliverk- kojen päälle huono.
Kiinnostuksen alueen mukainen suodatus	Pienentää M, B Kasvattaa P 	Määrittää ja käyttää informaation suodattimia objekteille.	
Synkronoitu simulaatio	Pienentää M Kasvattaa P, T 	Kaikilla solmuilla on absoluuttinen konsistenssi.	
Joukkolevitys	Pienentää H Kasvattaa M 	Viestit lähetetään solmujen osajoukolle. Osajoukko perustuu lähdesolmuun tai alueeseen.	

Kompensaatio- tekniikka	Vaikutus informaation periaatteen yhtälöön	Selitys	Soveltuvuus mobiilipe- leille
Projisoinnin yhteen kokoaminen	Pienentää H, B Kasvattaa M, P, T 	Ennalta määrätyt hienojakoiset joukkolevityksen ryhmät.	
Tarkkuuden tason pienentäminen	Pienentää H, B Kasvattaa M, P 	Epätarkempaa dataa kauempana oleville kohteille.	
Palvelinverkko	Pienentää P (per solmu) Kasvattaa T 	Käyttää useampia palvelimia, jotka kommunikoivat keskenään.	Soveltuu sijoitettaessa palvelimet lähelle ope- raattoreiden ydinverkko- ja.
Mukautuva kommunikoin- tiarkkitehtuuri	Pienentää T, B, M Kasvattaa T, B, M 	Valitsee joko vertaisverkko- tai asiakaspalvelinarkki- tehtuurin verkon ominaisuuksien perusteella.	Käyttökel- poinen moni- muotoisissa mobiili- verkoissa.

Smed ja Hakonen [2006] toteavat, että loppujen lopuksi resurssirajoituk-
sia ei voi kokonaan piilottaa, vaikka käytettäisiin mitä tahansa kompensaa-
tiotekniikkaa. He jatkavat, että pienellä onnella on kuitenkin mahdollista va-
lita paikat, joissa resurssirajoitukset ilmenevät pelaajalle. Paikkojen toivotaan
olevan sellaisia, että ne aiheuttavat mahdollisimman vähän harmia pelaajalle.
On myös huomattavaa, että vaikka Smed ja Hakonen tuovat suoraviivaises-
ti käyttöön resurssirajoitusten kompensatiotekniikat verkkopeleistä puhues-
saan, tekniikoiden alkuperä on lähtöisin verkotettujen virtuaaliympäristöjen

yhteydestä. Täten esiteltävien tekniikoiden käyttöalue mobiilien moninpelien yhteydessä on hieman rajoittunut. Vaikkakin voidaan ajatella, että esimerkiksi lautapeli siirrettynä mobiiliksi moninpeliksi sisältää jaetun virtuaalisen ympäristön, pelilaudan, ei tässä esitetyt tekniikat ole silti täysin käyttökelpoisia laudan kaltaisessa diskreetissä ympäristössä, jossa pelimaailman granulariteetti on suuri. Tällä tarkoitan, että esimerkiksi liikkeen estimointiin pureutuvat tekniikat eivät tällaisessa tilanteessa toimi tehokkaasti, kun liike ei ole jatkuvaa ja sulavaa vaan ennemminkin hyppäyksellistä. Informaation periaatteen ja sen yhtälön esittämä resurssirajoitusten osa-aluiden selventäminen on sen sijaan käyttökelpoinen myös matkapuhelinverkkojen välityksellä tapahtuvien mobiilipelien tiedonsiirtoa tarkasteltaessa.

5 FLOSTERS

Tässä luvussa tarkastelen pelisuunnittelun ja matkapuhelimen sosiaalisena laitteena suomia mahdollisuuksia mobiilipeleille. Tarkastelun kohteena on konsepti, jossa resurssirajoituksia kierretään pelisuunnittelullisesti käyttämättä kehittyneitä resurssirajoitusten kompensatiotekniikoita. *Flosters* on ideoimani ja kehittämäni mobiilipelikonsepti ja -prototyyppi. Konseptiosuus käsittelee mobiililaitteiden, -pelien ja -verkkojen ymmärtämisen huomioimista pelisuunnittelussa. Prototyyppi on tutkielmani konstrukttiivinen osuus. Luvun lopussa pohdin mobiilipelien suunnittelumallien näkökulmasta resurssien niukkuuden kiertämistä sekä esitän suunnitelman *Flostersin* jatkokehittämiseksi.

5.1 Konsepti

Flosters on sosiaalinen mobiilipelikonsepti. Se käyttää hyväkseen Internetin sosiaalisen median palveluita¹ ja niiden avoimia ohjelmointirajapintoja. Sosiaalisen median palveluiden hyödyntämisen idea syntyi siitä ajatuksesta, että varsinkin peliprototyyppeihin voi olla vaikeaa saada testipelaajia. Konseptin suunnittelun keskeisenä ajatuksena oli selvittää, mitä hyötyjä mobiileille moninpeleille ja niiden nopealle prototypoimiselle on Internetin sosiaalisen median palveluista. Sosiaalisen median palveluiden luokassa nimeltä sosiaaliset verkostot on yleensä ystävaverkostot (social graph) valmiina ja täten tiedot siitä, ketkä ovat kunkin käyttäjän ystäviä. Tällaisia sosiaaliset verkostot palveluita ovat esimerkiksi MySpace, Facebook, Twitter ja Jaiku. Palvelut mahdollistavat sovellusten toteuttamisen kolmansille osapuolille avaamalla sisäänpääsyn järjestelmään avointen ohjelmointirajapintojen (API) kautta.

Prototyypeissä ja peliprototyypeissä käytetään yleensä myös keksittyä valedataa. Sosiaalisen median palveluiden tiedot ovat käyttäjien itsensä lisäämiä ja siten jollain tavalla merkityksellisiä käyttäjille. Tällaisen tiedon käyttäminen prototyypissä lisää välittömästi sen mielenkiintoisuutta ja relevanttiutta testipelaajille. Heidän ei tarvitse kuvitella valedatan sisältämiä tietoja henkilökohtaisiksi vaan tieto on aitoa. Niin ikään ystävät ovat testipelaajan aitoja ystäviä - ainakin palvelun kannalta. Tämä oli myös yksi motiivi sosiaalisen median

¹ Sosiaalisen median palveluiden määrittelystä ja luokittelusta löytyy enemmän tietoa Katri Lietsalan ja Esa Sirkkusen kirjoittamasta Parteco-projektin piakkoin ilmestyvästä loppuraportista nimeltä "Social Media – Introduction to Tools and Processes of Participatory Economy". Projektin blogi löytyy osoitteesta <http://www.somelab.fi/>

palvelujen hyväksikäyttämiseksi prototypoinnissa.

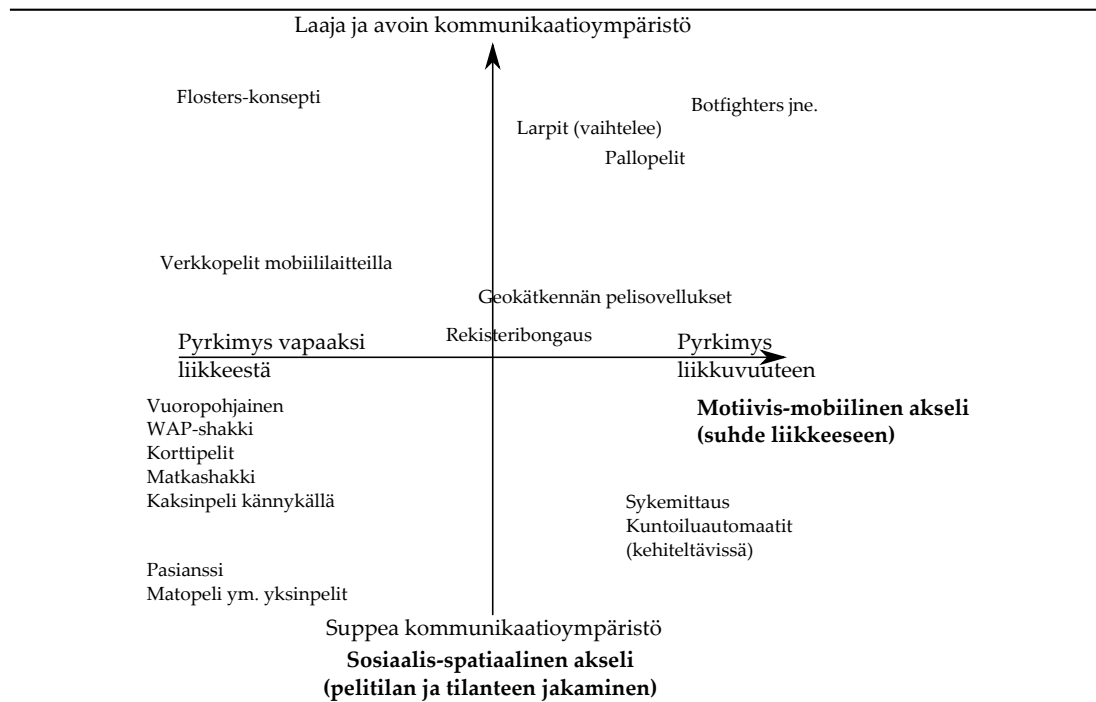
Konseptin kehittäminen lähti liikkeelle työni tarpeista. Työskentelen Tampereen yliopiston Hypermedialaboration pelitutkimusyksikössä. Projektin nimi on GameSpace² ja se on keskittynyt ajanvietteellisten monenpelaajan mobiilipelien ideoinnin, prototypoinnin ja evaluoinnin tutkimiseen. Flostere-prototyyppiä tarvittiin GameSpacen järjestämään toiseen prototypointityöpajaan. Työpaja on yksi GameSpacen kaikkiaan seitsemästä työpajasta. Näistä kolme on keskittynyt ideointiin, kaksi prototypointiin ja kaksi evaluointiin. Työpajan aiheena oli tarkastella kolmea keskitason prototypointimenetelmää, joista yksi sisälsi Flostereita.

