

IX Karakteristike transportnog sloja

- 1. Karakteristike Transportnog sloja**
- 2. Specifičnosti prenosa podataka u BSM**
- 3. Podela transportnih protokola u BSM**
- 4. STCP - *Sensor Transmission Control Protocol***
- 5. PSFQ - *Pump Slowly, Fetch Quickly***
- 6. RMST - *Reliable Multi-Segment Transport***
- 7. lwIP – *Light Weight Internet Potocol***

9.1 Karakteristike Transportnog sloja

- Uspešnost i efikasnost neke aplikacije u BSM u mnogome zavisi od **pouzdanе razmene podataka** između SČ-ova u BSM.
- Uloga ljudskog faktora je **svedena na minimum** pa SČ međusobnom razmenom paketa **potpuno samostalno organizuju i izvršavaju** zadatke.
- Primena jednog **efikasnog transportnog protokola** od primarne važnosti i kod ovih mreža, u odnosu na standardne mreže, ima još veći značaj:
 - 1. poveže** aplikacioni i mrežni sloj - **multipleksiranje/demultipleksiranje**
 - 2. obezbedi** pouzdan prenos od izvora do odredišta
 - 3. reguliše** intenzitet mrežnog saobraćaja
- Koriste se **dodatni komandni paketi** za komande i podatke.
- **Cilj** svakog energet.efikasnog protokola je da **broj komandnih paketa** svede na **najmanju moguću meru** kako bi potrošnja energije bila manja
- Energetski efikasan transportni protokol je važan i zbog činjenice da on može u mnogome **da smanji broj retransmisija** tj. ponovnih slanja
- Ušteda energije može se izvršiti i **formiranjem paketa** u kojima će veličina zaglavlja (**header**) biti **manja** od korisnih (**payload**) podataka
- Moraju da poseduju **mehanizam za kontrolu gustine saobraćaja**

9.1 Karakteristike Transportnog sloja

- Svi SČ-ovi moraju da **imaju isti tretman** i moraju da budu **potpuno ravnopravni u komunikaciji**.
- Očekuje se da uspostave **jednostavan početni prenos** između dva SČ kao i da imaju mogućnost **da sarađuju sa protokolima** susednog sloja
- Nije dovoljno samo primeniti **već postojeće standardne transportne protokole TCP i UDP**.
- Velika razlika u karakteristikama žičanih i bežičnih mreža u odnosu na BSM, prouzrokovala je da oni **nisu primenljivi u ovim uslovima**.
- Velika primenljivost BSM, i samim tim velika raznolikost aplikacija, uslovlila je **razvojem velikog broja različitih** transportnih protokola.
- Zbog **otežanih uslova rada**, vrlo teško napraviti mrežne protokole koji bi garantovali **100% pouzdan prenos** informacija.
- Protokoli ovog sloja su do sada i **najmanje istraženi** jer se smatralo da oni nemaju neku veliku ulogu u uštedi energije.
- Sa pojavom **multimedijalnih mreža**, a samim tim i sa povećanjem količine paketa koji se razmenjuju, **uloga** ovih protokola je **znatno porasla** sa gledišta uticaja na energetska efikasnost BSM.

9.1 Karakteristike Transportnog sloja

➤ **Glavni ciljevi** koje moraju da ispune protokoli transportnog nivoa:

1. Kontrola zagušenja

- ✓ usled pojačanog saobraćaja, mogu se pojaviti **pojedinačna zagušenja** kod nekih SČ a naročito oko *sink*-a.
- ✓ to može da prouzrokuje **gubitak paketa**
- ✓ kontrola zagušenja predstavlja **veoma važnu komponentu** koja treba da **omogući postizanje potrebne pouzdanosti** rada aplikacije u BSM.
- ✓ kontrola zagušenja ne samo da **povećava efikasnost mreže**, već i pomaže u **očuvanju oskudnih resursa** kojima raspolaže SČ.

2. Pouzdani transport

- ✓ u zavisnosti od zahteva aplikacije, očitani podaci koje SČ dostavljaju **zahtevaju određeni nivo pouzdanosti** koji mora da bude zadovoljen.
- ✓ kod mnogih aplikacija je potrebno dostaviti, u suprotnom smeru, **podatke o reprogramiranju SČ** (programski kod)
- ✓ ovi podaci zahtevaju **100% pouzdanost**, što opet predstavlja jako ozbiljan zadatak za ovaj sloj kako bi se **osiguralo funkcionisanje** aplikacije u BSM.

9.1 Karakteristike Transportnog sloja

3. (De) multipleksiranje

- ✓ različite aplikacije mogu biti instalirane u jednoj BSM.
- ✓ one imaju različite podatke koje šalju istim ili različitim rutama
- ✓ transportni sloj bi trebao da omogući da svaka aplikacija dobije odgovarajuće podatke koji joj pripadaju
- ✓ da bi izvršio ovaj zadatak koristi multipleksiranje/demultipleksiranje

- Za rešavanje ovih problema u okviru transportnog sloja razvijeno je nekoliko različitih metoda koje su primenjivane kod klasičnih mreža.
- Nažalost, ni jedna od ovih metoda nije premenljiva u BSM u formi kako je data, već one zahtevaju značajne modifikacije
- Tako na primer, većina postojećih rešenja se fokusira na pouzdan prenos podataka koji se zasniva na tehnici *end-to-end* koju diktira TCP
- Pojam *end-to-end* pouzdanosti zasniva se na slanju ACK i NACK koje se razmenjuju od izvornog do odredišnog čvora (od kraja do kraja).
- Takav način komunikacije, pre svega zbog *muti-hop* prirode BSM, zahteva značajne troškove za implementaciju kako u pogledu energetske efikasnosti tako i u povećanim zahtevima za resursima SČ

9.1 Karakteristike Transportnog sloja

- Ranije je napomenuto da se u BSM **najveći deo energije** utroši na prenos podataka.
- Energija koja je potrebna za prenos **1b** informacije gotovo identična energiji koja je potrebna da se izvrši **1000 operacija** u tipičnom SČ.
- **Smanjivanjem količine podataka** koji se šalju može postići jako velika ušteda u količini potrošene energije.
- Sa gledišta uštede energije, **bolje je opteretiti CPU** aplikacijom koja će smanjiti količinu podataka za samo 1B, **nego poslati taj bajt podataka**
- **Dva** su osnovna pristupa za to:
 1. da se izvrši **grupisanje pristiglih podataka** u SČ
 2. da se izvrši **sažimanje** (kompresija) podataka koji se šalju.
- Grupisanje/eliminisanje paketa predstavlja **najjednostavniji pristup**.
- U BSM više SČ-ova mogu da generišu **značajan broj sličnih podataka**
- Koriste se neke od sledećih funkcija: **eliminacijom duplih poruka, min, max** ili **izračunavanjem srednje vrednosti** od pristiglih podataka.
- Komresovanje podataka postižu su **dva cilja**: da se **minimizira saobraćaj** i da se podaci koji se šalju **zaštite od neželjenih primalaca**.

9.2 Specifičnosti prenosa podataka

- Ograničena količina energije, limitirana CPU snaga i jako oskudni hrdverski kapaciteti SČ donose dodatne problem za dizajn protokola
- Razvoj univerzalnog protokola transportnog sloja je neizvodljiv jer u njegovom dizajnu pored navedenih ograničenja SČ postoje i specifični zahtevi aplikacije koja prvenstveno određuje principe rada.
- Potrebno je da ukažemo na neke osnovne elemente koji u mnogome utiču na razvoj jednog transportnog protokola u BSM:

1. Smer prenosa podataka

- ✓ može biti *downstream* ili *upstream*.
- ✓ *Downstream* prenos predstavlja prenos paketa koji se koristi za slanje operativnih ili aplikaciono specifičnih podataka od *sink-a* ka SČ.
- ✓ To je *multihop* jedan prema-više (*multicast*) komunikacija.
- ✓ *Upstream* prenos predstavlja prenos podatak od SČ-ova ka *sink-u* i to je više prema jedan prenos.
- ✓ Obično taj prenos započinje *sink* kada se interesuje o nekom podatku i on predstavlja *agregatni, tj. glavni SČ* koji prikuplja informacije od više SČ-ova koji pripadaju jednom određenom regionu

9.2 Specifičnosti prenosa podataka

2. Mehanizam prenosa podataka

- a. **direktan** od *sink*-a do SČ tj. svaki SČ komunicira sa *sink* (*end-to-end*)
- b. **indirektan** gde se prenos odvija i *multi hop* režimu (*hop-by-hop*).

