

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA, ZAGREB
Zavod za elektroničke sustave i obradu informacija

Sustavi za praćenje i vođenje procesa

Aktivne mreže

Nino Talian (0036393159)

U Zagrebu, lipanj 2005.

SADRŽAJ

1. Uvod	3
2. Zašto aktivne mreže?	4
3. Aktivne mreže	5
3.1. Programabilni čvorovi - diskretni pristup.....	5
3.2. Kapsule - integrirani pristup.....	5
3.3. Programiranje kapsulama.....	6
3.4. Interoperabilni programski model.....	7
4. Razmatranja o arhitekturi mreže	8
5. Zaključci	9
6. Literatura	10

1. Uvod

Aktivne mreže omogućuju svojim korisnicima unos osobno kreiranih programa u čvorove mreže. One, dakle, nad podacima mogu obavljati razne funkcije. Korisnik, primjerice, može poslati svoj program za kompresiju u mrežni čvor (npr. router) i zatražiti da se tim programom kompresiraju podaci koji prolaze tim čvorom.

Aktivne arhitekture značajno povećavaju sofisticiranost obrade podataka koji putuju takvim mrežama. Omogućuju korištenje novih aplikacija, posebice multicast aplikacija, spajanje informacija te mnoge druge usluge bazirane na mrežnoj obradi i pohrani podataka.

Klasične mreže pasivno prenose bitove s jednog kraja sustava na drugi bez ikakvih modifikacija. Osim prijenosa bitova jedino što rade jest procesiranje zaglavlja (kod komutiranja paketa), odnosno signalizaciju (kod komutacije kanala). Aktivne tehnologije susreću se u slojevima iznad mrežnog, a ova ideja takvu praksu omogućuje i u mrežnom sloju.

Aktivne mreže su «aktivne» na dva načina:

- switchevi obrađuju podatke koji njima prolaze
- korisnik može ubaciti svoj program u mrežu i time odrediti na koji način će čvorovi procesirati podatke

Takva ideja u svojoj najekstremnijoj verziji predlaže uporabu «kapsula» umjesto klasičnih paketa podataka. Kapsule uz korisničke podatke sadrže i djeliće nekog programa koji se izvršava u svakom routeru ili switchu kroz koji prolaze. Osim što može usaditi program u mrežni čvor, kapsula može i aktivirati već postojeći program u čvoru.

Postoje tri osnovne prednosti u baziranju arhitekture mreže na razmjeni aktivnih programa umjesto pasivnih paketa:

- 1) razmjena koda predstavlja osnovu za prilagodljive protokole, time što omogućuje bogatiju interakciju nego kod razmjene običnih paketa
- 2) kapsulama je omogućena implementacija točno određenih funkcija u strateške točke unutar mreže
- 3) zahvaljujući programabilnoj arhitekturi korisnik može prilagoditi infrastrukturu sebi; usluge implementirane u aktivnu mrežu rade brže nego u nekom višem sloju (npr. aplikacijskom)

2. Zašto aktivne mreže?

Aktivne arhitekture omogućuju da se obrada nad podacima raspodjeli po mreži gdje je potrebno. Područja kojima su ovakve arhitekture posebno zanimljive, jer zahtijevaju transparentnu preraspodjelu obrade podataka, su:

Firewall

Firewall aplikacije implementiraju filtre koji određuju koji paketi će se propustiti, a koji će biti blokirani. Uz routing paketa, one u routere ugrađuju aplikacijske i korisničke funkcije. Za firewall je nužno da se redovito obnavlja (update), a aktivnim mrežama taj proces može biti automatiziran tako što dopušta aplikacijama ovlaštenih pružatelja usluga da, nakon autentifikacije, ubace potrebne module u firewall.

Web poslužitelji

Web poslužitelji (web proxies) su primjer aplikacijske usluge napravljene za posluživanje i privremeno pohranjivanje podataka (cache). U hijerarhijskoj shemi čvorovi za cache nalaze se na rubovima mreže tj. kod krajnjeg korisnika ili organizacije. Shema se može unaprijediti tako da se omogući čvorovima hijerarhije da se nalaze na strateški važnim mjestima u mreži pružatelja usluga i veza za razmjenu podataka. Razvijaju se algoritmi i alati koji automatski uravnotežuju hijerarhiju razmještanjem samih memorija za privremenu pohranu podataka (caches), a ne samo podataka (cached information).