Itse konsepti on muunnelma sanapeleistä. Perinteisesti sanapeleissä muodostetaan annetuista kirjaimista sanoja (vrt. *Scrabble*). Flostere mahdollistaa tämän mutta lisäksi siinä on mahdollisuus löytää ystävien nimiä, koska kirjainjoukko on muodostettu kahdesta ystävästä. Kaksi ystävästä valitaan satunnaisesti ja toiselta valitaan sukunimi sekä toiselta etunimi. Nämä kirjaimet sekoitetaan ja pelaajan tehtävänä on löytää kirjaimista joko ystävä tai muodostaa mitä tahansa mahdollisimman pitkiä sanoja. Ystävien löytämisestä tulee myös konseptin nimi Flostere, joka nimenä muodostuu sanoista "friends lost in letters". Ystävien löytämisestä saa enemmän pisteitä kuin muista sanoista. Sanat pisteytetään myös niiden pituuden ja sanan sisältämien kirjaimien perusteella. Harvinaisempia kirjaimia sisältävät sanat saavat enemmän pisteitä. Pisteet tallennetaan tietokantaan ja pelaaja voi halutessaan katsoa, mihin kohtaan listaa hän muiden pelaajien suhteen sijoittuu. Ystävästä löytämisestä palkitaan lisäksi näyttämällä kyseisen ystävästä profiilikuva, joka myöskin saadaan sosiaalisen median palvelusta.

Ystävästä nimien tunnistaminen kirjaimista voi aluksi vaikuttaa haastavalta. Siksi Flostere-konseptiin suunniteltiin aputoiminto, jota käyttämällä pelaaja saa vihjeitä ystävästä. Vihjeinä annetaan kolme satunnaista tietoa ystävästä. Tiedot tulevat sosiaalisen median palvelun käyttäjien profiileista ja voivat sisältää esimerkiksi ystävästä asuinpaikan, työ- tai opiskelupaikan, syntymäajan, hänen suosikkimusiikkiaan, lempielokuvia, tv-sarjoja ja niin edelleen.

Koska Flostereissa on kyse omien ystävästä tunnistamisesta ja sen vuoksi sosiaalisesta pelaamisesta, konsepti ottaa huomioon myös viestien lähettämisen ystävästä. Konseptissa on mahdollista lähettää viesti löydetylle ystävästä. Viesti voi olla jokin valmiista malliteksteistä tai pelaaja voi itse kirjoittaa sen.

² <http://gamelab.uta.fi/GameSpace>



Kuva 5.1: Flosters Suomen kartalla

Viesti lähetetään ystävälle sosiaalisen median palvelun saapuneihin viesteihin tai käytetään jotain muuta palvelun tarjoamaa viestintätapaa. Viestien välittämiseen tulevat kysymykseen myös matkapuhelimen tarjoamat viestintätavat, kuten soittaminen sekä teksti- ja multimediaviestit. Kuvassa 5.1 olen sijoittanut Flostersin Suomen esittelemälle kartalle. Flosters sijoittuu kuvassa vasempaan reunaan, koska se ei ole liikkeeseen kannustava mobiilipeli Järvisen toivomalla tavalla. Sen sijaan sen sijoittuminen kuvion sosiaalisella akselilla tapahtuu melko ylös Flostersin mahdollistamien useiden kommunikointitapojen vuoksi. Konseptin kehittäminen siten, että se olisi sijoitettavissa enemmän oikealle motiivis-mobiilisella akselilla, voisi tapahtua integroimalla se yhteen Jaikon kaltaisten mini-bloggaukspalveluiden kanssa. Bloggauksen lisäksi Jaiku tukee käyttäjien presenssi- sekä kontekstietoa. Kontekstiedosta käy summittaisesti ilmi käyttäjän sijainti ja tätä tietoa hyödyntäen olisi Flostersissa mahdollista löytää ystäviä, jotka ovat myös fyysisesti lähellä pelaajaa.

5.2 Prototyyppi

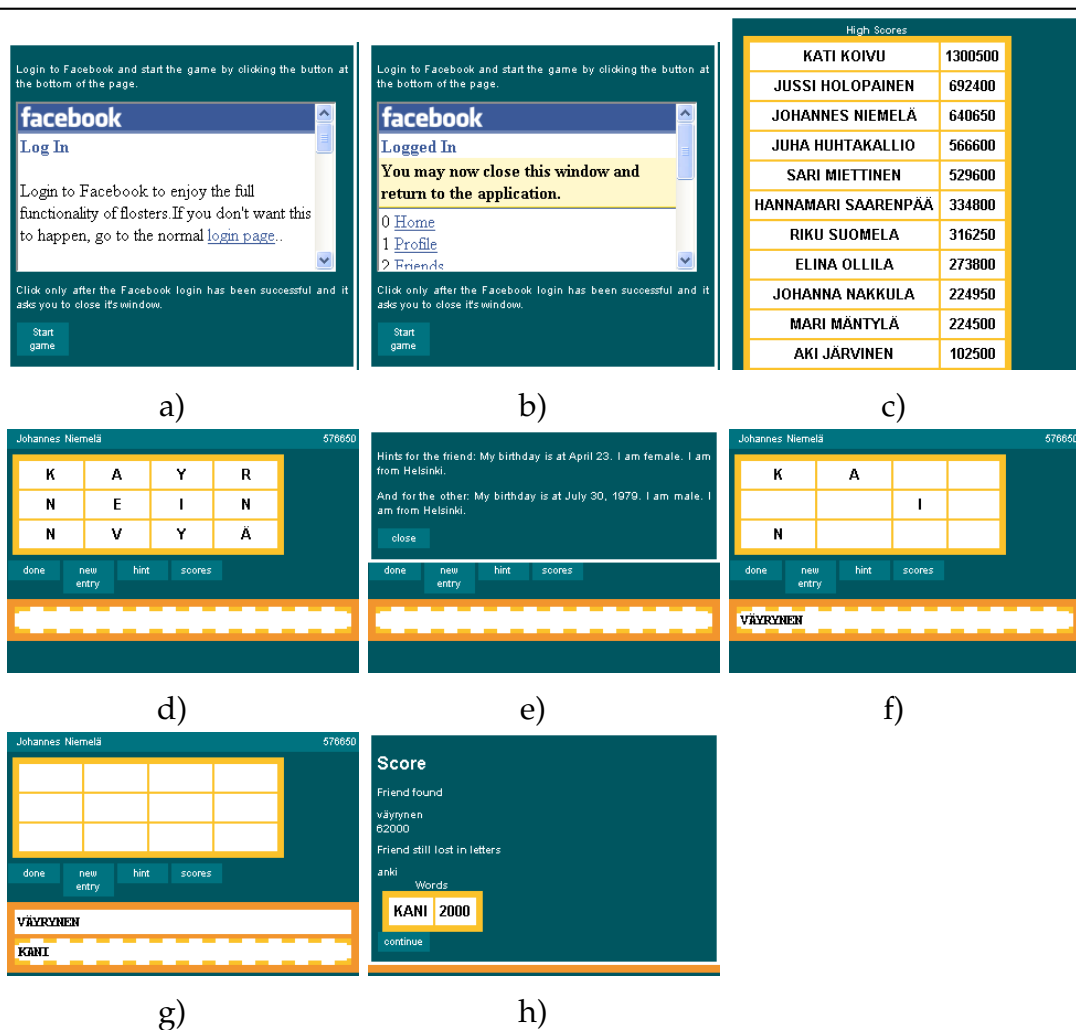
Toteutettu Flostere-prototyyppi³ toimii matkapuhelimen selaimessa. Prototyyppi on sikäli karsittu versio konseptista, että siinä ei ole toteutettuna mitään suoraan kommunikointiin tai ystävien yhteydenpitoon liittyviä ominaisuuksia. Flostere-prototyyppi keskittyy pelikonseptin ydinmekaniikan toimivuuden selvittämiseen. Prototyypistä voisi sanoa, että se ei ole vielä tällä hetkellä moninpeli. Suoraa tai epäsuoraa interaktiota pelaajien välillä ei ole muuten kuin pisteiden listauksen kautta. Prototyyppi kuitenkin valaisee sosiaalisen median palvelun ja valmiiden ystävä tietojen sekä -verkostojen käytön soveltuvuutta ja mielenkiintoisuutta mobiilipeleille.