3. Brzina prenosa

- ✓ predstavlja **vrlo bitan faktor** kod prenosa nekih podataka (*end-to-end*)
- ✓ kod prenosa multimedijalnih podataka gde postoje **stroge granice dozvoljenog max. kašnjenja** primljenih podataka (*real-time* prenos).
- ✓ **teško je zadovoljiti** ovu karakteristiku zbog velikih ograničenja BSM u pogledu **komunikacionih uslova, višeskokovite topologije, uskog propusnog opsega, asimetričnih veza i mnogim greškama.**

4. Potvrda prijema podataka

- ✓ dve vrste potvrde prijema i to: **potvdu ispravno primljenog paketa (ACK)** i **negativnu potvdu** da paket nije stigao do odredišta (**NACK**).

5. Vrsta adresiranja

- ✓ potreba za *unicast*, *broadcast* ili *multicast* adrese kod slanja paketa.

6. Tip podataka

- ✓ vrlo bitan za pouzdanost prenosa: **sistemske** ili **obične** podaci

9.2 Specifičnosti prenosa podataka

1. Mehanizam za kontrolu intenziteta saobraćaja

- ✓ treba posebno istaći **efekat promenljive gustine saobraćaja**, kako u **prostornom i vremenskom** domenu tako i u **smeru slanja** paketa.
- ✓ intenzitet saobraćaja **na periferiji BSM je jako slab**, i on raste sa približavanjem glavnom SČ, *sink-u*
- ✓ različiti **intezitet saobraćaja u jedinici vremena**, koji se može klasifikovati na **periodičan** ili **događajem izazvan** (*event driven*).
- ✓ Ovde se radi o saobraćaju koji postaje jako intezivan **samo kada se dogodi neka promena** u nadgledanom regionu.
- ✓ Svo ostalo vreme **intezitet je jako mali** tj. gotovo da ga i nema.
- ✓ Kod detekcije promene javlja se **veliki broj redundantnih podataka**
- ✓ To proizilazi iz činjenice da veliki broj SČ-ova **mogu da detektuju isti događaj**, pa se veoma velika količina paketa nepotrebno šalje.
- ✓ Intenzitet saobraćaja od SČ ka *sink-u* (*upstream*) **znatno veći** nego u suprotnom smeru (*downstream*).
- ✓ Za uspešno rešavanje ovih problema razvijeno je **više algoritama** koji se zasnivaju na **kontroli iskorišćenosti bafera** u SČ i na **kontroli frekvencije/brzine slanja podataka** (ACC-Active Congestion Control)

9.2 Specifičnosti prenosa podataka

2. Mehanizam za kontrolu pouzdanog prenosa paketa

- ✓ pouzdan prenos podataka kod BSM-a u većini slučajeva **može imati drugačije značenje** nego što je kod tradicionalnih žičanih veza.
- ✓ za neke aplikacije kod BSM dovoljno je primiti **samo jedan ispravan paket od SČ-a**, a ne od svih, jer većinom oni **daju istu informaciju**.
- ✓ daje nam **veću slobodu** u razvoju transportnih protokola za BSM.
- ✓ u nekim aplikacijama, prenos **systemskih podataka**, promena ili **izmena programa** (reprogramiranje) u SČ-ima ili kod **prenosa više podataka** (više paketa iste poruke) **moramo obezbediti potpuni sigurni prenos**.
- ✓ za rešavanje ovog problema bolje je koristiti mehanizam **hop-to-hop** (HtH) od tradicionalnog **end-to-end** (EtE) mehanizma
- ✓ na taj način **smanjuje se gustina saobraćaja** (poruke razmenjuju samo dva SČ), a samim tim i ušteda energije je veća.
- ✓ na raspolaganju imamo **ACK i NACK mehanizme** potvrde i negacije prijema ispravnih poruka.
- ✓ njihovom kombinacijom moguće je **na efektan način signalizirati gubitak pojedinih paketa** ili ispravnost njihovog prijema.

9.2 Specifičnosti prenosa podataka

3. Zahtevi aplikacije

- ✓ jedna od razlika u funkcionisanju BSM ogleda se u tome što pouzdanost podataka u mnogome **zavise od primenjene aplikacije**.
- ✓ SČ se mogu koristiti radi izvršavanja **kontinuiranog praćenja** određenog fenomena kao što je nadgledanje temperature ili može se zahtevati **koordinirani rad više SČ** za praćenje nekog događaja
- ✓ neke aplikacije mogu zahtevati **detekciju lokacije** kao i **lokalnu kontrolu aktuatora** na osnovu informacije koja je dobijena očitavanjem podataka sa više SČ.
- ✓ za neke aplikacije, kao što je nadgledanje neke pojave, najbitnija komponenta je **pouzdanost primljenih podataka**, dok za aplikacije za otkrivanje događaja **pravovremenost je ključna komponenta**
- ✓ Iz tog razloga fokus primenjenog transportnog protokola **treba da bude prilagođen aplikaciji** u kojoj se on primenjuje.
- ✓ aplikacije se mogu razvijati i za širok spektar različitih namena u vojnoj, životnoj sredini, zdravstvu, i td. koje sve imaju **različite prioritete pojedinih parametara** transportnog protokola

9.2 Specifičnosti prenosa podataka

4. Karakteristična primena

- ✓ BSM su obično implementirane sa **velikim brojem SČ** koji imaju jako ograničene, siromašne resurse, a koji su povezani sa nekim *sink*.
- ✓ Ograničena moć CPU i kapacitet radne memorije **onemogućavaju izvršavanje sofisticiranih algoritama** u okviru svakog SČ.
- ✓ algoritmi transportnog sloja treba da budu dizajnirani da se **većina matematičke funkcionalnosti** izvodi na *sink*-u sa min. potrebnim funkcionalnostima u okviru samih SČ na terenu.
- ✓ sva inteligencija aplikacije bi trebala da bude **kod sink-a** dok bi svi ostali SČ bili sa **znatno smanjenim funkcionalnostima**.
- ✓ na taj način **maksimalno bi oslobodili SČ**, smanjili njihovu energetska potrošnju i samim tim **produžili životni vek** aplikacije
- ✓ prenos podataka u BSM pokazuje **značajno drugačije karakteristike** u zavisno od smera upućivanja tih podataka.
- ✓ dok prenos podataka u smeru *SČ-sink* može **zahtevati blagovremenu isporuku** sa određenom tolerancijom na gubitak podataka, suprotan smer, *sink-SČ*, **obično zahteva visok stepen pouzdanosti** prenosa

9.2 Specifičnosti prenosa podataka

5. Ograničeno usmeravanje/adresiranje

- ✓ u BSM aplikacijama SČ su **retko dodeljene jedinstvene adrese**.
- ✓ za razliku od protokola kao što je TCP, u dizajniranju protokola transportnog nivoa za BSM treba imati u vidu da **ne postoji klasično globalno adresiranje *end-to-end*** tj. od izvorišta do odredišta.
- ✓ adresiranje SČ zasnovano je na **atributima baziranim** osobinama a rutiranje se vrši na osnovu **očitanih podataka** koje zahteva aplikacija.

6. Jednostavnost u inicijalnom, početnom prenosu

- ✓ većina aplikacija u BSM-a je **reaktivnog tipa** gde SČ predstavljaju pasivne aktere nekog događaja.
- ✓ oni jednostavno posmatraju okolinu, očekujući da se dogodi neki događaj, a **reaguju tek na neku promenu ili prozivku** od nadređenog SČ ili *sink*-a
- ✓ sve te promene mogu se smestiti u **svoga nekoliko paketa** koji se šalju nadređenom SČ-u.
- ✓ zato je potrebno da inicijalni prenos kod uspostavljanja veze **bude što kraći i jednostavniji**.

