

## Problemi sinhronizacije u distribuiranim real-time grafičkim sistemima

---

*Pri implementaciji distribuiranih real-time grafičkih sistema, javljaju se problemi sinhronizacije pri određivanju položaja objekata. Pravilnim projektovanjem sistema, ovi problemi se mogu smanjiti do nivoa na kome greške u položaju objekata ne utiču na subjektivni doživljaj korisnika. Razmatraju se alternative u projektovanju servera i klijenata za distribuirane real-time grafičke sisteme, sa posebnim akcentom na višekorisničkim video-igramama, kao i praktična iskustva u implementaciji višekorisničke video igre na datim sistemima. Za platformu na kojoj se radi uzeta je mreža UNIX i Windows mašina, s tim da se Windows mašine koriste samo kao X serveri. U ovom trenutku, projekat je u početnoj fazi razvoja.*

**Ključne reči:** Sinhronizacija, distribuirani sistemi, X Window System

---

### Model sistema

U svom osnovnom obliku, ideja je da se sistem sastoji od većeg broja radnih mesta koja se nalaze u istom virtuelnom prostoru, pri čemu svako radno mesto simulira neko vozilo ili letelicu. Suština je da su korisnici povezani medijumom čiji je propusni kapacitet ograničen. Zbog toga postoji latencija u prenosu podataka između klijenata, te se javljaju problemi sinhronizacije.

Postoje dva osnovna pravca realizacije ove ideje. Prvi je da se uvede server, preko koga klijenti komuniciraju jedni s drugima, a drugi je da se, s obzirom da se koristi X Window, napravi samo jedan program, koji kontroliše veći broj X display-a. Druga koncepcija pri povećanju broja klijenata vrlo brzo dostiže ograničenje propusnog opsega, s obzirom na veliki broj slika koje je potrebno iscrtati i zahtevnost X protokola. Na manjem broju klijenata, ova koncepcija ima prednosti jer ne postoje nikakvi problemi sinhronizacije.

S obzirom da smo mi želeli da radimo s većim brojem radnih mesta, prva koncepcija je bila mnogo interesantnija. Da bi se ona realizovala,

---

*Veljko Krunic (1974),  
Beograd, Milana  
Rakića 79/11, student  
ETF u Beogradu  
vkrunic@phoenix.  
psc.ac.yu*

*Boris Dragovic  
(1979), Beograd,  
Vlajkovićeva 4, učenik  
3. razreda V  
beogradske gimnazije  
lynx@phoenix.psc.  
ac.yu*

morao se razraditi model klijenata i servera. Pri tome uvodimo da u daljnjem tekstu jedan klijent odgovara jednom radnom mestu.

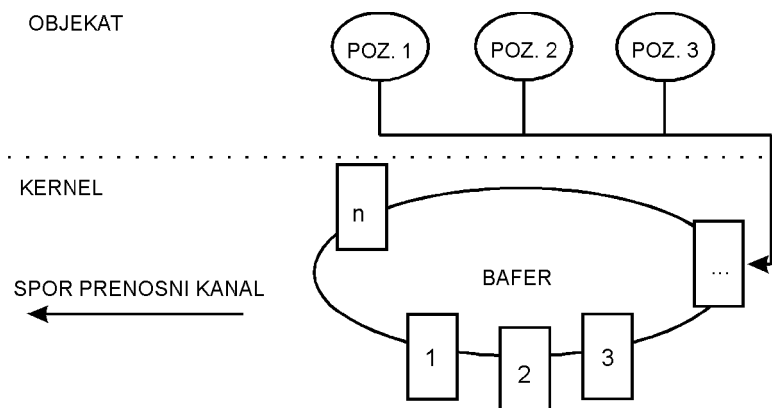
Takođe, uvodimo i pojam autoriteta na sledeći način: *klijent A ima autoritet nad objektom O ako u slučaju neslaganja pozicije objekta O na klijentima A i B, smatramo da je pozicija na klijentu A tačna.*

## Koncepcije klijenata i servera

Prvu koncepciju ćemo nazvati prosti distributivni server: ideja je da klijenti sami vode računa o kompletnom virtuelnom svetu, i da se podaci o objektima za koje dati klijent ima autoritet preko servera distribuiraju ostalim klijentima. U ovom slučaju, server je u ulozi posrednika koji poruku primljenu od jednog klijenta prosleđuje svim ostalim. Moguća je i koncepcija u kojoj server uopšte ne postoji, već sami klijenti distribuiraju podatke jedni drugima.