Prototyypin toteutukseen valitsin ystäväverkosto palveluiden joukosta Facebookin. Facebook valittiin sen suosion mutta myös useiden eri kielillä toteutettujen API:en vuoksi. Prototyyppi oli ensiksi tarkoitus toteuttaa Flash Litellä, joka on Internetissä suosittu Flashin mobiililaitteille kehitetty versio. Flash Litessä ongelmaksi muodostui tietojen hakeminen palvelimelta. Flash Lite halusi tarkistaa ja vahvistaa jokaisen yhteydenoton yhteydessä käytettävät verkkoa-setukset ja sen, saako yhteyttä ylipäättään muodostaa. Tämä ei ollut hyväksyttävää, koska prototyypin pelikokemus olisi merkittävästi kärsinyt tästä. Flostere sin pitää kuitenkin melko säännöllisin väliajoin kommunikoida palvelimen kanssa.

Saattaa olla, että vähäisen Flash Lite -kokemukseni vuoksi en osannut ohjelmoida prototyyppiä optimaalisesti. Ongelma saattoi koskea pelkästään Flash Liten versioita 1 ja 1.1; uudemmassa 2.0 versiossa ei ehkä olisi ilmennyt ongelmia. Ongelma olisi siis mahdollisesti ollut kierrettävissä, mutta tiukan aikataulun vuoksi toteutus alustaa päätettiin vaihtaa. Vaihtoehtona oli selainpohjainen prototyyppi. Selainpohjainen tarkoittaa sitä, että peli toteutettiin aivan tavalliseksi XHTML-sivuksi. Toiminnallisuutta varten käytettiin JavaScriptiä. Alussa näytti, että matkapuhelimen selain ei tukisi JavaScriptiä riittävästi. Tutkimalla asiaa hieman tarkemmin ja jatkokehittämällä sain kuitenkin tarpeelliset toiminnot toimimaan myös puhelimen selaimessa, jolloin kehitystyö pääsi tällä alustalla jatkumaan. Kommunikointi sivun ja palvelimen kanssa hoidetaan AJAX:a käyttäen, vaikkakaan palvelin ei käytä XML-muotoista vastausviestiä vaan pelkkää tekstivastausta.

³ Prototyyppi on pelattavissa osoitteesta <http://gamelab.uta.fi/flostere/>. Pelaamiseen tarvitaan voimassaoleva Facebook-tunnus.

Toteutetun prototyypin käyttöliittymäkuvat on esitelty kuvassa 5.2. Ensimmäiset kaksi kuvaa (a ja b) esittävät kirjautumista Facebookiin ja onnistunutta kirjautumista. Kuva 5.2 c) näyttää pistelistauksen. Kuvissa d-h on pelitilannekuvia. Kuva d) esittää pelin alkua, e) näyttää vihjeitä kahdesta ystävästä. Kuvassa f) on muodostettu löydetyn ystävän nimi ja kuvassa g) on muodostettu sana ”kani”. Viimeinen kuva (h) näyttää pisteet löydetylle ystävän nimelle ja muille muodostetuille sanoille sekä näyttää, mikä ystävän nimi jäi löytymättä.



Kuva 5.2: Flostersin käyttöliittymäkuvat: a) Kirjautuminen Facebookiin, b) Onnistunut kirjautuminen, c) Pistelistaus, d) Pelinäkö, e) Vihjeet, f) Muodostettu nimi, g) Muodostettu sana ja h) Löytyneen ystävän ja sanan pisteet.

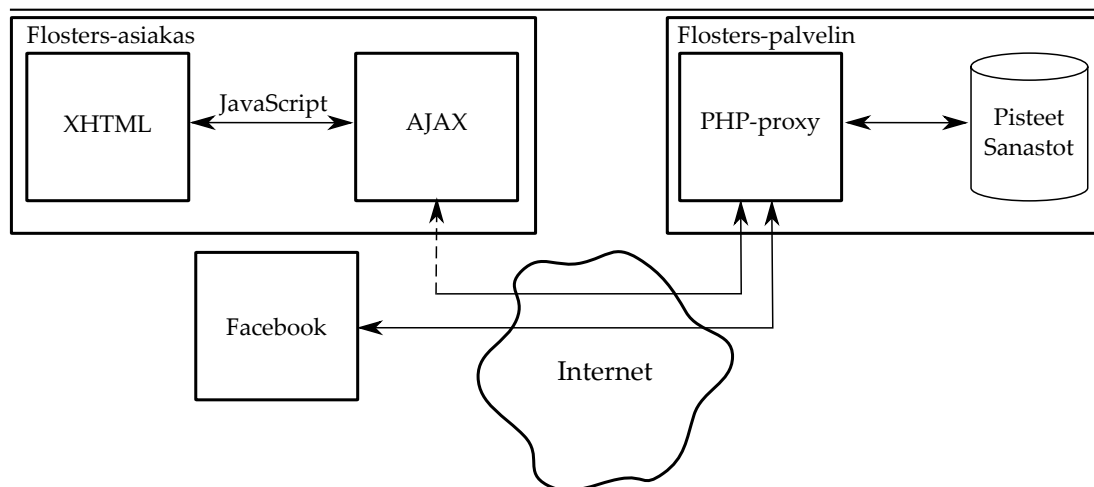
Palvelin on toteutettu PHP-kielellä, ja se tekee kyselyt ja päivitykset Facebookiin API:a käyttäen Facebookin palvelimille. Tämän osalta palvelin ei oi-

keastaan toimi palvelimena vaan sen voitaisiin katsoa olevan proxy-tyyppinen välittäjä. PHP-sivu valittiin alun perin proxyksi Flash Lite -toteutuksen takia, koska Flash Lite ei sisällä ActionScript -kielistä Facebook API:a, jolla voitaisiin kommunikoida Facebookin kanssa suoraan. Facebook tarjoaa kyllä PHP-kieliset API:t. Sosiaalisen median palvelujen nopeaa kehitystä kuvaa se, että kehitystyön puolivälissä Facebookin JavaScript API tuli saataville. Tuolloin oli kuitenkin liian myöhäistä muuttaa toteutusta ja vaihtaa PHP-proxy JavaScriptiin.

Suoraan sivulle toteutettu JavaScriptiä käyttävä kommunikaatio Facebookin kanssa olisi todennäköisesti helpottanut kirjautumisprosessia. Nyt kirjautumisprosessi Facebookiin on hieman hankala, minkä myös testipelaajat raportoivat. Toisaalta mahdolliseen Flash Lite -toteutukseen verrattuna kirjautuminen oli nyt yksinkertaisempaa, koska pelaaja käytti koko ajan vain selainta. Flash Litella tehdyssä toteutuksessa olisi jouduttu avaamaan Flash Lite -soittimen lisäksi myös erillinen selain kirjautumista varten.

Toisaalta palvelimen voidaan katsoa olevan todellinen palvelin, koska se käyttää tietokantaa pisteiden tallennukseen. Tietokantaa käytetään myös sanastoihin. Tällä hetkellä prototyyppi tarkistaa muodostetut sanat sekä englannin- että suomenkielisistä sanastoista. Kummatkin kielet ovat tuettuina, koska GameSpace -projektin virallinen kieli on englanti ja osa yrityskumppaneiden edustajista ei ole suomenkielisiä. Suomenkielinen versio on kuitenkin tarjottu suomalaisille pelaajille. Toisaalta eri kielten tarkistusten toteuttaminen tukee sen tyylistä pelaamista, jossa pelaaja yrittää parantaa vieraan kielen taitojaan keksimällä vieraskielisiä sanoja. Kuvassa 5.3 on esitetty Flosters-pelin rakenne asiakkaan, palvelimen ja Facebookin kannalta.

Vaikka Flosters on uudentyyppinen mobiilipeli, ja yksi sen suunnittelun lähtökohdista oli kiertää resurssirajoituksia pelisuunnittelun keinoin, käyttää se silti yhtä kohdassa 4.2 esiteltyä resurssirajoitusten kompensatiotekniikkaa. Prototyypissä käytetään protokollan optimointia. Tarkemmin sanottuna prototyyppi käyttää viestien yhdistämistä ja tiedon bufferointia. Myös viestien pakkaaminen olisi ollut kohtuullisen helposti mahdollista, koska viestit olivat tekstiä. AJAX:a käyttäen olisi ollut helppoa tarkistaa palvelimelta, ovatko muodostetut sanat pelaajan ystävien nimiä. Näin toteutinkin prototyypin aluksi mutta nopeasti selvisi, että pelikokemus heikkenee, kun tarkistamiseen menee verkon viiveen takia pitkä aika. Tämän vuoksi päädyin tilanteeseen, jossa asiakkaan ja palvelimen välillä vaihdettujen ensimmäisten peliviestien joukossa lähetetään viesti, johon on tiivistetty kaikki pelaajan ystävien nimet.



Kuva 5.3: Flostersin arkkitehtuuri

Näin nimien tarkistaminen saadaan siirrettyä palvelimelta asiakkaalle ja vaste siltä osin nopeutuu huomattavasti. Sanojen tarkistus säilyy silti palvelimella, sillä sanastojen laajuuden vuoksi niiden siirtäminen resursein taan rajoitettuun matkapuhelimeen ei ole järkevää.