Još jedan argument za korištenje aktivnih tehnologija za web caching je taj što se neki djelići web stranica dinamički obnavljaju i nisu prigodni za klasični (pasivni) caching. To predlaže korištenje sheme web poslužitelja koja podržava aktivne cache memorije za pohranu i izvršavanje programa koji generiraju web stranice.

Mobilna/nomadaska obrada podataka

Ovo područje opisuje obradu i način razmjene podataka između rubnog sustava i mreže. Modul koji se nalazi na toj granici zove se «nomadski» router. On promatra i prilagođava se tome kako je sustav spojen na mrežu. Može, primjerice, više podataka stavljati u međuspremnik i tako slati sporije nego što prima, ako je veza s mrežom spora. Također, može kompresirati podatke ili uvesti dodatne sigurnosne mjere (enkripcija). Nomadski agenti i gateway-i su čvorovi u mreži koji podržavaju mobilnost. Nalaze se na mjestima koja spajaju područja u mreži različite brzine i pouzdanosti prijenosa. Najbolji primjer jest spoj između žičane mreže i bežične mreže.

Nove vrste aplikacija

Postoji mnoštvo aplikacija koje za svoj rad zahtijevaju sofisticirane mrežne usluge koje podržavaju distribuciju i spajanje podataka. Ideja je razvitiak multi-point komunikacijskih strategija koje su mnogo fleksibilnije od postojećih IP multicast usluga. Primjer aplikacije koja koristi spajanje podataka (fusion) jest paralelno prikupljanje podataka sa senzora, simulacija i daljensko upravljanje. U mreži se tako sakupljaju i grupiraju podaci blizu njihovih izvora i to na mrežnom sloju. To zahtijeva manji bandwidth prema korisniku koji se nalazi na drugoj strani mreže te proširuje ograničenja oko povratne veze između senzora i mrežnog čvora gdje se izvršava interaktivna aplikacija.

3. Aktivne mreže

Aktivne mreže su visoko programabilne mreže koje obrađuju korisničke podatke što njima prolaze. Možemo im pristupiti na dva načina: diskretni i integrirani; ovisno o tome da li se programi i podaci prenose diskretno tj. u odvojenim porukama, ili integrirano.

3.1. Programabilni čvorovi - diskretni pristup

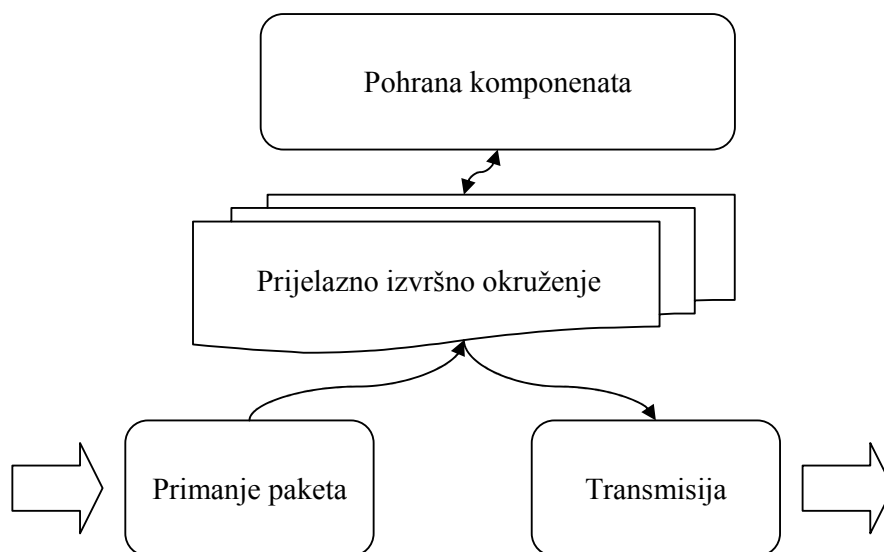
Procesiranje poruka može, što se tiče arhitekture, biti odvojeno od samog postupka implementacije programa u čvor. Kroz takav programabilni čvor korisnici šalju podatke na uobičajen način. Kad paket stigne, njegovo zaglavlje se pregledava i program se aktivira nad zadanim parametrima. Program aktivno obrađuje paket i po potrebi mijenja njegov sadržaj. Način obrade podataka ovdje može odrediti korisnik jer zaglavlje paketa govori koji će se program izvršiti.