9.2 Specifičnosti prenosa podataka

7. Mali broj ponovljenih slanja paketa (*retransmission*)

- ✓ vrlo bitna karakteristika koja mora da bude zadovoljena kako bi se **izbeglo nepotrebno trošenje električne energije**.
- ✓ ako i dođe do neophodnosti ponovnog slanja paketa treba voditi računa **da što manji broj SČ-ova bude uključen** u taj prenos.
- ✓ u tom smislu sigurno je da **HtH prenos ima prednostu** u odnosu na EtE, jer su samo dva SČ uključena u ponovljeno slanje paketa.
- ✓ mana ovakvog prenosa je da on **ne može da nam garantuje pouzdan prenos** na nivou cele rute od predajnika do prijemnika, kao EtE.

8. Malo zaglavlje (*header* znatno manji od *payload* podataka)

- ✓ u većini BSM aplikacija, **komunikacija se odvija** po mehanizmu **HtH**
- ✓ svaki od SČ pored osnovne funkcije, detekcije nekog događaja, **treba da zadovolji i funkciju rutera** kod preusmeravanja velikog broja paketa koji kroz njega prolaze.
- ✓ kako se u tim paketima najčešće nalazi **veoma mali broj payload** podataka od velikog je značaja da **header** tih paketa **smanji na najmanju moguću meru** kako bi se smanjila ukupna veličina paketa

9.2 Specifičnosti prenosa podataka

9. Svi SČ moraju da imaju ravnopravan tretman u komunikaciji

- ✓ kako se radi o mreži sa velikim brojem SČ koji nadgledaju neki region, **uvek postoji mogućnost** da neki od SČ-ova bude **zapostavljen** u komunikaciji.
- ✓ iz tog razloga informacije koje dolaze do *sink*-a **mogu biti pogrešne jer nisu kompletne**, pa mogu da dovedu do pogrešnih odluka.
- ✓ zato treba obezbediti da svi SČ u BSM-u budu **ravnopravno tretirani**, kako bi se obezbedila pravovremena i realna informacija o svim promenama u nadgledanom regionu.

10. Saradnja sa susednim čvorovima

- ✓ poželjno je da **postoji međusobna komunikacija** između susednih slojeva tj. sa mrežnim i aplikativnim slojem.
- ✓ ako ta komunikacija postoji tada *routing* protokol sa nižeg sloja može **da obavesti transportni protokol o nekim problemima** koji su se javili u komunikaciji
- ✓ na primer **da obavesti da je gubljenje paketa** zbog prekida nekog puta - *route failure* a ne zbog pojačanog saobraćaja.

9.2 Specifičnosti prenosa podataka

11. Korišćenje više puteva

- ✓ *multy-hop* topologija BSM omogućuje da od izvora do odredišta (*sink*) imamo **više različitih puteva** po kojima možemo da šaljemo podatke.
- ✓ **nekoliko razloga** koja preporučuju ovakav način slanja podataka i to:
 - a) **podela velikih podataka** (*I-frame* ili ROI podatak) na manje delove koji će moći da stanu u limitirane resurse-bafere prolaznih SČ-ova,
 - b) **ograničena propusna moć kanala** po kojima se podaci šalju, pa ako bi podaci išli samo po jednom putu, mi ne bi smo mogli da dobijemo u realnom vremenu dovoljno podataka da bi imali kontinualnu informaciju o posmatranom događaju.
 - c) veliki broj različitih puteva **doprinosi i sigurnijem i pouzdanijem prenosu**. Često se dešava da pojedini putevi otkazu ili su otežani. Zbog prirode multimedijalnih podataka, od bitnog značaja je da se pojedini paketi **usmere drugim putevima**, *real-time* prenos podataka
 - d) **ravnomernije opterećenje rada** svih SČ-ova u mreži. U slučaju konstantnog opterećenja jednog puta, zbog limitiranih energetske kapaciteta SČ, **veoma brzo će neki od njih otkazati rad**.

9.2 Specifičnosti prenosa podataka

12. Potrošnja energije

- ✓ energetska efikasnost je **daleko najvažniji parametar** gotovo svakog protokola koji se primenjuje u BSM
- ✓ **limitirani energetske resursi** utiču na izradu transportnog protokola.
- ✓ funkcionalnost transportnog sloja treba da bude **energetski efikasna**, tj. osnovni ciljevi ovog sloja moraju da budu postignuti uz **minimalnu potrošnju energije**.
- ✓ ako su nivoi pouzdanosti kod *sink*-a veći od onog koji je potreban za otkrivanje događaja, izvorni SČ mogu sačuvati energiju **smanjenjem količine poslatih informacija ili produženom neaktivnom periodom**
- ✓ *end-to-end* tehnika koja se uspešno koristi kod tradicionalnog umrežavanja za pouzdanost slanja podataka, **obično zahteva značajnu potrošnju energije** u *multi-hop* mreži.
- ✓ prema tome, **ova rešenja nisu prikladna** za primenu kod BSM.
- ✓ protokol transportnog sloja može se projektovati tako da se nivo pouzdanosti podataka, od izvora do odredišta, **često može žrtvovati** kako bi se smanjila potrošnja energije u lokalnim SČ koji su u ruti.

9.2 Specifičnosti prenosa podataka

13. Povezivanje sa TCP/IP mrežama

- ✓ veliki broj aplikacija danas zahtevaju da se **SČ izađu na Internet**
- ✓ integracija između proizvoljno raspoređenih SČ-ova i fiksne mrežne infrastrukture kao što je Internet **predstavlja jako veliki problem**
- ✓ osnovni problem se sastoji u **adresibilnosti svakog od SČ-ova**
- ✓ skup TCP/IP protokola spada u jako zahtevne protokole, jer gotovo svi oni **zahtevaju jake resurse**: veća memorija i jači CPU.
- ✓ kako za većinu stacionarnih uređaja to nije problem, ovi protokoli su **danas postali mrežni standard**, koji zbog svoje funkcionalnosti i efikasnosti gotovo **svi poštuju i primenjuju**.
- ✓ jasno je da bi puna primena TCP/IP u BSM-a **bio potpuni promašaj** sa gledišta isplativosti i efikasnosti, jer bi morali da za 2-3 bajta korisnih podataka prenosimo **30B** (samo su zaglavlja dosta velika (**IPv4-24 bajta, IPv6-40 bajta, UDP-8 bajta, TCP-24 bajta**), .
- ✓ Zato su **razvijene mnoge druge alternativne metode** koje su uspele da sa jedne strane zadovolje velike zahteve TCP/IP protokola i iste prilagode skromnim mogućnostima kojima raspolažu SČ-ovi

9.2 Specifičnosti prenosa podataka

- Sagledavajući sve gore izložene specifičnosti, idealno bi bilo **da sve one budi ispunjene** kod primene transportnih protokola u BSM.
- Međutim to je samo idealan slučaj koji je teško izvodljiv, ali se zato mogu definisati **četiri osnovna cilja** kojima treba da težiti svaki energetski efikasan transportni protokol a to su:
 1. **mehanizam za kontrolu sigurnog prenosa,**
 2. **smanjena potrošnja električne energije** svakog SČ tj. minimizacija vremena za prenos podataka,
 3. **laka implementacija** tj. minimalni potrebni resursi za realizaciju protokola.
 4. **promenljiva pouzdanost** prilagođena aplikaciji