Palaute prototyypistä kerättiin sekä Facebookin ryhmätoiminnon avulla että työpajassa. Flostersia pelattiin työpajaan osallistujien toimesta noin viikon ajan. Tuona aikana en paljoa iteroinut prototyyppiä eteenpäin. Tein parannuksia lähinnä ulkoasuun, jotta prototyyppi olisi toiminut useammilla matkapuhelimien näyttöjen resoluutioilla. Paransin kuitenkin vihjesysteemin toimivuutta lähinnä niin, että saatavilla olevista tiedoista valittiin näytettäväksi satunnaisesti valittuja tietoja. Ensimmäisessä versiossa näytettiin kolme ensin löytynyttä tietoa. Palautetta pelaajilta kerättiin niin ikään Facebookin kautta, jonne olin perustanut Flosters-nimisen ryhmän. Sen keskusteluryhmään tulikin käyttäjien kommentteja ja parannusehdotuksia. Lopuksi prototyypistä keskusteltiin työpajan lopetuskokouksessa.

Prototyyppiä on testannut pistelistauksen mukaan 20 henkilöä. Useiden kommenttien mukaan prototyyppi toimi kuitenkin niin, että se laittoi pelaajan ajattelemaan ystäviään ja sitä, mitä pelaaja heistä tietää. Lisäksi yhteydenottoa löydettyihin ystäviin toivottiin. Jane Vincent [Vincent, 2006] puhuu kaverialueesta (buddy space) kertoessaan matkapuhelinten käyttäjien matkapuhelinten ympärille luomasta intiimistä ja yksityisesti tilasta. Hän jatkaa, että joka kerta kun käyttäjä käyttää puhelintaan, se herättää käyttäjässä eräänlaisen läheisyyden ja läsnäolon kavereihin, joita matkapuhelimella voi tavoittaa,

vaikka he eivät ole käyttäjän välittömässä läheisyydessä. Näin tapahtuu, vaikka käyttäjä ei suoraan olisikaan juuri sillä hetkellä kommunikaatioyhteydessä kaveriin. Mielestäni Flosters toimii samanlaisella periaatteella. Se herättää läsnäolon tunnetta poissaolevista (absent presence) ystävistä.

Muut palautteissa esille nousseet seikat koskivat pelin tasapainotusta ja asioita, jotka tunnistin ongelmakohtiksi jo etukäteen. Yksi näistä oli epätasainen ystävien määrä, jolloin satoja ystäviä omaavalla pelaajalla peli on paljon vaikeampi kuin muutamia tai korkeintaan muutamia kymmeniä omaavalle pelaajalle. Nämä kaikki parannusehdotukset liittyivät ehkä enemmänkin pelin valmiiksi saattamiseen kuin prototyypin tehtävän eli perusmekaniikan parantamiseen. Perusmekaniikka nähtiin siis olevan kunnossa ja pelaajat olisivat halunneet, että prototyypistä kehitettäisiin valmiimpi, lopullista peliä muistuttavampi ja ominaisuuksiltaan rikkaampi versio.

Merkittävin palaute ja konseptin toimivuuden osoitus oli kuitenkin se, että kukaan testipelaajista ei maininnut pelanneensa tyyllillä, jossa kirjaimista yritettiin muodostaa mitä tahansa sanoja. Kaikkien mielestä ystävien nimien löytäminen oli mielenkiintoisin tapa pelata. Tämä on erittäin vahva indikaattori sosiaalisen ja ystävyys-suhteisiin perustuvan pelaamisen vetovoimasta. Tässä mielessä konsepti ja prototyyppi onnistuivat tehtävässään osoittaen sosiaalisten aspektien esiin nostamisen olevan kiinnostava tapa mobiilille moninpelaamiselle.

Flosters-prototyypin ja -konseptin jatkokehitysideoita on monia. Yksi tapa olisi sitoa peli vielä tiukemmin matkapuhelimiin ja niiden sosiaalisuutta hyväksikäyttäväksi, koska matkapuhelimet sisältävät jo valmiina kohtuullisen laajan sosiaalisen verkoston. Verkosto löytyy matkapuhelimen osoitekirjasta, johon on talletettuna nimiä ja tietoja käyttäjän sukulaisista, kollegoista ja muista ihmisistä, joihin käyttäjä on ollut tai haluaa olla yhteydessä. Jaikun kaltaiset palvelut lisäävät osoitekirjaan presenssi-tiedon. Näkisin, että integroimalla osoitekirjan ja verkon sosiaalisen median palveluiden tietoja Flostersista voitaisiin kehittää vieläkin paremmin matkapuhelimiin sopiva ja sosiaalisuutta korostava peli. Matkapuhelimen osoitekirjan tietojen käyttö mahdollistaisi esimerkiksi sen, että ystävän nimen löydyttyä kirjaimista, peli ei käyttäisikään hyväkseen viestintää ystävälle Facebookin tai muun sosiaalisen median palvelun kautta, vaan käyttäisi esimerkiksi tekstiviestiä tai suoraa soittoa ystävälle. Näin pelin välittömyys kasvaisi, koska ystävän ei tarvitsisi ensin mennä Internetiin tarkastamaan viestejä sosiaalisen median palvelusta tai sähköpostista, vaan hän saisi viestin suoraan aina mukanaan olevaan matkapuhelimeen.

Pelin ja sen tietoisuuden levittämiseen sosiaalisen median palvelut ja sosiaaliset verkosto palvelut voivat tarjota hyviä mahdollisuuksia. Jos ystäväsi on löytänyt sinun nimesi pelin avulla ja peli tai ystävä lähettää sinulle siitä viestin, on todennäköisempää, että kokeilet peliä. Tämä seikka nousi esiin myös Nokian tutkimuksessa, jossa melkein 80 prosenttia vastaajista aikoi testata pelidemoa, jos ystävä on lähettänyt pelin, demon tai suositellut peliä [Nokia, 2006]

5.3 Mobiilipelien pelisuunnittelumallit

Pelisuunnittelumallit (game design patterns) ovat olleet yksi tapa jäsentää pelien suunnittelua ja analyysia. Alun perin Staffan Björk ja Jussi Holopainen esittelivät ne kirjassaan *Patterns in Game Design* [Björk and Holopainen, 2004]. Pelisuunnittelumallit ovat kokoelma mahdollisia suunnitteluvaihtoehtoja peleille. Pelisuunnittelumalleilla voidaan heidän mukaansa auttaa pelien suunnittelua, minkä lisäksi olemassa olevia pelejä voidaan analysoida niiden avulla. Pelisuunnittelumallit keskittyvät pelaajan interaktioon pelin sekä muiden pelaajien suhteen. Björkin ja Holopaisen mukaan pelisuunnittelumallit tarjoavat lisäksi yhteisen terminologian pelin, sen elementtien ja pelaajien vuorovaihtukselle. Pelisuunnittelumallien ja ohjelmistosuunnittelusta tuttujen mallien välillä voidaan nähdä yhtäläisyyksiä. Kumpikin niistä tarjoaa valmiita, dokumentoituja ja joihinkin tilanteisiin sopivia suunnittelumalleja. Pelisuunnittelumallit eivät sinällään kuitenkaan ratkaise ongelmia, vaan suunnittelijan on osattava päättää, milloin kunkin mallin käyttäminen on järkevintä.

Perinteisesti suunnittelumallit ovat rakentuneet ongelma-ratkaisu -ajattelulle ja tämän kuvaamiselle mallissa. Pelisuunnittelumallit eivät lähde liikkeelle tästä näkökulmasta, koska Björkin ja Holopaisen mukaan ongelmalähtökohtainen ajattelu olisi rajoittanut pelisuunnittelumallien avulla tapahtuvaa luovaa suunnittelutyötä liiaksi. Björk ja Holopainen määrittelevätkin pelisuunnittelumallit seuraavasti. Pelisuunnittelumallit ovat puoliformaaleja toisistaan riippuvia kuvauksia yleisesti toistuvista pelin interaktiota koskevista osista.

Björkin ja Holopaisen kirjassa esitellään 295 pelisuunnittelumallia. Myöhemmin Ola Davidsson, Johan Peitz ja Staffan Björk laajensivat mallit myös mobiilipeleille. He esittelevät raportissaan "Game Design Patterns for Mobile Games" [Davidsson *et al.*, 2004] 24 uutta mallia mobiilipelejä koskien sekä 50 tarkistettua mallia, jotka uudelleen muotoiltuna liittyvät erityisesti mobiilipeleihin.

Davidssonin *et al.* 24 uuden mobiilipelien suunnittelumallin joukossa ei ole

yhtään, jotka jotenkin liittyisivät mobiiliverkkojen niukkojen resurssien kiertämiseen ja pelin saattamiseen reaaliaikaisemmaksi. Joukossa on kuitenkin kaksi, joiden voi tulkita jollain tasolla ottavan huomioon matkapuhelinverkkoja. Ensimmäinen on malli numero 7: "Myöhään saapuvat pelaajat" ja toinen on numero 14: "keskeytettävyyys". Myöhään saapuvien pelaajien tukeminen liittyy aikaisemmin esittelemääni asynkroniseen pelaamiseen, jossa pelaajien ei tarvitse aloittaa eikä lopettaa pelaamista samaan aikaan. Matkapuhelin toimii akulla, ja pelkästään tämän takia pelin keskeytettävyyys on tärkeää. Lisäksi tulevat puhelut ja viestit voivat keskeyttää pelaamisen. Verkkojen näkökulmasta katvealueelle joutuminen on seikka, jonka pelin tulisi käsitellä asianmukaisesti keskeytettävyyden avulla.