Korisno je odvojiti postupke učitavanja i aktiviranja programa u aktivnom čvoru. To se primjerice koristi u inteligentnim mrežama. Svaki mrežni čvor može imati «back door» kroz koja novi program može unositi jedino operator routera uz odgovarajuću autentifikaciju.

3.2. Kapsule - integrirani pristup

Kod ovog pristupa svaka poruka je ujedno i dio programa. Svaka poruka, tj. kapsula, koja putuje između čvorova sadrži djelić programa (fragment) koji može sadržavati i podatke. Kad kapsula dođe do aktivnog čvora, njen sadržaj se pregledava, nad njom se izvršavaju programi te se šalje dalje.

Na slici je konceptualni prikaz organizacije aktivnog čvora. Bitovi koji stižu do čvora prvo su procesirani klasičnim mehanizmom koji razlučuje granice kapsule. Sadržaj kapsule prosljeđuje se u prijelazno izvršno okruženje gdje se sadržaj pregledava. Tamo se nad njime izvršavaju programi sastavljeni od «primitivnih instrukcija». Kapsule mogu aktivirati već postojeći program iz memorije čvora ili izgraditi novi program čije djeliće donose. Kad se kapsula obradi, prijelazno izvršno okruženje se uništava.



3.3. Programiranje kapsulama

Navedenim postupcima razina obrade i procesiranja podataka spušta se do mrežnog sloja pa tako sama mreža može obavljati funkcije kao što su procesiranje IP paketa, veza, tokova podataka, protokola za usmjeravanje itd. Sve to omogućeno je upravo programiranjem kapsulama.

U jednostavnim aplikacijama, kapsula jedino određuje svoj skok do slijedećeg čvora i po potrebi se kopira za slanje nekim drugim linkovima. Potrebno je napraviti mehanizme koji određuju i nazivaju izlazne veze kojima se kapsule dalje šalju. U IP protokolu ovaj mehanizam je ugrađen u svaki čvor, a sami paketi trebaju sadržavati samo svoju polazišnu i odredišnu adresu. Ne trebaju znati ništa o vezama kojima putuju. Kapsule, pak, dinamički određuju broj i svojstva svih puteva koje čvor ima na raspolaganju te se prema tome po potrebi modificiraju i šalju dalje.

Postavlja se pitanje do kojeg stupnja dati kapsulama pristup nekom objektu u mrežnom čvoru, kao što je routing tablica. Tri su glavna načina na koje programi nadilaze prijelazno izvršno okruženje (vidi sliku):

Temeljne komponente

- univerzalno dostupne usluge implementirane izvan kapsule
- one implementiraju eksterne metode koje omogućuju kontrolirani pristup resursima izvan prijelaznog izvršnog okruženja
- druge komponente predstavljaju ugrađenu hijerarhiju klasa koja služi kao osnova za razvoj programa iz kapsula

Aktivna obrada podataka

- svojstvo da se modificira stanje u kojem je memorija čvora ostavljena nakon obrade kapsule
- moguće je otvoriti vezu tako da se prva kapsula obrađuje u svakom čvoru ostavljajući trag za ostale kapsule koje slijede iza nje
- informacije koje kapsula ostavi u memoriji čvora nazivaju se «stanje toka» jer koriste za usmjeravanje kapsula; ako te informacije više nisu potrebne, one se brišu iz memorije čvora
- kapsule mogu čekati jedna drugu u čvoru što je posebno pogodno za spajanje informacija (fusion) ili za multi-casting

Programska proširivost

- mogućnost da programi definiraju nove klase i metode u aktivnom mrežnom čvoru
- programe mogu donositi i implementirati kapsule, ali također mogu biti pohranjeni u nekoj pričuvnoj memoriji čvora pa biti aktivirani kapsulom

3.4. Interoperabilni programski model

Da bi bile «pravi» alat, kapsule zahtjevaju mobilnost da bi programi mogli biti prenošeni mrežom, te mogućnost ugradnje u mrežne komponente. Razvoj modela za programiranje mrežnih čvorova te opis i dodjeljivanje resursa zahtjeva ispunjenje slijedećih uvjeta:

Mobilnost prijenosa

- svojstvo prijenosa kapsula i njihovog izvršavanja na različitim platformama

Sigurnost

-ograničavanje resursa dostupnih kapsulama

Učinkovitost

- omogućiti gore navedeno bez narušavanja performansi mreže, barem u najčešćim slučajevima

Klasične mreže (koje prenose pakete) postižu interoperabilnost standardizacijom sintakse i oblika paketa. Aktivni čvorovi nad paketima mogu izvršavati mnoštvo različitih programa. Mrežna interoperabilnost ipak nije postignuta standarizacijom tih programa, već standardizacijom seta instrukcija i resursa dostupnih kapsulama.