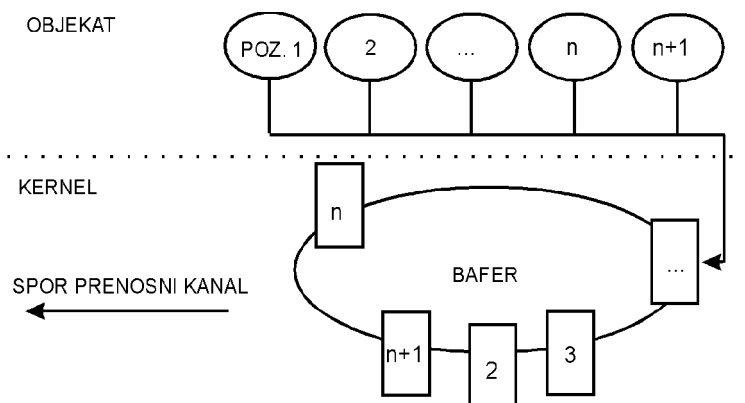
Ova koncepcija će iskazati svoje slabosti čim se dogodi da podaci, zbog problema na komunikacionoj liniji, stignu sa zakašnjenjem do nekog klijenta. S obzirom da ne bi dobio podatke o novim položajima klijenata, on bi neminovno izgubio sinhronizaciju. Slično tome, ako bi neki klijent imao problema pri slanju paketa, ostali klijenti ne bi dobili ažurne podatke o njegovom položaju. Treba imati u vidu da se ovakvi zastoji događaju relativno često, i da bi oni stvorili neprihvatljive probleme pri animaciji. Takođe, za glatku animaciju se zahteva protok velike količine podataka, što pri velikom broju klijenata dovodi do zagušenja.

Pogledajmo na primeru jedan od načina na koji se manifestuje gubljenje sinhronizacije: neka se objekat A nalazi ispred objekta B, i neka svaki ima autoritet nad svojom pozicijom. Pretpostavimo da se objekti



Slika 1.  
Proces slanja podataka o pozicijama objekata kroz kružni buffer.

Figure 1.  
Sending of object positions data through queue.



Slika 2.  
Situacija u kojoj je došlo do prekoračenja kružnog buffer-a i do prepisivanja neažurnih podataka

Figure 2.  
Queue overrun situation. Inaccurate data are replaced.

kreću u istom pravcu, smeru i istom brzinom i da klijent A upravlja objektom A, a klijent B objektom B. Najjednostavnija koncepcija je da klijent A ima autoritet nad pozicijom objekta A, a da čeka od B podatke o položaju B. Međutim, s obzirom da ovi podaci kasne, objekat A čita poziciju objekta B od pre nekoliko trenutaka, dok svoju poziciju ažurira korektno. S obzirom na ograničen kapacitet kanala za prenos, objekat A ne dobija nove pozicije objekta B istom brzinom kojom ih B generiše. Tako se događa da objekat A „misli“ da je brži od objekta B, odnosno da mu, uslovno rečeno, beži. U isto vreme, objekat B „misli“ da stiže objekat A!

Pokazuje se da se situacija pogoršava u slučaju većih izlaznih bafera ka mreži, jer se u tom slučaju veliki broj paketa baferiše i čeka na slanje, stvarajući opisani efekat (slika 1). S druge strane, ako je bafer podešen da u slučaju prepunjavanja zanemari prethodni sadržaj, biće odbačene sve prethodne pozicije objekata u baferu. S obzirom da je došlo do prepunjavanja bafera, očigledno je da pozicije nisu čitane duže vreme, pa su samim tim neažurne (slika 2). S obzirom da se neažurni podaci zanemaruju u korist ažurnijih, vrši se korekcija.

U skladu sa prethodnim, predlažemo da se smanji veličina izlaznog bafera, što izaziva gubljenje paketa, čime se praktično vrši korekcija pozicije bez interpolacije. Ovo rešenje je vrlo jednostavno za implementaciju, ali izaziva diskontinuitet u kretanju objekata.