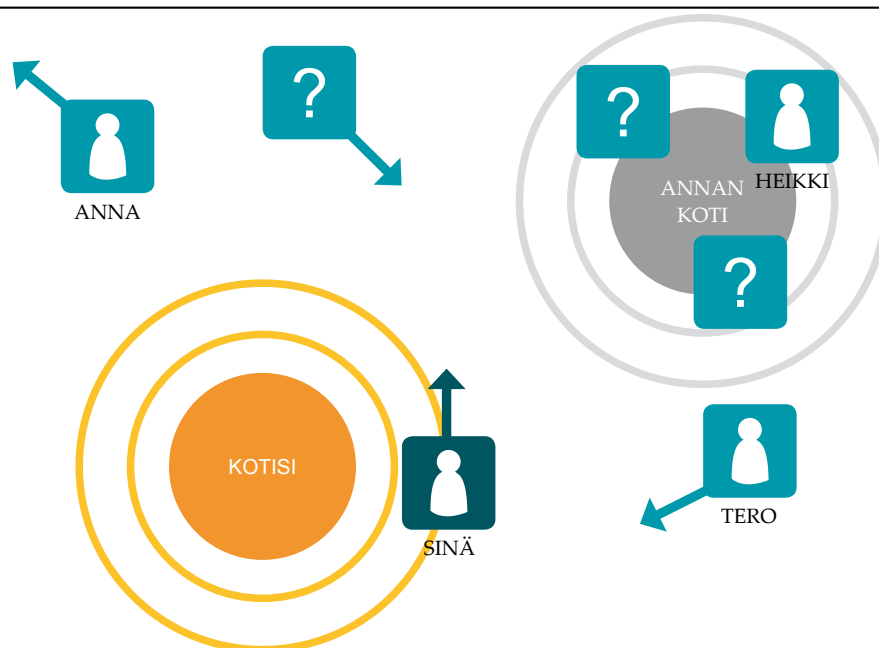
Uusien mobiilipelien suunnittelumallien joukossa on kuitenkin useita, jotka ottavat huomioon moninpelaamisen, sosiaalisuuden ja liikkumisen sosiaalisessa ympäristössä. Nämä kaikki ovat seikkoja, joita käsittelin kohdassa 2. Tarkistettujen mallien joukosta löytyvät seuraavat: 36: "Viivytetyt seuraukset", 37: "Seisokkiaika", 38: "Vuorojen ottaminen", 66: "Reaaliaikaiset pelit", 67: "Asynkroniset pelit", 68: "Synkroniset pelit" ja 70: "Kohtuulliset odotteluajat". Näiden katsoisin ottavan huomioon jollain tasolla matkapuhelinverkkojen ja reaaliaikaisuuden problematiikkaa.

Flostersissa on sisällä monia mobiilipelien suunnittelumalleissa mainittuja ominaisuuksia. Se sisältää moninpelaamiseen, sosiaalisuuteen, kommunikointiin, palkitsemiseen, pisteisiin ja tavoitteisiin liittyviä mobiilipelien suunnittelumalleja. Suurin ryhmä Flostersista puuttuvista malleista liittyy paikkatiedon hyväksikäyttöön. Kaikkien mahdollisten suunnittelumallien käyttäminen ei kuitenkaan tee pelistä parempaa. Kerroin kohdassa 2.2 Suominen muistutuksesta hänen karttansa yhteydessä, että eri alueille sijoituvista peleistä ei voi automaattisesti sanoa jonkun toisen alueen pelien olevan parempia kuin toiset.

5.4 Jatkokehityssuunnitelma

Flostersin nykyinen konsepti eikä prototyyppi käytä hyväkseen resurssien kompensointitekniikoita. Syy tähän on se, että Flostersin tässä kehitysvaiheessa minulla oli aikaa keskittyä vain peruspelimekaniikan ja sosiaalisen median palveluiden tietojen käyttämisen testaamiseen. Flostersin tulevaisuuteen kuuluu kuitenkin suunnitelma, jossa resurssirajoitusten kompensointitekniikoita tullaan käyttämään moninpelaajan mobiilipelissä.

Konseptia on tarkoitus kehittää suuntaan, jossa Flostersin pelimaailma on pysyvä. Pysyväksi pelimaailman tekee se, että pelimaailmassa on pelaajien hahmoja, jotka ovat liikkeessä ja pelimaailmassa, vaikka pelaaja ei juuri sillä hetkellä olisikaan aktiivisesti mukana pelissä. Suunnitellun pelin ideana on, että jokaisella pelaajalla on pelimaailmassa alue, johon he yrittävät saada mahdollisimman monta ystävää. Aluetta voi nimittää pelaajan kodiksi. Kuvassa 5.4 näkyy kahden pelaajan kodit. Ystävien ohjaaminen kotiin tapahtuu siten, että pelaaja valitsee ystävän, jonka jälkeen hänen tulee muodostaa annetuista kirjaimista mahdollisimman pitkä sana. Mitä pitemmän sanan pelaaja saa muodostettua, sitä isommalla voimalla vaikutetaan ystävän liikkeeseen. Voima vaikuttaa aina kohti pelaajan kotia muuttaen ystävän liikevektoria. Kun kotona on tarpeeksi ystäviä ja pelaajan oma hahmokin on siellä, järjestetään kodissa juhlat ja pelaaja saa paljon pisteitä.



Kuva 5.4: Flosters-konseptin tulevaisuus

Pelaaja voi koska tahansa vaikuttaa omaan hahmoonsa edellä kuvatulla menetelmällä. Kuvassa 5.4 Annan olisikin päästävä kotiinsa, jotta juhlat alkaisivat ja hän saisi ison pistepotin. Sen sijaan pelaajan täytyy ensin hankkia mahdollisuus vaikuttaa muihin hahmoihin. Tämä tapahtuu nykyisen Flosters-prototyypin mukaisella mekaniikalla, jossa pelaajalle annetaan kirjaimia ja pelaajan tulee tunnistaa ystävä näistä kirjaimista muodostamalla ystävän nimi.

Näin pelaaja tunnistaa ystävänsä pelimaailmasta ja voi ryhtyä ohjailemaan häntä kotiaan kohti. Kuvassa 5.4 pelaaja on tunnistanut Annan, Teron ja Heikin ja pystyy siis vaikuttamaan heidän liikkeeseensä. Sen sijaan kolme ystävää on vielä tunnistamatta. Myös pelaajien fyysinen sijainti voi vaikuttaa heidän sijainteihinsa ja liikkeeseen pelimaailmassa. Tämä sijainti ja liike voidaan paikantaa solu- tai GPS-menetelmillä. Voidaan ajatella, että vaikka pelaaja ei aktiivisesti pelaisi peliä, hänen liikkeistään voidaan ottaa syötteitä joillakin aikaväleillä, jotka vaikuttavat pelaajan kuvan liikkumiseen Flostersisten pelimaailmassa. Tällöin Flostersisten siirtyisi kuvassa 5.1 olevalla Suomen kartalla enemmän oikealle motiivis-mobiilisella akselilla ja olisi myös enemmän Järvisen määrittelemä todellinen mobiilipeli.

Koska kyseessä on kohtuullisen hitaasti ja melko deterministisesti tapahtuva liikkuminen, parhaiten sovellettava kompensatiotekniikka olisi tällöin synkronoitu simulaatio (ks. 4.2.5). Liikkeen laskenta tapahtuisi synkronisen simulaation puitteissa deterministisesti. Pelaajat vaikuttaisivat kuitenkin simulaatioon synnyttämällänsä tapahtumilla. Toinen mahdollinen vaihtoehto olisi käyttää liikkeen ennustamista (ks. 4.2.2). Tämä vaihtoehto voisi olla soveltuvampi tilanteessa, jossa pelaajia paikannettaisiin pitkin päivää paljon useammin kuin vain muutamia kertoja päivässä.

Pelaajien määrän ollessa suuri esimerkiksi tapauksissa, joissa pelaajan verkostossa on satoja ystäviä, kyseeseen voi tulla myös suodattaminen kiinnostuksen alueen mukaan (ks. 4.2.4). Tällöin Flostersisten pelaaja olisi kiinnostunut vain niistä objekteista, jotka kulloinkin olisivat lähellä hänen kotiaan, tai niistä, jotka näkyisivät ruudulla. Erittäin suuren pelimaailman kysymyksessä koko pelimaailmaa voitaisiin jakaa osiin. Osien välillä tietoa välitettäisiin vain samoissa soluissa oleville pelaajille ja tätäkin tietomäärää voitaisiin suodattaa pienemmäksi.

Flostersisten on jo tällä hetkellä osoittautunut toimivaksi ja mielenkiintoiseksi sosiaalisena mobiilipelinä. Näillä suunnitelluilla jatkokehitysideoilla peli tulisi entisestäänkin monipuolisemmaksi ja se ottaisi huomioon myös pelaajan liikkumisen tehden siitä tavallaan enemmän aidomman mobiilipelin. Tuleva kehitys ja prototyyppi näiden suunnitelmien pohjalta näyttää, mikä tai mitkä kompensatiotekniikoista toimivat matkapuhelinverkkojen päällä pelattavassa pelaajan liikkeen huomioivassa sosiaalisessa mobiilissa moninpelissä.