4. Razmatranja o arhitekturi mreže

Količina obrade podataka u aktivnim mrežama može dinamički varirati. Ta obrada podataka je prilagodljiva korisniku i aplikacijama. Ovdje se u mrežnom sloju susreću funkcije koje su inače iz drugih slojeva. Takva odstupanja od klasičnog modela postavljaju brojna pitanja.

Kako je postignuta interoperabilnost?

Ključ interoperabilnosti je u osnovnim funkcijama mrežnog sloja. Kod interneta, primjerice, postoje točno određene specifikacije sintakse i oblika paketa IP protokola koje moraju biti implementirane u svoj mrežnoj opremi. Na ovaj način mrežni elementi na jednak način obrađuju podatke. Aktivni čvorovi vrše različitu obradu podataka, ali prema istom modelu uz standardizirane kodove. Tako je, na mrežnom sloju, interoperabilnost bazirana na dogovorenim kodovima i okruženju obrade podataka umjesto standardiziranih paketa i jednake obrade podataka.

Nije li funkcionalnost u mreži smanjena?

Povećanjem fleksibilnosti i količine obrade podataka u mreži korisnicima se omogućuje veća kontrola nad mrežom i onime što procesira.

Zašto sve ovo nije učinjeno prije?

Pristup koji se ovdje razmatra predlaže sintezu brojnih tehnologija: aktivnih mrežnih platformi, softverski inženjering baziran na komponentama, te mobilnost programskog koda. Bilo je nekih «programabilnih mreža» u prošlosti, ali ti modeli nisu spuštali mobilnost programskog koda do mrežne razine stavljajući njegove fragmente u kapsule. Danas postoje tehnologije i resursi koji omogućuju da aktivne tehnologije funkcioniraju sigurno i učinkovito.

5. Zaključci

Ova ideja opisuje aktivnu mrežnu arhitekturu koja bi bila programabilna od strane korisnika. Aktivne mreže omogućuju rad niza novih aplikacija kao dodatak već postojećim koje se oslanjaju na programabilnost mreže.

Programiranje mreže

Nameće se razvoj komponenata protokola koje mogu biti «krojene» i generirane od strane korisnika te izvršavati određene funkcije nad paketima. Ti programski alati temeljeni su na metodi enkapsulacije tj. stavljanja djelića programa u kapsule.

Infrastrukturne inovacije

Aktivne mreže zahtijevaju razvoj mrežne opreme i usluga pošto se mrežne usluge uzdižu na viši nivo od pukog hardware-a. Konvencionalni mrežni routeri bazirani su na odgovarajućem hardware-u podržanom određenim software-om. Aktivne mreže mijenjaju strukturu mrežne industrije odvajanjem funkcija hardware-a i software-a.

Općenito

Aktivne mreže krše mnoga klasična pravila u arhitekturi mreže koja su postavljena kod izrade prvobitnih mrežnih modela. Pasivne mreže s razmjenom paketa i komunikacijom s kraja na kraj nam služe dobro, ali aktivne mreže otvaraju prostor novim aplikacijama, čine mrežu programabilnom, podesivom i upravljivom korisniku. Ovakve ideje potiču na razvoj novih generacija fleksibilnih mreža i ubrzavaju inovaciju infrastrukture.

6. Literatura

1. David L. Tennenhouse and David J. Wetherall, Towards an active network architecture, *Computer Communication Review*, Vol. 26, No. 2, April 1996.
2. Samrat Bhattacharjee, Kenneth L. Calvert, Ellen W. Zegura, An Architecture for Active Networking, IFIP 1996. Published by Chapman & Hall
3. David M. Murphy, Building an Active Node on the Internet, Massachusetts Institute of Technology, May 1997.