Zato je bolja koncepcija po kojoj klijenti imaju slobodu da rade predikciju budućeg položaja objekata. Pri ovome se postavlja pitanje ko ima autoritet nad njihovom pozicijom, kao i na koji način se vrši korekcija pozicije. Predikcija se vrši na osnovu dotadašnje tendencije njegovog kretanja. Ukoliko se dogodi da se procena položaja objekta bitno razlikuje od prave pozicije (što se utvrđuje kada se dobije sledeći podatak o pravog

poziciji od servera) korekcija se mora vršiti postepeno da ne bi došlo do diskontinuiteta u kretanju objekta. Ovaj problem nije trivijalan za rešavanje, a njegovo ignorisanje može dovesti do čitavog niza problema sa sinhronizacijom.

Ova koncepcija ima i tu prednost da je saobraćaj na mreži lakše redukovati, a moguće je i uvesti prioritete pri korekciji položaja objekata (češće se koriguju oni za koje je procenjeno da imaju veći uticaj na subjektivni doživljaj greške). Jedna od varijanti postavljanja prioriteta bila bi ona po kojoj bi bliži objekti imali najviši prioritet (jer su greške kod njih najprimetnije za klijenta) dok bi dalji objekti imali sve manje prioritete kao i objekti koji nisu u vidnom polju (podaci o njima su nam potrebni samo povremeno radi promene slike na radaru u našem slučaju). Jedan od metoda za kvantifikaciju greške je opisan kod Habermajera (Habermeyer 1997). O prioritetima objekata bi morao da vodi računa server. Razmatrane su i mogućnosti da se, bazirano na ovoj tehnici, smanji veličina saobraćaja na mreži, kao i koncepcija adaptivne predikcije, u zavisnosti od očekivanog kašnjenja podataka kroz mrežu.

Pri ovome se postavlja pitanje ko je autoritativan za određivanje pozicije objekata. Jedna koncepcija opisana takođe kod Habermajera (ibid.), je takozvana koncepcija autoritativnog servera sa predikcijom klijenata. Ideja je da klijenti rade predikciju, i da svoj položaj dobijaju od servera. Server je autoritativnog tipa, odnosno stanje na njemu je garantovano validno, i u slučaju da se ne slaže sa stanjem na klijentu, stanje na serveru ima prednost. Klijenti rade predikciju kretanja svih objekata, i kada stigne ispravka sa servera, rade postepenu korekciju položaja objekta.

Nažalost, kod koncepcije autoritativnog servera se javljaju problemi pri povećanju složenosti fizičkog modela kretanja objekata, jer se vrlo brzo dolazi do granica mogućnosti servera. Kod Habermajera (ibid.) predloženo je da se to reši distribucijom servera na više mašina povezanih mrežom velike brzine, ali to nije rešenje za komplikovane modele, jer se javlja velika nesimetričnost između iskorišćenosti resursa na klijentima i serveru. S druge strane, problemi distribucije na serveru bi bili slični problemima distribucije među klijentima koji bi se donekle mogli poboljšati time što bi serveri bili na posebnom segmentu mreže velike brzine rezervisanom isključivo za njihovu komunikaciju, usled čega bi se problemi sa gubljenjem sinhronizacije javljali ređe, ali ne bi bili potpuno isključeni.

Naše rešenje je kompromisno, sa punim autoritetom klijenta nad pozicijom objekta koji simulira, i jednostavnim predikcijama položaja ostalih objekata. Podaci o položaju klijenta bi se distribuirali serveru. Sve prethodno razmatrane optimizacione tehnike bi se mogle primeniti, a smatramo da bi se dobili bolji rezultati od primene jednostavnijeg fizičkog modela, bar zbog toga što bi svaki korisnik imao bolji utisak realnosti.

## Trenutni status projekta

Mada je naš projekat tek u ranoj radnoj fazi, u ovom trenutku je naj-sličniji višekorisničkoj verziji igre „Elita“. U ovom trenutku, postoji mogućnost da se dva igrača kreću po svemiru.

Projekat je realizovan na operativnom sistemu Linux, pod X Window sistemom. Kijenti su obavezno UNIX mašine (u našoj implementaciji Linux). U ovoj verziji smo se odlučili za koncept prostog distributivnog servera bez posebnih napora za smanjenje količine saobraćaja na mreži, sa smanjenom veličinom bafera za prijem i slanje poruka. Pokazalo se da ovo jednostavno rešenje na malom broju klijenata može rešiti problem, ali izaziva pogoršanje performansi mreže, i zbog toga nije primenljivo na veliki broj klijenata. Ovaj nedostatak će u budućnosti biti ispravljen uvođenjem kompresije i prioriteta za prenos podataka. Zbog krajnjih ciljeva celog projekta, neće se ići na autoritativni server, mada ćemo razmotriti mogućnosti da server dobije jedan deo autoriteta (klijenti imaju pun autoritet nad svojom pozicijom, ali server ima autoritet nad pozicijom svih ostalih objekata). Razlog za ovo je što se u daljoj fazi razvoja može ići na prilično računski intenzivan fizički model kretanja objekata.