6 POHDINTAA JA YHTEENVETO

Tämä pro gradu -tutkielma kartoitti matkapuhelimien ja matkapuhelinverkkojen soveltuvuutta moninpelattaville mobiilipeleille. Matkapuhelimilla ja niiden käyttäjillä on erityislaatuinen suhde, joka juontaa juurensa matkapuhelimien sosiaalisesta ulottuvuudesta. Myös moninpeleissä on sosiaalinen ulottuvuus ja tämän vuoksi matkapuhelimet ja moninpelit sopivat yhteen. Matkapuhelimet aiheuttavat kuitenkin rajoituksia moninpeleille. Rajoitukset syntyvät lähinnä matkapuhelinverkkojen resurssirajoituksista. Viive on suurin yksittäinen ongelma matkapuhelinverkoissa moninpelejä ajatellen.

Kollaboratiivisten virtuaaliympäristöjen ja armeijan simulaatioiden kehittämiseksi on tutkittu verkkojen resurssirajoitusten kompensatiotekniikoita. Kompensatiotekniikoilla on yritetty kiertää resurssirajoituksia ja rajata resurssirajoitusten ilmeneminen tiettyyn paikkaan. Miten nämä kompensointitekniikat sitten soveltuvat moninpelattaville mobiilipeleille? Tutkielmani mukaan jotkin kompensatiotekniikat voivat alustavasti soveltua mobiileille moninpeleille. Joidenkin toisten kohdalla on sen sijaan selvästi nähtävissä, että niiden soveltuvuus ei täytä vaatimuksia. Alustavasti soveltuvaksi ajateltuja tekniikoita pitäisi seuraavaksi testata aidoissa matkapuhelinverkkoympäristöissä, jotta asiasta saataisiin tarkempaa tietoa.

Matkapuhelimet laitteina ovat erittäin heterogeenisiä. Matkapuhelinverkot ja langattomat verkot yleensä ovat heterogeenisiä. Tällöin pelisuunnittelulle ja toki myös toteutukselle tulee rutkasti haasteita. Miten saada aikaan yhtä pelattava ja nautittava peli sekä vaatimattomien ominaisuuksien puhelimella ja heikompia yhteyksiä hyödyntävän pelaajan sekä korkean tason puhelimella ja nopeitten verkkojen ulottuvissa olevan pelaajan välille. Pelin tulisi olla myös tasapuolinen eikä paremmasta laitteisto tai verkkoyhteydestä pitäisi saada hyötyä. Tämä pakottaa suunnittelemaan pelejä pienimmän mahdollisen yhteisen nimittäjän mukaan. Ainakin silloin, jos tavoitellaan mahdollisimman laajaa tukea eri laitteille ja verkoille.

En ole tässä tutkimuksessa ottanut huomioon 3G-verkkojen mahdollistamaa IMS-arkkitehtuuria ja SIP-protokollaa. Ne voivat kuitenkin tulevaisuudessa tarjota paremman teknologisen alustan myös mobiileille moninpeleille. Tarkoitukseni oli tarkastella nykyisiä ja vallitsevia matkapuhelinverkkoja. Koska GSM dominoi vielä selkeästi laitteita ja verkkoja, en sen takia tutkinut ISM:ä. SIP:n käyttöönotto edellyttää myös noin 20 eri asetuksen asettamista matkapuhelimessa sekä SIP-tilin tekemistä, joko operaattorille tai palve-

lun tarjoajalle [Mäkeläinen, 2006]. Tämä on eräs syistä, joka varmasti karkottaa ajanviete-pelaajat.

Kuten Flosters-konsepti ja -prototyyppi näyttää, sosiaalinen mobiilipeli nähdään mielenkiintoisena mobiilipelaamisen muotona. Flosters myös osoittaa, että yksinpelitkin voivat hyötyä matkapuhelinverkkojen ja niiden tarjoaman tiedonsiirron käyttämisestä. Mielestäni pitäisi hyödyntää sekä pelisuunnittelua että kompensatiotekniikoita yhdessä, jotta mobiileista moninpeleistä saataisiin pelaajien toiveet täyttävät saavuttamalla verkkojen mahdollistama reaaliaikaisuus.

LÄHDELUETTELO

- [Aggarwal *et al.*, 2004] Sudhir Aggarwal, Hemant Banavar, Amit Khandelwal, Sarit Mukherjee, and Sampath Rangarajan. Accuracy in Dead-Reckoning Based Distributed Multi-Player Games. In *Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games*, pages 161–165, Portland, Oregon, USA, 2004. ACM.
- [Ahmad, 2005] Aftab Ahmad. *Wireless and Mobile Data Networks*. John Wiley & Sons, 2005.
- [Akkawi *et al.*, 2004] Amjad Akkawi, Sibylle Schaller, Oliver Wellnitz, and Lars Wolf. *Entertainment Computing - ICEC 2004*, chapter Networked Mobile Gaming for 3G-Networks, pages 457–467. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Eindhoven, The Netherlands, 2004.
- [Armitage and Stewart, 2004] Grenville Armitage and Lawrence Stewart. Limitations of Using Real-World, Public Servers to Estimate Jitter Tolerance of First Person Shooter Games. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 257–262, Singapore, 2004. ACM.
- [Ballard, 2007] Barbara Ballard. *Designing the Mobile User Experience*. John Wiley & Sons, 2007.
- [Beigbeder *et al.*, 2004] Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, and Mark Claypool. The Effects of Loss and Latency on User Performance in Unreal Tournament 2003. In *Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games*, pages 144–151, Portland, Oregon, USA, 2004. ACM.
- [Björk and Holopainen, 2004] Staffan Björk and Jussi Holopainen. *Patterns in Game Design*. Charles River Media, 2004.
- [Bogost, 2004] Ian Bogost. Asynchronous Multiplay: Futures for Casual Multiplayer Experience. In *Proceedings of Other Players Conference 2004*, Copenhagen, Denmark, December 2004.

- [Boukerche, 2006] Azzedine Boukerche. *Handbook of Algorithms for Wireless Networking and Mobile Computing*. CRC Press, 2006.
- [Chen and Kotz, 2000] Guanling Chen and David Kotz. *A Survey of Context-Aware Mobile Computing Research*, November 2000.
- [Claypool *et al.*, 2003] Mark Claypool, David LaPoint, and Josh Winslow. Network Analysis of Counter-strike and Starcraft. In *22nd IEEE International Performance, Computing, and Communications Conference*, Phoenix, Arizona, USA, April 2003. IPCC.
- [Davidsson *et al.*, 2004] Ola Davidsson, Johan Peitz, and Staffan Björk. *Game Design Patterns for Mobile Games*, 2004.
- [Derksen *et al.*, 2006] Jan Derksen, Robert Jansen, Markku Maijala, and Erik Westerberg. HSDPA Performance and Evolution. *Ericsson Review*, 82:117–120, 2006.
- [Dovey and Kennedy, 2006] Jon Dovey and Helen W. Kennedy. *Game Cultures: Computer Games as New Media*. Open University Press, 2006.
- [Elliott, 2008] Phil Elliott. *Mobile Game Downloads Continue Slow Growth*, 2008. Viitattu 24.3.2008, <http://www.gamesindustry.biz/articles/mobile-game-downloads-continue-slow-growth>
- [Elverdam and Aarseth, 2007] Christian Elverdam and Espen Aarseth. Game Classification and Game Design: Construction Through Critical Analysis. *Games and Culture*, 2:3–22, 2007.
- [Evensen *et al.*, 2008] K. R. Evensen, A. Petlund, C. Griwodz, and P. Halvorsen. Redundant Bundling in TCP to Reduce Perceived Latency for Time-Dependent Thin Streams. *IEEE Communications Letters*, 4:334–336, 2008.
- [Fleishman, 2005] Glenn Fleishman. *Wi-Fi Stands for...Nothing (and Everything)*, November 2005. Viitattu 1.3.2008, <http://wifinetnews.com/archives/006029.html>
- [Friedl, 2002] Markus Friedl. *Online Game Interactivity Theory*. Charles River Media, October 2002.

- [Gartner, 2008a] Gartner. *Gartner Says Worldwide Mobile Phone Sales Increased 16 Per Cent in 2007*, February 2008. Viitattu 26.3.2008, <http://www.gartner.com/it/page.jsp?id=612207>
- [Gartner, 2008b] Gartner. *Gartner Says Worldwide PC Shipments to Grow 11 Percent in 2008, Market Could Fall Victim to Weaker Global Economy*, March 2008. Viitattu 26.3.2008, <http://www.gartner.com/it/page.jsp?id=631107>
- [Gibson, 2006a] Bruce Gibson. *Whitepaper Mobile Fun & Games III*. Juniper Research Ltd, 2006.
- [Gibson, 2006b] Ellie Gibson. *Mobiles will overtake consoles as mass market grows - I-play*, December 2006. Viitattu 24.3.2008, <http://www.gamesindustry.biz/articles/mobiles-will-overtake-consoles-as-mass-market-grows-i-play>
- [Google, 2008] Google. *Android - An Open Handset Alliance Project*, 2008. Viitattu 5.4.2008, <http://code.google.com/android/>
- [Grüter and Oks, 2007] Barbara Grüter and Miriam Oks. *Situated Play and Mobile Gaming*. In *Proceedings of DiGRA 2007 Situated Play*, Tokyo, Japan, 2007. The University of Tokyo.
- [GSM Association, 2007] GSM Association. *GSM - The Wireless Evolution*, 2007. Viitattu 14.3.2008, <http://www.gsmworld.com/technology/index.shtml>
- [Henderson, 2003] Tristan Nicholas Hoang Henderson. *The Effects of Relative Delay in Networked Games*. Dissertation, Department of Computer Science, University College London, February 2003.
- [Hentunen, 2007] Heidi Hentunen. *Internetyhteyden käyttöönotto GPRS:n avulla käyttäjän näkökulmasta*. Pro gradu, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos, Vuorovaikutteinen teknologia, May 2007. 14.4.2008. Viitattu 14.4.2008, http://www.cs.uta.fi/research/theses/masters/Hentunen_Heidi.pdf
- [Holden, 2007] Windsor Holden. *Mobiles... Playing the Game*, November 2007.