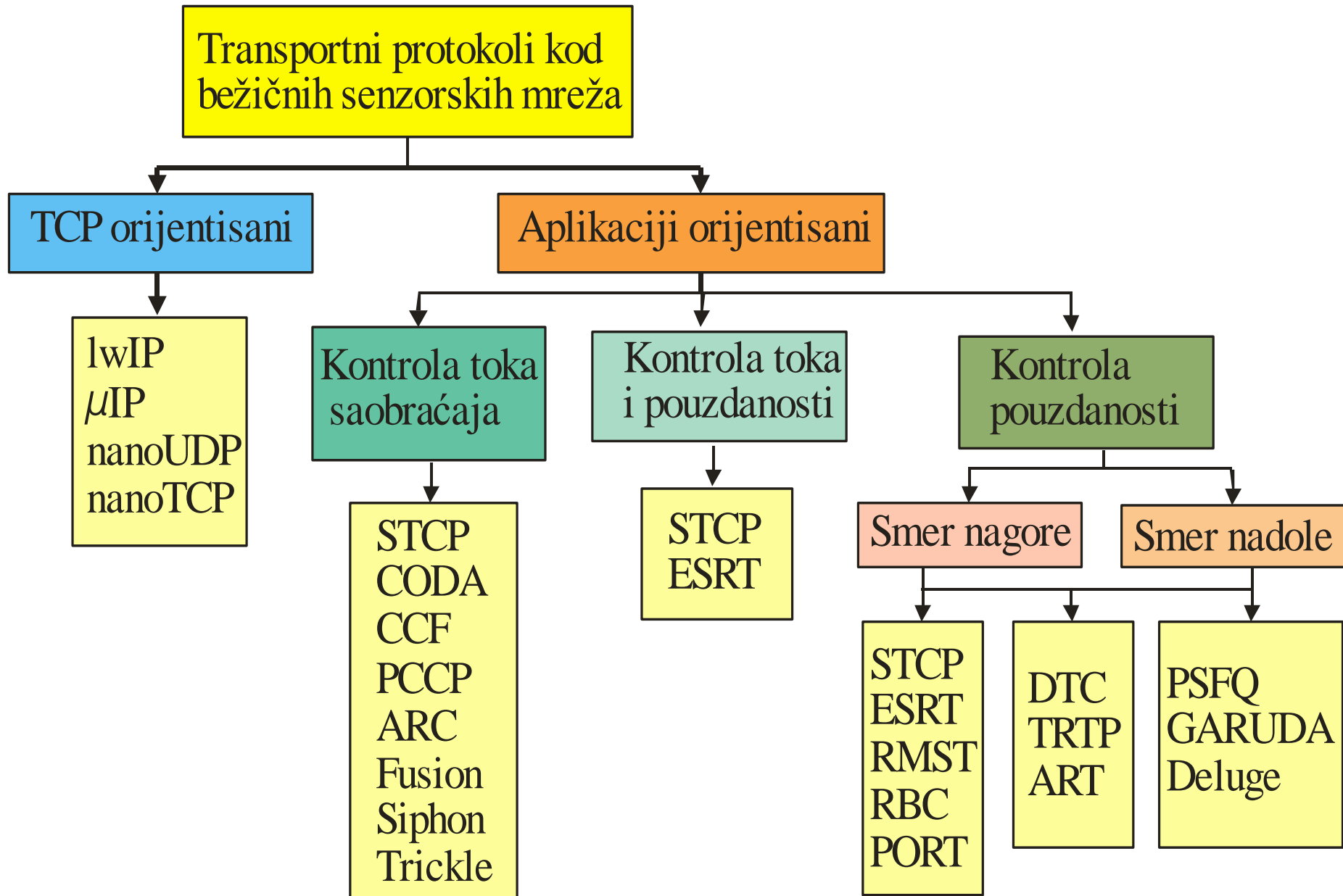
9.3 Podela transportnih protokola

- Zbog specifične prirode BSM, razvijeni su **mnogi transportni protokoli** koji su većinom bili **zavisni od aplikacije** u kojoj su bili primenjeni.
- **Nijedan od tih protokola** nije uspeo da postane standardni transportni protokol za BSM-e, kao što je to bio slučaj sa TCP protokolom
- Postoji više podela ovih protokola i one većinom zavise od **namene i karakteristike aplikacije** u kojoj je taj protokol implementiran.
- **Generalno prema nameni** mogu se svrstati u protokole koji:
 1. **favorizuju pakete** (*packet driven*)
 2. **favorizuju događaj** (*event driven*).
- Kod *packet driven* protokola **pouzdanost je na prvom mestu** jer oni obezbeđuju da svaki poslani paket stigne do destinacije.
- Za protokole tipa *event driven* najbitnije samo da se **detektuje promena koja se dogodila** u nagledanom regionu.
- U zavisnosti od smera slanja paketa delimo ih na dve velike grupe:
 1. **naviše** (*upstream* ili *forward path*)
 2. **naniže** (*downstream* ili *reverse path*)
- Intenzitet *upstream* saobraćaja **mnogo je veći** od *downstream* smera

9.3 Podela transportnih protokola

- Zbog postojanja velikog broja SČ-ova koji prikupljaju podatke i šalju ih prema jednom agregatnom čvoru - *sink* (**tip veze više prema jedan**), postoji **veoma visoki nivo poklapanja ili korelacije** prikupljenih podataka, pa zbog toga **ne postoji potreba za pouzdanim prenosom** tj. **EtE** potvrdom ispravnog prijema podataka.
- Kod ove vrste protokola nije potrebno **ni da se tačno znaju adrese** svih SČ-ova, već je dovoljno da se samo zna **geografska lokacija grupe senzora** od kojih se dobija informacija.
- Sa druge strane kada se podaci prenose od *sink*-a ka SČ-ima (**tip veze jedan prema više**), najčešće se radi o **systemske podacima** (dogradnja ili slanje novog softverskog koda za SČ-ove).
- Ovakav prenos zahteva **jako pouzdan prenos**, jer se radi o bitnim podacima od kojih gubljenje **samo jednog paketa** može da izazove probleme u funkcionisanju aplikacije.
- Na narednom slajdu **prikazani su neki od poznatijih protokola** koji su razvijeni na transportnom nivou za BSM.
- Neki od njih biće **detaljnije objašnjeni u nastavku**.

9.3 Podela transportnih protokola



9.3 Podela transportnih protokola

1. PSFQ (*Pump Slowly, Fetch Quickly*)

- ✓ obezbeđuje **pouzdan prenos** od *sink*-a ka SČ-ima (*downstream*) sa relativno malom brzinom slanja paketa.
- ✓ *Sink* šalje **konstantnom brzinom pakete** svim susednim čvorovima (*broadcast*), a oni prosleđuju te pakete na dalje (HtH organizacija).
- ✓ Prenos na nivou **jednog preskoka je pouzdan** jer se vrši upravljanje prenosom paketa.
- ✓ Potrebno da svaki SČ **memoriše paket** koji prosleđuje dalje, kako bi ga sačuvao i bio u mogućnosti da ga opet pošalje ako dođe do greške.
- ✓ Ukoliko dođe do lošeg prijema nekog paketa, **on se odmah obnavlja na nivou dva susedna čvora** između kojih je došlo do problema
- ✓ Loše primljeni paket se **odmah zaustavlja** tj. ne prenosi se dalje.
- ✓ Kako je priliv novih paketa relativno spor, SČ **ima dovoljno vremena da ponovno pošalje isti paket** (uzima ga iz svog keša) i na taj način ponovo uspostavi normalni prenos podataka.