Grafički podsistem je u ovom trenutku realizovan tehnikom *back-face culling*-a, što ograničava trenutne objekte na konveksne. U sledećoj verziji ovo ograničenje će biti otklonjeno prelaskom na *BSP tree*. Takođe, biće uvedeni i sprajtovi, u cilju boljeg prikaza eksplozija. Pri razvoju su, korišćene standardne nehomogene koordinate tačaka, sa transformacionim matricama u razvijenom obliku. Ovo je urađeno jer su matrice za transformaciju homogenih koordinata tipično retko popunjene, pa je ova koncepcija nešto brža. Ipak, pokazalo se da dobici nisu veliki.

Što se tiče *lighting* modela, trenutno se koristi *flat* senčenje, a takođe nisu podržane ni teksture. Pitanje je da li će ovo biti menjano, jer bi se u tom slučaju izgubila mogućnost razdvajanja X servera i klijenta. Naime, slanje tekstura X serveru koji nije na istoj mašini sa X klijentom bi izazvalo veliko zagušenje mreže.

## Zaključak

Pokazalo se da koncepcija prostog distributivnog servera bez interpolacije položaja kod klijenata može u nekim slučajevima izazvati diskontinuitet u kretanju objekata. Takođe, pokazalo se da rešenje nije skalabilno na veći broj klijenata. Ipak, ovo rešenje je jednostavno za implementaciju i može da podnese veći broj radnih mesta od koncepcije u kojoj se koristi jedan program koji otvara veći broj *X display*-a, a uz to je naše rešenje moguće skalirati na veći broj radnih mesta. Pokazalo se i da se prihvatljivi

rezultati mogu dobiti bez predikcije, ali da se u određenim situacijama javljaju diskontinuiteti u kretanju.

Što se vizuelizacije tiče, pokazalo se da X Window predstavlja sistem koji je u pogledu performansi ravan ili bolji od MS Windows platformi, s tim što ako server i klijent nisu na istoj mašini senčenje i tekstuiranje nije moguće sa prihvatljivim performansama na standardnom 10 Mbit/s Ethernet-u.

## Dalji razvoj

Krajnji cilj projekta je razvoj višekorisničkog simulatora letenja, pri čemu bi se uzela u obzir mogućnost da jednog korisnika opslužuje veći broj računara, koji bi između sebe podelili vizuelizaciju i fiziku kretanja pri komplikovanim fizičkim modelima leta. Svako radno mesto bi imalo poseban segment mreže, a bio bi odvojen i segment mreže za vezu servera sa svakim radnim mestom. Očekuje se da će pri ovakvoj realizaciji koncept autoritativnog servera biti inferioran u odnosu na ovakvo rešenje, jer bi se sa povećanjem broja korisnika vrlo brzo došlo do granice performansi koje server prihvatljive cene može da pruži. S druge strane, prosti distributivni server ima manje problema pri povećanju broja korisnika. Tek treba ispitati koliki deo autoriteta treba da ostane u serveru, kao i mogućnost distribucije servera na više mašina povezanih na mrežu velike brzine.

---

## Literatura

Foley, J.D., Van Dam, A., Feiner, S.K., Hughes, J.F. 1995. *Computer Graphics, Principles and Practice*, (2nd ed. in C). Addison Wesley.

Habermeier, B. 1997. *Internet Based Client/Server Network Traffic Reduction*. Dostupan preko <http://www.bolt-action.com>.

Stewens, R.W. 1990. *UNIX Network Programming*. Prentice Hall.

---

*Veljko Krunić & Boris Dragović*

## The Problems of Synchronization in Real-time Graphical Systems

While implementing distributed real-time graphical systems we encounter various problems in synchronizing objects. These problems could be reduced up to a point where they don't influence the user's subjective impression by means of designing the system in a proper way. We will

discuss some alternative ways of designing server and client applications for real-time distributed graphical systems, with special emphasis on multi-user video games, as well as practical experience in designing the above mentioned. The platform used for this project is a network composed of interconnected UNIX and Windows machines, where Windows machines represent only X servers. At the moment the project is in its starting phase.