- [Holma *et al.*, 2006] Harri Holma, Antti Toskala, and Harri Holma. *HSD-PA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications*. John Wiley, 2006.
- [Huang, 2006] Jiung-yao Huang. *Chapter 7. Resource Management for Scalability and Performance*, 2006. Viitattu 20.5.2008, <http://web.ntpu.edu.tw/~jyhuang/Course/PDS/Chapter07.pdf>
- [IDC, 2007] IDC. *IDC Finds Slower Growth in the Mobile Phone Market in 2007 While Samsung Captures the Number Two Position For the Year*, 2007. Viitattu 26.3.2008, <http://www.idc.com/getdoc.jsp?containerId=prUS21053908>
- [IDC, 2008a] IDC. *IDC Releases Worldwide Mobile Phone, PC and x86 Server Supply Chains Predictions for 2008*, February 2008. Viitattu 26.3.2008, <http://www.idc.com/getdoc.jsp?containerId=prUS21099108>
- [IDC, 2008b] IDC. *PC Market Is Expected To Continue Double-Digit Growth Despite Increasing Economic Concerns, According to IDC*, March 2008. Viitattu 26.3.2008, <http://www.idc.com/getdoc.jsp?containerId=prUS21138308>
- [Järvinen, 2002] Aki Järvinen. *Milloin pelistä tulee mobiili?*, 2002. Viitattu 25.2.2008, <http://www.m-cult.net/mediumi/article.html?id=29>
- [Jehaes *et al.*, 2003] Tom Jehaes, Danny De Vleeschauwer, Toon Coppens, Bart Van Doorselaer, Eva Deckers, W. Naudts, K. Spruyt, and R. Smets. *Access Network Delay in Networked Games*. In *Proceedings of the 2nd Workshop on Network and System Support for Games*, pages 63–71, Redwood City, California, 2003. ACM.
- [Jүүл, 2005] Jesper Juul. *Half-Real: Video Games between Real Rules and Fictional Worlds*. MIT Press, 2005.
- [Kallio *et al.*, 2007] Kirsi Pauliina Kallio, Kirsikka Kaipainen, and Frans Mäyrä. *Gaming Nation? Piloting the International Study of Games Cultures in Finland*. Tampereen yliopisto, Hypermedialaboratorio, Tampere, 2007.
- [Kasesniemi and Rautiainen, 2001] Eija-Liisa Kasesniemi and Pirjo Rautiainen. *Kännyssä piilevät sanomat - Nuoret, väline ja viesti*. Tampere University Press, Vammala, 2001.

- [Kerr, 2006] Aphra Kerr. *The Business and Culture of Digital Games: Gamework/gameplay*. Pine Forge Press, 2006.
- [Kerry, 2008] Stuart J. Kerry. *Official IEEE 802.11 Working Group Project Timelines - 03/03/08*, March 2008. Viitattu 1.3.2008, http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm
- [Koster, 2002] Raph Koster. *Online World Timeline*, February 2002. Viitattu 1.4.2008, <http://www.raphkoster.com/gaming/mudtimeline.shtml>
- [Kreher and Rüdebusch, 2007] Ralf Kreher and Torsten Rüdebusch. *UMTS Signaling: UMTS Interfaces, Protocols, Message Flows and Procedures*. John Wiley & Sons, second edition, 2007.
- [Kuittinen *et al.*, 2007] Jussi Kuittinen, Annakaisa Kultima, Johannes Niemelä, and Janne Paavilainen. Casual Games Discussion. In *Proceedings of the 2007 Conference on Future Play*, pages 105–112, Toronto, Canada, 2007. ACM.
- [Lagerström, 2008] Samu Lagerström. *Viestintävirasto Telepalveluiden käyttötutkimus 2007*, 2008. Viitattu 16.3.2008, http://www.ficora.fi/attachments/suomi_R_Y/5vnqZFclE/Files/CurrentFile/Tutkimusraportti2007Telepalveluidenkayttotutkimus.pdf
- [Laurio, 2007] Juha-Matti Laurio. *Lehti: Göran Persson sisäisti salakuuntelun vaarat*, March 2007. Viitattu 25.2.2008, <http://www.tietoviikko.fi/doc.te?fid=1143684>
- [Lee, 2006] Youngseok Lee. Measured TCP Performance in CDMA 1x EV-DO Network. In *Proceedings of the Passive and Active Measurement Conference 2006*, Adelaide, Australia, April 2006.
- [Leidenius, 2008] Kim Leidenius. Kännykkä vanhenee kolmessa vuodessa. *Tietokone*, 27:14, March 2008.
- [Linnake, 2008] Tuomas Linnake. *Matkapuhelinten analogiaika päättyy Yhdysvalloissa*, 2008. Viitattu 25.2.2008, http://www.itviikko.fi/page.php?page_id=46&news_id=20084944

- [Majewski *et al.*, 2006] C. Majewski, C. Griwodz, and P. Halvorsen. Translating Latency Requirements into Resource Requirements for Game Traffic. In Steven Furnell and Paul Dowland, editors, *Proceedings of the International Network Conference (INC 2006)*, pages 113–120. University of Plymouth, 2006.
- [Mäkeläinen, 2006] Sami Mäkeläinen. *Service-based Requirements for Future Mobile Networks*. Master's thesis, University of Helsinki, Faculty of Science, Department of Computer Science, August 2006.
- [Manninen, 2007] Tony Manninen. *Pelisuunnittelijan käsikirja - ideasta eteenpäin*. Kustannus Oy Rajalla, Tallinna, Viro, 2007.
- [MarketResearch.com, 2006] MarketResearch.com. *Mobile Games ERP : Overview*, April 2006. Viitattu 15.10.2007, <http://www.marketresearch.com/product/display.asp?productid=1281002&xs=r&SID=76889477-397267498-322498479&curr=USD&kw=&view=toc>
- [MarketResearch.com, 2008] MarketResearch.com. *2008 Global Mobile Communications - Statistics, Trends and Forecasts*, February 2008. Viitattu 24.3.2008, <http://www.marketresearch.com/product/display.asp?productid=1687234&xs=r&SID=57350582-411140328-468359518&curr=USD&kw=&view=abs>
- [Marshall *et al.*, 2004] D. Marshall, D. Delaney, S. McLoone, and T. Ward. *Challenges in modern Distributed Interactive Application design*. Technical Reports - 2004. National University of Ireland Department of Computer Science, Maynooth, 2004.
- [Mäyrä, 2008] Frans Mäyrä. *An Introduction to Game Studies: Games in Culture*. Sage, 2008.
- [Mikkonen, 2004] Tommi Mikkonen. *Mobiiliohjelmointi*. Valikko. Talentum, Jyväskylä, second edition, August 2004.
- [Moon *et al.*, 2006] Kyung-Seob Moon, Vallipuram Muthukkumarasamy, and Anne T. Nguyen. Reducing Network Latency on Consistency Maintenance

- Algorithms in Distributed Network Games. In *Proceedings of the IADIS International Conference: Applied Computing 2006*, pages 137–144, San Sebastian, Spain, 2006. IADIS Press.
- [Newman and Simons, 2004] James Newman and Iain Simons. *Difficult Questions About Video Games*. Suppose Partners, 2004.
- [Nichols and Claypool, 2004] James Nichols and Mark Claypool. The Effects of Latency on Online Madden NFL Football. In *Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video*, pages 146–151, Cork, Ireland, 2004. ACM.
- [Nokia, 2006] Nokia. *Evolution of Mobile Gaming - Exploring worldwide mobile gaming habits*. Nokia, 2006.
- [Olsder, 2007] Arjan Olsder. *Online Casual Gamers Twice As Likely To Play Mobile Games*, October 2007. Viitattu 7.4.2008, http://mobilegames.blogs.com/mobile_games_blog/2007/10/online-casual-g.html
- [Open Handset Alliance, 2007] Open Handset Alliance. *Android*, 2007. Viitattu 5.4.2008, http://www.openhandsetalliance.com/android_overview.html
- [Pantel and Wolf, 2002a] Lothar Pantel and Lars C. Wolf. On the Impact of Delay on Real-Time Multiplayer Games. In *Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 23–29, Miami, Florida, USA, 2002. ACM.
- [Pantel and Wolf, 2002b] Lothar Pantel and Lars C. Wolf. On the Suitability of Dead Reckoning Schemes for Games. In *Proceedings of the 1st workshop on Network and System Support for Games*, pages 79–84, Bruanschweig, Germany, 2002. ACM.
- [Parks Associates, 2007] Parks Associates. *Mobile Phone Remains a Casual Gaming Platform*, December 2007. Viitattu 24.3.2008, http://newsroom.parksassociates.com/article_display.cfm?article_id=4999
- [Pellegrino and Dovrolis, 2003] Joseph D. Pellegrino and Constantinos Dovrolis. Bandwidth Requirement and State Consistency in Three Multiplayer