9.3 Podela transportnih protokola

2. GARUDA

- ✓ Predstavlja *downstream* protokol koji se sastoji iz tri komponente:
 - I. **WFP** (*Wait-for First-Packet*) koja **garantuje uspešan prenos** jednog inicijalnog paketa. Taj paket **treba da odredi senzore** koji će predstavljati **okosnicu puta** preko kojih će se pouzdan prenos podataka odvijati (*Core sensor nodes*).
 - II. Za **izbor tih senzora** zadužena je druga komponenta protokola. Važi pravilo da samo oni SČ koji imaju rastojanje **$3*i$** skoka (*HopCount*), gde je *i* ceo broj, **mogu biti kandidati da postanu *Core sensors***.
 - III. Treća komponenta je zadužena **za retransmisiju loših paketa** i sastoji se iz dva dela:
 1. retransmisije za *Core sensors*
 2. retransmisije za *non-Core sensors* korišćenjem NACK sekvence
- ✓ Glavni nedostaci ovog protokola su što **garantuje pouzdan prenos samo u jednom pravcu** od *sink*-a ka SČ (*downstream*) i što **nema kontrolu intenziteta saobraćaja**

9.3 Podela transportnih protokola

3. RMST (*Reliable Multi-Segment Transport*)

- ✓ protokol koji obezbeđuje pouzdan prenos podataka od **SČ-ova do sink-a (*upstream*)** kako na MAC tako i na transportnom nivou.
- ✓ Glavni cilj je da ispita **na kom nivou treba ostvariti** pouzdani prenos.
- ✓ Pouzdanost na MAC nivou se postiže **bez-ARQ, uvek-ARQ, ili selektivnim-ARQ**-om.
- ✓ Pouzdanost na transportnom nivou se postiže korišćenjem **NACK**-ova (bilo da su oni tipa *end-to-end* ili *hop-to-hop*).
- ✓ U svom radu oslanja se na protokol rutiranja **Directed Diffusion** od koga **dobija rutu kojom treba da idu podaci** od SČ do *sink*-a.
- ✓ Za otkrivanje izgubljenih paketa koristi se **mehanizam vremenskog tajmera** i kada on istekne šalje se NACK poruka prethodnom SČ
- ✓ Moguće je koristi **HtH** režim rada tj. vrši se keširanje primljenih paketa u svakom SČ ili **EtE** režim rada kada se paketi ne pamte u SČ
- ✓ **Nedostaci** ovog protokola su da **ne vodi računa o gustini saobraćaja, nema mehanizam za kontrolu potrošene el.energije u SČ i garantuje pouzdan prenos samo za pojedine pakete** ali ne i za nivo aplikacije

9.3 Podela transportnih protokola

4. ESRT (*Event to Sink Reliable Transport*)

- ✓ Cilj ovog protokola je da obezbedi siguran prenos podataka, koje su SČ-ovi prikupili o nekom događaju, do *sink*-a (*upstream*), tj. **obezbeđuje kolektivnu pouzdanost tipa događaj → sink.**
- ✓ Kontrola sudara i pouzdanost se postižu **podešavanjem frekvencije izveštavanja u SČ.**
- ✓ *Sink* meri pouzdanost posmatranog događaja, aplikacija određuje pouzdanost događaja, *sink* emituje svima novu brzinu izveštavanja
- ✓ Ideja je da se **vrši prilagođavanje brzine prikupljanja** tih podataka u zavisnosti **od kvaliteta komunikacione veze i zahteva aplikacije.**
- ✓ *Sink* **periodično vrši procenu pouzdanosti** prijema paketa u jedinici vremena, i na osnovu toga vrši prilagođavanje brzine slanja
- ✓ Detekcija sudara ili provera kvaliteta veze se vrši u svakom SČ, bilo **nadgledanjem lokalnog reda čekanja ili korišćenjem bita koji ukazuje da je veza do tog čvora loša.**
- ✓ Garantuje pouzdanost **na nivou aplikacije** a ne za pojedinačni paket
- ✓ **Nedostaci** su da je **brzina slanja podataka ista** za sve SČ-ove u BSM kao i da se **koristi kanal sa pojačanom snagom emitovanja (*one-hop*)**

9.3 Podela transportnih protokola

5. CODA (*Congestion Detection and Avoidance*)

- ✓ Osnovni cilj je da detektuje i izbegne sudare (*congestion*) u prenosu
- ✓ Jako je sličan ESRT protokolu i obezbeđuje kolektivni pouzdani transport tipa događaj – *sink*.
- ✓ CODA može da signalizira sudare specifičnim čvorovima regiona.
- ✓ Detekcija sudara se vrši nadgledanjem kanala.
- ✓ On to radi na osnovu posmatranja bafera SČ-ova kao i opterećenosti bežičnog kanala.
- ✓ Kada SČ detektuje da je došlo do prekoračenja osnovnog praga tih vrednosti, obaveštava njegov najbliži *upstream* susedni čvor da smanji brzinu prenosa.
- ✓ Spada u *upstream* protokole i sličan je ESRT-u.
- ✓ Osnovni nedostatak ovog protokola je da ne razmatra pouzdan prenos podataka već samo kontrolu kanala za prenos.

9.3 Podela transportnih protokola

6. STCP (*Sensor Transport Control Protocol*)

- ✓ Omogućuje **generički, skalabilan i pouzdan** transportni protok podataka, koji svoj rad zasniva na funkcionalnosti koja je **postavljena u baznoj stanici – sink-u**.
- ✓ Svaki SČ može biti **izvor različitih podataka** koji imaju različite transportne karakteristike kao:
 - a. tip prenosa,
 - b. brzina prenosa
 - c. stepen pouzdanosti.
- ✓ STPC podržava BSM-e sa različitim vrstama aplikacija koje vrše sakupljanje **velikog broja raznovrsnih tipova podataka**.
- ✓ Omogućava dodatne funkcionalnosti kao što su:
 - a. **kontrolisanje stepena pouzdanosti podataka** koji se primaju,
 - b. **kontrolu mogućih kolizija** kod slanja tih podataka
 - c. **poseduje mehanizme izbegavanja tih kolizija**

9.3 Podela transportnih protokola

7. **ART** (*Asymmetric and Reliable Transport*)

- ✓ Predstavlja transportni protokol koji se **bazira na događaju**.
- ✓ Pouzdanost prenosa podataka **garantuje se u oba smera** i *upstream* i *downstream* ali ne u potpunosti.
- ✓ U *upstream* smeru, ART podrazumeva da je informacija koja stiže od susednog SČ-a **usklađena sa informacijama** koju daju njegovi susedni čvorovi, tako da se **garantuje pouzdanost detekcije događaja** u prostoru koji kontrolišu ti SČ-ovi.
- ✓ To znači da nije potrebno da se **pouzdanu prenesu svi paketi** koje šalju SČ-ovi iz tog regiona.
- ✓ U suprotnom pravcu, ovaj protokol **garantuje pouzdan prenos poruke do nekih SČ-ova** iz regiona koji se pokriva, ali ne i do svih SČ

9.3 Podela transportnih protokola

8. RBC (*Reliable Bursty Convergecast*)

- ✓ Protokol koji obezbeđuje pouzdan prenos **od senzora ka sink-u**.
- ✓ Ovaj protokol je zasnovan na prenosu podataka do kojih dolazi kada se **dogodi neka promena u nadgledanom regionu** (*event-driven*).
- ✓ Protokol omogućava da se **velika količina podataka** prenese u *real-time* režimu rada.
- ✓ Svaki SČ ima svoj **prioritetni red čekanja** i mora da podrži dva osnovna zahteva:
 1. da je svaki SČ sposoban **da osluškuje svoje susedne SČ** i da na bazi toga **predvidi** kada su njegovi okolni susedi primili ili prosledili odgovarajući paket,
 2. da svi SČ-ovi **rade u potpunoj vremenskoj sinhronizaciji**.
- ✓ Ono što predstavlja nedostatak ovog protokola za rad sa multimedijalnim podacima je **slaba optimizacija potrošnje električne energije** kao i **ograničena veličina bafera** u kojoj je smešten prioritetni red čekanja za okvire koji se primaju i prosleđuju.

9.3 Podela transportnih protokola

9. FUSION

- ✓ Predstavlja protokol koji omogućava kontrolu intenziteta saobraćaja od senzora ka *sink-u* koristeći nekoliko mehanizma:
 - a) HtH kontrolu intenziteta protoka podataka,
 - b) ograničavanjem izvorne količine podataka
 - c) korišćenjem odgovarajućeg prioritetnog MAC protokola.
- ✓ Problem korišćenja ovog protokola u BSM sastoji se u tome što ovaj protokol nema razvijen mehanizam obnavljanja primljenih paketa što utiče na njegovu pouzdanost i rad u *real-time* režimu.

