FER - Preddiplomski studij računarstva
Ak. god. 2011/12
Predmet: Komunikacijske mreže

Algoritmi usmjeravanja – algoritam vektora udaljenosti, primjer

Jedan od problema algoritama vektora udaljenosti je spora konvergencija, pri čemu se algoritam bolje ponaša kod "dobrih vijesti" (dodavanje čvora), nego "loših vijesti" (ispad čvora u topologiji).

Slučaj 1.
Reakcija na "dobre vijesti": čvor A bio je isključen, a sada se uključuje – sa svakom porukom, promjena se propagira za jedan korak dalje. U ovom primjeru, nakon izmijenjene 4 poruke, svi čvorovi u mreži znaju da je A dohvatan i kako usmjeravati prema njemu. Općenito vrijedi: ako je najduža staza usmjeravanja u nekoj podmreži N, onda je potrebno N razmjena poruka da bi svi čvorovi "saznali" za novi čvor i put prema njemu. Očito da ovaj algoritam ne bi bio dobar za globalni Internet, ali za mreže "s manjim radijusom" nije loš.

	A (X)	B (X)	C (X)	D (X)	E (X)
početno	OFF	∞	∞	∞	∞
promjena	**OK**				
poruka 1		1	∞	∞	∞
poruka 2		1	2	∞	∞
poruka 3		1	2	3	∞
poruka 4		1	2	3	4

Slučaj 2.
Reakcija na "loše vijesti": čvor A bio je uključen, a zatim dolazi do ispadanja s mreže. U trenutku kad B više ne dobija tablicu udaljnosti od A, zadnja informacija (poruka 1) kojom raspolaže upućuje da C ima vezu prema A (d=2). B osvježava svoju tablicu, dodajući 1 (udaljenost od B do C) na udaljenost koju objavljuje C (d=2+1=3), dobivajući tako vrijednost 3.
Ta vrijednost će sljedećom porukom (poruka 2) stići do čvora C. Čvor C, znajući da mu je udaljenost prema A jednaka udaljenosti 1+(udaljenost od A do B), osvježava svoju tablicu (d=1+3=4) i tu informaciju šalje u sljedećoj poruci (poruka 3). Po primitku poruke (radi jednostavnosti, pretpostavimo da je poruka 3 poslana višeodredišno, tj. istovremeno na B i D), čvorovi B i D osvježavaju svoje stanje na temelju nove udaljenosti od A do C, dodajući 1 (d=4+1=5). U sljedećem koraku, novo stanje utječe na čvorove C i E, itd. S vremenom, udaljenost raste u beskonačno, pa se stoga ovaj problem i naziva "brojanje u beskonačnost" (engl. *count to infinity*).

	A (X)	B (X)	C (X)	D (X)	E (X)
početno	OK	1(izravno)	2(preko B)	3(preko C)	4(preko D)
promjena	**OFF**				
poruka 1		3(preko C)	2	3	4
poruka 2		3	4(preko B)	3	4
poruka 3		5(preko C)	4	5(preko C)	4
poruka 4		5	6	5	6(preko D)
...		.	.	.	.
...		.	.	.	.
...		∞	∞	∞	∞

Za problem "brojanja u beskonačnost" u praksi je predloženo više rješenja, jedno od kojih je poboljšanje algoritma (engl. *split horizon*) koje se često koristi. Na žalost, može se pokazati da i to rješenje u nekim slučajevima može zakazati! Nadalje, vrijednost ∞ ima smisla postaviti na neki konačni iznos – ako je najduža staza usmjeravanja u nekoj podmreži N, dovoljno postaviti taj iznos na N+1. Zato se protokoli utemeljeni na vektoru udaljenosti uglavnom koriste u manjim mrežama (primjer je protokol *Routing Information Protocol* (RIP) koji postavlja vrijednost za ∞ na 15).