- Game Architectures. In *Proceedings of the 2nd Workshop on Network and System Support for Games*, pages 52–59, Redwood City, California, 2003. ACM.
- [Pietarinen, 2008] Harri Pietarinen. *Ericsson: Wlan-hotspotit häviävät kuin puhe-linkopit*, March 2008. Viitattu 16.3.2008, <http://www.tekniikkatalous.fi/ict/article63591.ece>
- [Pulse, 2006] Pulse. *What is GSM? - Global System for Mobile Communications*, 2006. Viitattu 7.4.2008, http://www.pulsewan.com/data101/gsm_basics.htm
- [Rapeli, 2008] Katja Rapeli. *Symbian-toimitusten kasvu kiihtyi edelleen*, November 2008. Viitattu 24.2.2008, http://www.tietoviikko.fi/doc.te?f_id=1258110&s=r
- [Salen and Zimmerman, 2003] Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. The MIT Press, 2003.
- [Schaefer *et al.*, 2002] Christian Schaefer, Thomas Enderes, Hartmut Ritter, and Marina Zitterbart. Subjective Quality Assessment for Multiplayer Real-Time Games. In *Proceedings of the 1st Workshop on Network and System Support for Games*, pages 74–78, Braunschweig, Germany, 2002. ACM.
- [Sheldon *et al.*, 2003] Nathan Sheldon, Eric Girard, Seth Borg, Mark Claypool, and Emmanuel Agu. The Effect of Latency on User Performance in Warcraft III. In *Proceedings of the 2nd Workshop on Network and System Support for Games*, pages 3–14, Redwood City, California, 2003. ACM.
- [SIG, 2008] Bluetooth SIG. *Bluetooth.com | Compare with Other Technologies*, 2008. Viitattu 1.3.2008, <http://www.bluetooth.com/Bluetooth/Technology/Works/Compare/>
- [Siitonen, 2007] Marko Siitonen. *Social Interaction in Online Multiplayer Communities*. Dissertation, University of Jyväskylä, September 2007.
- [Smed and Hakonen, 2003] Jouni Smed and Harri Hakonen. *Towards a Definition of a Computer Game*, September 2003.

- [Smed and Hakonen, 2006] Jouni Smed and Harri Hakonen. *Algorithms and Networking for Computer Games*. John Wiley & Sons, Chippenham, Wiltshire, Great Britain, April 2006.
- [Smed, 2005] Jouni Smed. *Compensatory Techniques. Consistency and Responsiveness. Scalability.*, 2005. Viitattu 26.3.2008, http://staff.cs.utu.fi/~jounsmmed/mcg_05/slides/slides051115.pdf
- [Spohn, 2005] Dave Spohn. *The History of Online Gaming 1969 - 1990*, November 2005. Viitattu 1.4.2008, <http://internetgames.about.com/od/gamingnews/a/timeline.htm>
- [STT, 2008] STT. *Suomessa vuodenvaihteessa 5 300 484 asukasta*, March 2008. Viitattu 16.3.2008, <http://www.hs.fi/kotimaa/artikkeli/Suomessa+vuodenvaihteessa+5+300+484+asukasta/1135234718462>
- [Suominen, 2003] Jaakko Suominen. *Liikkeelle vai liikkeestä pois? - Mobiilipelejä luokittelemassa*, 2003. Viitattu 25.2.2008, http://www.widerscreen.fi/2003/2-3/liikkeelle_vai_liikkeesta_pois_mobiilipeleja_luokittelemassa.htm
- [Tercek, 2007a] Robert Tercek. *1997 - 2007: The First Decade of Mobile Games*, March 2007. Viitattu 18.3.2008, <http://www.roberttercek.com/resources/Decade+of+Mobile+Games+Part+1+GDCM07+Tercek.pdf>
- [Tercek, 2007b] Robert Tercek. *1997 - 2007: The First Decade of Mobile Games Part Two*, March 2007. Viitattu 18.3.2008, <http://www.roberttercek.com/resources/Decade+of+Mobile+Games+PART+TWO+GDCM07+Tercek.pdf>
- [The RFID Centre, 2006] The RFID Centre. *Wireless Standards*, 2006. Viitattu 26.2.2008, http://www.rfidc.com/docs/introductiontowireless_standards.htm
- [Turpeinen *et al.*, 2005] Marko Turpeinen, Risto Sarvas, and Fernando Herrera. *It is a Phone Not a Console!* In *GDC Mobile White Paper*, San Francisco, CA, USA, 2005. Game Developers Conference.
- [VG Chartz, 2008] VG Chartz. *VGChartz Hardware data for the period 22nd Apr 1989 to 22nd Mar 2008*, March 2008. Viitattu 26.3.2008,

<http://vgchartz.com/hwtable.php?cons%5B%5D=Wii\&cons%5B%5D=PS3\&cons%5B%5D=X360\&cons%5B%5D=PSP\&cons%5B%5D=DS\&cons%5B%5D=XB\&cons%5B%5D=GC\&cons%5B%5D=GBA\&cons%5B%5D=PS2\&cons%5B%5D=PS\&cons%5B%5D=GB\®%5B%5D=Total\&start=32621\&end=39530>

[Viestintävirasto, 2007] Viestintävirasto. *Markkinakatsaus*, March 2007. Viitattu 16.3.2008, http://www.ficora.fi/attachments/suomi_M_Q/5tI1cWfGK/Files/CurrentFile/Markkinakatsaus_3_2007.pdf

[Vihinen, 2007] Janne Vihinen. *Supply and Demand Perspectives on Mobile Products and Content Services*. Dissertation, Helsinki School of Economics, November 2007.

[Vincent *et al.*, 2005] Jane Vincent, Leslie Haddon, and Lynne Hamill. The Influence of Mobile Phone Users on the Design of 3G Products and Services. *The Journal of The Communications Network*, 4:69–73, December 2005.

[Vincent, 2005] Jane Vincent. *Mobile World Past, Present and Future*, chapter Emotional Attachment to Mobile Phones: An Extraordinary Relationship, pages 93–104. Computer Supported Cooperative Work. Springer London, 2005.

[Vincent, 2006] Jane Vincent. Emotional Attachment and Mobile Phones. *Knowledge, Technology, and Policy*, 19:39–44, March 2006.

[Wibree, 2007] Wibree. *Wibree forum merges with bluetooth SIG*, June 2007. Viitattu 1.3.2008, <http://www.wibree.com/press/>

[Wikipedia, 2006] Wikipedia. *Structure of a GSM network (key elements)*, 2006.

[Wikipedia, 2008a] Wikipedia. 4G, 2008. Viitattu 7.4.2008, <http://fi.wikipedia.org/wiki/4G>

[Wikipedia, 2008b] Wikipedia. *UMTS-TDD*, 2008. Viitattu 7.4.2008, <http://en.wikipedia.org/wiki/UMTS-TDD>

[Wolf, 2002] Mark J. P. Wolf. *The Medium of the Video Game*. University of Texas Press, 2002.

- [Zagal *et al.*, 2000] José Pablo Zagal, Miguel Nussbaum, and Ricardo Rosas. A Model to Support the Design of Multiplayer Games. *Presence: Teleoper. Virtual Environ.*, 9:448–462, 2000.
- [Zander and Armitage, 2004] Sebastian Zander and Grenville Armitage. Empirically Measuring the QoS Sensitivity of Interactive Online Game Players. In *Australian Telecommunication Networks and Applications Conference (ATNAC 2004)*, Sydney, Australia, December 2004. ATNAC.
- [Zander *et al.*, 2005] Sebastian Zander, Ian Leeder, and Grenville Armitage. Achieving Fairness in Multiplayer Network Games Through Automated Latency Balancing. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 117–124, Valencia, Spain, 2005. ACM. 16.3.2008. Viitattu 16.3.2008, <http://portal.acm.org/citation.cfm?id=1178493&coll=Portal&dl=ACM&CFID=59535345&CFTOKEN=40775342&ret=1#Fulltext>
- [Zyda, 2005a] Michael Zyda. *Chapter Five - Managing Dynamic Shared State*, September 2005. Viitattu 16.3.2008, <http://gamepipe.usc.edu/~zyda/courses/ZydaNetworkedGamesFall2005/Lecture8-22Sept2005.ppt>
- [Zyda, 2005b] Michael Zyda. *Chapter Seven - Resource Management for Scalability and Performance*, October 2005. Viitattu 16.3.2008, <http://gamepipe.usc.edu/~zyda/courses/ZydaNetworkedGamesFall2005/Lecture13-20Oct2005.ppt>
- [Zyda, 2005c] Michael Zyda. *How to Build A Networked Game*, October 2005. Viitattu 16.3.2008, <http://gamepipe.usc.edu/~zyda/presentations/HowToBuildANetworkedGame/HowToBuildANetworkedGame.ppt>
- [Zyda, 2005d] Michael Zyda. *The Origin of Networked Games*, August 2005. Viitattu 16.3.2008, <http://gamepipe.usc.edu/~zyda/courses/ZydaNetworkedGamesFall2005/Lecture3-30Aug2005.ppt>