10. MRTP (*Multiflow Real-time Transport Protocol*)

- ✓ Razvijen za potrebe *ad-hoc* bežičnih mreža ali zbog svojih osobina primenljiv i u BSM-a.
- ✓ Pogodan za *real-time* prenos podataka sa multimedijalnim sadržajima jer omogućuje prenos podataka preko različitih puteva.
- ✓ Loše strane ovog protokola su te da ne vodi računa o potrošnji električne energije i što nema mehanizam za ponovno slanje nekog paketa – retransmisiju paketa.

9.4 Sensor transmission control protocol

- STCP protokol je generički *end-to-end upstream* transportni protokol koji istovremeno kontroliše i **stepen zagušenja** i **pouzdanost** u prenosu
- Prolazni SČ koriste tehniku RED (*random early detection*) **algoritam za otkrivanje zagušenja** na osnovu **napunjenosti svojih bafera**
- Ako dođe do lokalnih zagušenja, prolazni SČ **postavlja bit zagušenja** koji se šalje sa paketom prema *sink*-u.
- Kada *sink* primi ovaj bit on može da, putem poruka **ACK** ili **NACK**, obavesti polazni (izvorni) SČ **da je došlo do zagušenja na toj ruti**
- Izvorni SČ **može da izabere drugu rutu** ili **da smanji brzinu** kojom šalje pakete kako bi pokušao da ublaži tj. **smanji zagušenje u mreži**.
- STCP obezbeđuje **kontrolisanu promenjivu pouzdanost** u prenosu paketa koja je **prilagodljiva** različitim aplikacijama u BSM.
- STCP inteligentno **uključuje različite transportne funkcije** i koristi različite mehanizme u transportnom protokolu **za različite primene**.
- U SCTP-u postoji **proces iniciranja sesije**, preko koje izvorni SČ može da informiše *sink* o **nekim parametrima** koje zahteva aplikacija kao što su: **vrsta podataka, brzina prenosa i potrebna pouzdanost** u slanju.

9.4 Sensor transmission control protocol

- Za aplikacije koje **kontinualno** šalju pakete/podatke (*continuous data flows*), *sink* **proverava trenutnu pouzdanost veze**.
- Ako je ona ispod potrebnog nivoa pouzdanosti i **očekivani paketi ne stižu u predviđenom roku**, *sink* će poslati **NACK** poruku.
- Kod aplikacija sa **asihronim događajima** (*event-driven data flows*), *sink* koristi **ACK** da informiše izvorni SČ bez obzira da li su paketi **izgubljeni ili uspešno primljeni**.
- Tada izvorni SČ može **ponovo preneti izgubljene pakete** ako potrebna pouzdanost prenosa trenutno nije zadovoljena.
- Aplikacije zasnovane **na zhtevanju nekog podatka** (*data-centric data flows*) mogu **uključiti veliki broj** izvornih SČ.
- Zato nije **praktično i energetske opravdano** da prihvati sve izvorne SČ i da svima njima odgovori sa **ACK** ili **NACK** porukom već se on ponaša **kao tradicionalni UDP protokol**.
- Ukratko, STCP implementira **različitu kontrolu pouzdanosti i zagušenja** kako bi zadovoljio zahteve različitih aplikacija uz istovremeno **poboljšanje energetske efikasnosti** i drugih performansi.

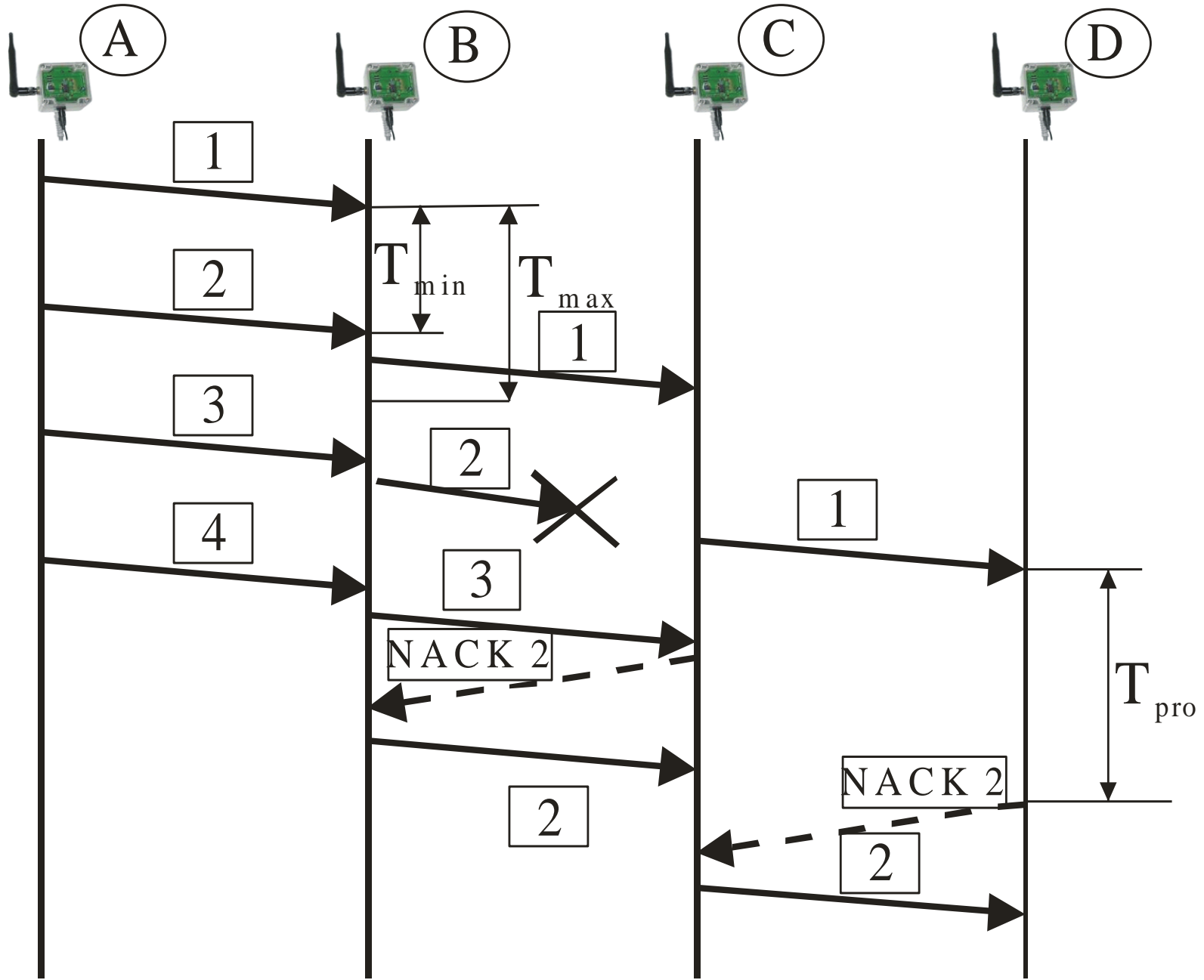
9.5 PSFQ - Pump slowly, Fetch quickly

- U nekim aplikacijama potrebno je izvršiti reprogramiranje/ponovno učitavanje programa kod SČ koji se nalaze na terenu
- U takvim okolnostima potrebno je obezbediti da se svim SČ u BSM pošalje novi programski kod koji će oni besprekorno primiti
- PSFK protokol je upravo razvijen da omogući ovakav vid prenosa.
- Obezbeđuje pouzdan prenos podataka od sink-a ka SČ-ima i pripada tkz. *downstream* transportnim protokolima
- Pretpostavlja se da je izvršena fragmentacija prog.koda koji se šalje u manje delove tj. da je izvršeno segmentiranje potrebne informacije.
- Svaki segment na osnovu broja sekvence (*sequence number*) koja je sastavni deo segmenta predstavlja potpuno jedinstveni paket.
- Od ključnog je značaja da nijedan segment ne bude izgubljen za bilo koje odredište (SČ), jer bi onda bilo nemoguće regenerisati prog. kod, pa bi on bio potpuno beskoristan u slučaju da nedostaje čak i mali deo.
- Osnovni cilj PSFK protokola je da isporući segmente bez i jednog gubitka, da minimizira količinu kontrolnih poruka, a takođe da pruži i neku formu garancije u vidu vremenskog perioda koji će omogućiti da svi SČ mogu da prime poslate segmente.

9.5 PSFQ - Pump slowly, Fetch quickly

- Osnovna ideja sastoji se u tome da SČ **svaki segment šalje** (*pump* - propagira kroz mrežu) **malom brzinom**.
- Kako je ta brzina slanja između dva SČ mala, SČ koji je u ovom slučaju prijemnik **ima dovoljno vremena da otkrije**, na osnovu broja sekvence svakog segmenta, **da li postoji gubitak nekog segmenta**.
- Ako se to dogodilo on **odmah zahteva retransmisiju** (*fetch*) od SČ od koga prima segmente tkz. *hop-by-hop* retransmisiju.
- Sve dok SČ **prima segmente sekvencijalno** on **ne preduzima nikakvu radnju** već ih samo prosleđuje.
- Ako ispravno primi segment koji ima sekvencijalni broj koji se razlikuje od predhodnog sekvencijalnog broja (segment koji je prethodno primio) **za više od jedan, detektuje se greška** tj. gubitak segmenta.
- Kada je otkriven događaj o gubitku, **SČ prestaje da prosleđuje segmente** sve dok ne primi segment koji nedostaje.
- Na taj način **lokalizuje se događaj o gubicima** samo na taj SČ i sprečava dalje širenje greške i mogućnost prepunjavanja bafera u susednim SČ.
- Slanjem zahteva za nedostajućim segmentom **omogućava se da svi susedni SČ** koji imaju ispravno primljen taj segment isti pošalju

9.5 PSFQ - Pump slowly, Fetch quickly



9.5 PSFQ - Pump slowly, Fetch quickly

➤ PSFK protokol omogućava sledeće **tri funkcije**:

1. Funkcija slanja (*pump operation*): pošto je pouzdanost slanja i prijema podataka mnogo važnija od brzine slanja, PSFK se koristi mehanizmom koji se zasniva na sporoj predaji segmenata tj. slanjem segmenata malim brzinama. Svaki SČ na putu do odredišta čeka određeno vreme pre prenosa segmenta.

2. Funkcija preuzimanja (*fetch operation*): u slučaju detektovanja greške u paketima, SČ trenutno zahteva od susednih SČ da mu pošalju segment koji je primio sa greškom i prekida dalje emitovanje

3. Izveštavanje o status (*status reporting*): PSFQ takođe omogućuje funkcionalnost pravljenja izveštaja koja daje informaciju o komunikacionom linku između senzora i *sink*-a tako da sink ima informacije koji se odnose na rad mreže.

➤ **Funkcija slanja segmenata** je podrazumevana strategija kojom se koristi PSFK za prosleđivanje informacija od sink-a ka sensorima sve dok ne dođe do pojave greške.

9.5 PSFQ - Pump slowly, Fetch quickly

- Postoje dva vremenska brojača, **Tmin** i **Tmax**, koji se koriste za **određivanje vremena prenosa** za čvorove duž rute od SČ-a do *sink*-a.
- SČ emituje pakete svojim susedima **uvek nakon isteka** perioda **Tmin**.
- Po prijemu paketa, susedni čvorovi prenose pakete **nakon slučajnog vremena čekanja**, koji je izabran **između Tmin i Tmax**
- SČ mora da čeka **najmanje period Tmin** između prenosa paketa.
- Ova pauza između prenosa paketa treba da omogući da SČ može da **povrati nedostajuće segmente**.
- Pored toga ovo nasumično kašnjenje omogućava **smanjenje broja redundantnih emitovanja** istih segmenata od susednih SČ-a.
- Ako neki segment prosleđuje jedan od SČ, **njegovi susedni SČ suzbijaju da taj segment prenose**.
- SČ šalje segmente koji se odnose na određenu poruku **sa uzastopnim brojevima sekvence**.
- Ako je čvor na putu od *sink*-a do senzora detektovao **preskakanje u broju sekvence**, to znači da je došlo do greške i odmah suzbija dalje slanje segmenata i **pokreće operaciju preuzimanja (fetch)** - šalje NACK

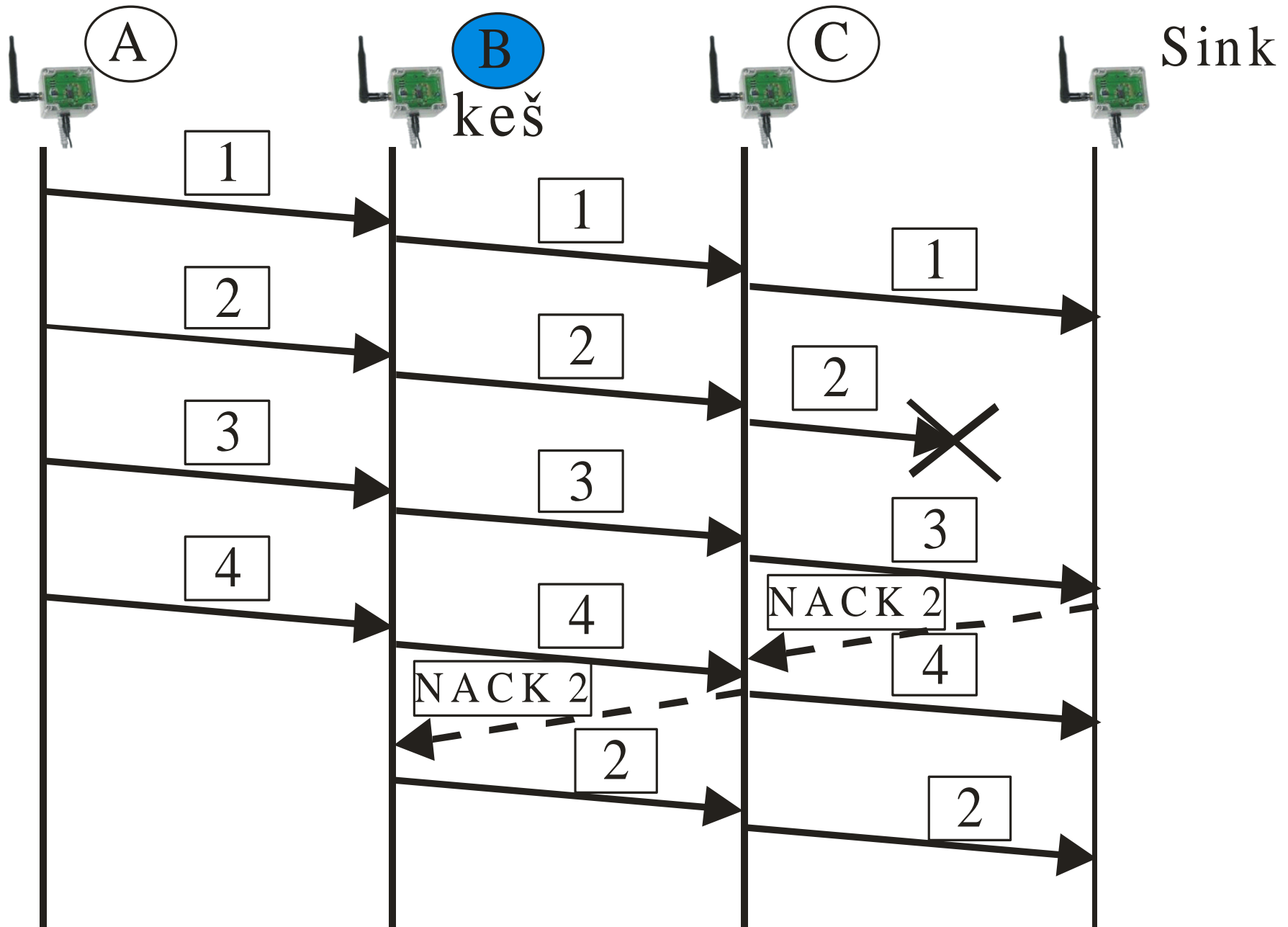
9.5 RMST-Reliable Multi-Segment Transport

- Protokol RMST je **jedan od prvih BSM protokola** transportnog sloja
- Glavni cilj RMST-a je da omogući pouzdan prenos od predajnika do prijemnika tkz. ***end-to-end* pouzdanost**.
- RMST je izgrađen na **principima usmerenog difuzionog protokola** i koristi neke od njegovih funkcionalnosti.
- RMST je dizajniran kao **neka vrsta filtera** koji se može postaviti na usmereni difuzioni protokol.
- RMST obezbeđuje **dve od tri osnovne funkcije** potrebne za funkcionisanje protokola transportnog nivoa:
 - a. **pouzdan prenos podataka**
 - b. **multipleksiranje/demultipleksiranje**.
- Multipleksiranje i demultipleksiranje se izvršavaju na **izvornim SČ i sink-u** respektivno.
- RMST **kontroliše kretanje podataka kroz celu rutu**, od SČ do *sink*-a, i poseduje mehanizme **kojima rešava eventualnu pojavu greške** na ruti.
- RMST **koristi keširanje** u mreži i **obezbeđuje garantovanu isporuku** paketa podataka koji dolaze od SČ.

9.6 RMST-Reliable Multi-Segment Transport

- RMST koristi **usmereni difuzioni mehanizam** rutiranja za određenu rutu između predajnika i prijemnika.
- Upravo zato, podrazumeva se da paketi uvek, od SČ do *sink*-a, **idu istom rutom**, osim ako ne dođe do otkaza nekog SČ u ruti.
- Ako se to desi, pretpostavlja se da će **protokol usmerene difuzije izvršiti promenu rute** i omogućiti nesmetan prolaz paketa.
- Na osnovu ove pretpostavke, RMST ima **dva načina rada**:
 - 1. Režim bez keša**: sličan radu standardnih protokola koji rade u transportnom sloju, gde samo predajni SČ i prijemni SČ igraju ulogu u obezbeđivanju pouzdanosti. Zato se gubici paketa detektuju samo kod sink-a, koji putem NACK zahteva, koji upućuje izvornom SČ, traži da se ponovi slanje paketa. Prednost ovog režima je u tome što ne zahteva učešće, dodatnu obradu, memorisanje i potrošnju energije
 - 2. Režim sa kešom**: prolazni SČ koji se nalaze u ruti vrše keširanje primljenih paketa u svojoj memoriji. Na taj način smanjuju se troškovi prenosa paketa prilikom eventualnih retransmisija paketa u *end-to-end* prenosu.

9.6 RMST-Reliable Multi-Segment Transport



9.6 lwIP - Light Weight Internet Protocol

- Dizajn gotovo svih Internet protokola (TCP/IP skup) zasnovan je na principu **praćenja paketa od predajnika do prijemnika** (*end-to-end*).
- IP je odgovoran za **adresiranje izvora i destinacije**, kao i za **rutiranje datagrama/paketa od izvora do destinacije** kroz jednu ili više IP mreža.
- IP **definiše format paketa** i obezbeđuje adresni sistem koji **identifikuje izvor i destinaciju** i pružanje **usluge logičke lokacije destinacije**.
- Mrežna infrastruktura koja povezuje predajnik sa prijemnikom se smatra **standardno nepouzdanom** na bilo kom pojedinačnom mrežnom segmentu, uređaju ili prenosnom mediju, i pretpostavlja se da je ona **dinamična u smislu dostupnosti veza i čvorova**.
- **Ne postoji centralno sredstvo** za praćenje/merenje performansi mreže
- U cilju smanjenja kompleksnosti mreže, inteligencija u mreži se namerno uglavnom nalazi u **krajnjim čvorovima prenosa podataka**.
- Fokus implementacije lwIP TCP/IP je **da smanji upotrebu resursa** dok još ima TCP u punoj veličini.
- U tradicionalnom TCP/IP *stacku*, protokoli su **strukturirani na slojevit** način, pri čemu se svaki **sloj protokola** bavi različitim **delom komunikacionog problema**.

9.7 lwIP - Light Weight Internet Protocol

- Ovakav dizajn dovodi do situacije u kojoj će komunikacioni zahtevi između slojeva protokola **degradirati njegove ukupne performanse**
- U većini TCP/IP implementacija postoji **stroga podela** između sloja aplikacije i nižih slojeva protokola u kojima se protokoli nižeg sloja **implementiraju kao sastavni deo jezgra operativnog sistema.**
- OS koji se koriste u malim (*embedded*) sistemima, koji koriste lwIP, **uglavnom ne postavljaju strogu podelu** između jezgra i sloja aplikacije
- Postoji **jednostavnija šema** za komunikaciju između jezgra OS i aplikacionog sloja koja se izvršava **pomoću zajedničke memorije i mehanizma koji upravlja baferima** koji koriste niži slojevi.
- Ovo povećava efikasnost pošto **aplikacioni sloj može ponovo koristiti bafere**, a takođe i **čitati/pisati direktno u te bafere**, čime se izbegava potreba **da se podaci kopiraju** i time nepotrebno troši RAM memorioja.
- U lwIP-u, svaki protokol se **implementira kao zasebni**, pojedinačni modul **sa nekoliko zasebnih funkcija**
- lwIP se sastoji od nekoliko modula **koji omogućuju direktnu implementaciju skupa TCP/IP protokola (IP, ICMP, TCP i UDP)** plus druge module koji daju podršku implementaciji drugih protokola.

9.7 lwIP - Light Weight Internet Protocol

- Ovi moduli podrške formiraju u okviru OS jedan poseban sloj tkz. **sloj emulacije** koji omogućuje **pod sisteme za upravljanje baferima i memorijom, funkcije mrežnog interfejsa i funkcije za kontrolu greške.**
- lwIP takođe sadrži i **apstraktni API** koji ga čini kompatibilnim sa **različitim platformama i OS.**
- lwIP koristi model procesa gde svi **protokoli žive u jednom procesu.**
- Aplikacioni programi mogu ili da žive u samom procesu lwIP ili da budu **u odvojenim procesima** u kojima se komunikacija između TCP/IP steka i aplikativnih programa vrši pomoću apstraktnog API-ja.
- Glavna prednost njegovog pristupa je to što **omogućava prenos lwIP-a preko različitih operativnih sistema.**
- lwIP je **široko korišćen TCP/IP stack otvorenog koda** dizajniran za male ugrađene (*emdedded*) sisteme.
- Osnovni cilj lwIP je da ublaži jako zahtevne resurse koji su neophodni za implementaciju protokola iz TCP/IP skupa.
- To omogućava da je lwIP pogodan za upotrebu u ugrađenim sistemima sa svega **10 kB slobodne RAM memorije** i oko **40 kB ROM memorije za programski kod.**

9.7 lwIP - Light Weight Internet Protocol

➤ lwIP protokol podržava veliki broj protokola iz TCP/IP skupa i to:

Internet sloj

- ✓ IP (*Internet Protocol*) uključujući paketno prosleđivanje preko više mrežnih interfejsa
- ✓ ICMP (*Internet Control Message Protocol*) za održavanje mreže i debugovanje
- ✓ IGMP (*Internet Group Management Protocol*) za *multicast* upravljanje

Transportni sloj

- ✓ UDP (*User Datagram Protocol*) uključujući eksperimentalno proširenje UDP-lite
- ✓ TCP (*Transmission Control Protocol*) sa kontrolom zagušenja, procena RTT-a i brzog oporavka/brzog ponovnog prenosa

Aplikacioni sloj

- ✓ DNS (*Domain names resolver*)
- ✓ SNMP (*Simple Network Management Protocol*)
- ✓ DHCP (*Dynamic Host Configuration Protocol*)

Sloj veze

- ✓ PPP (*Point-to-Point protocol*)
- ✓ ARP (*Address Resolution Protocol*) za Ethernet

Hvala na pažnji !!!



Pitanja

? ? ?