5. Транспортни слој

Транспортни слој у референтним *OSI* и *TCP/IP* моделима рачунарских комуникација задужен је за пренос података апликација између две комуникационе стране. Он чини слој између слоја апликације и мрежног слоја.

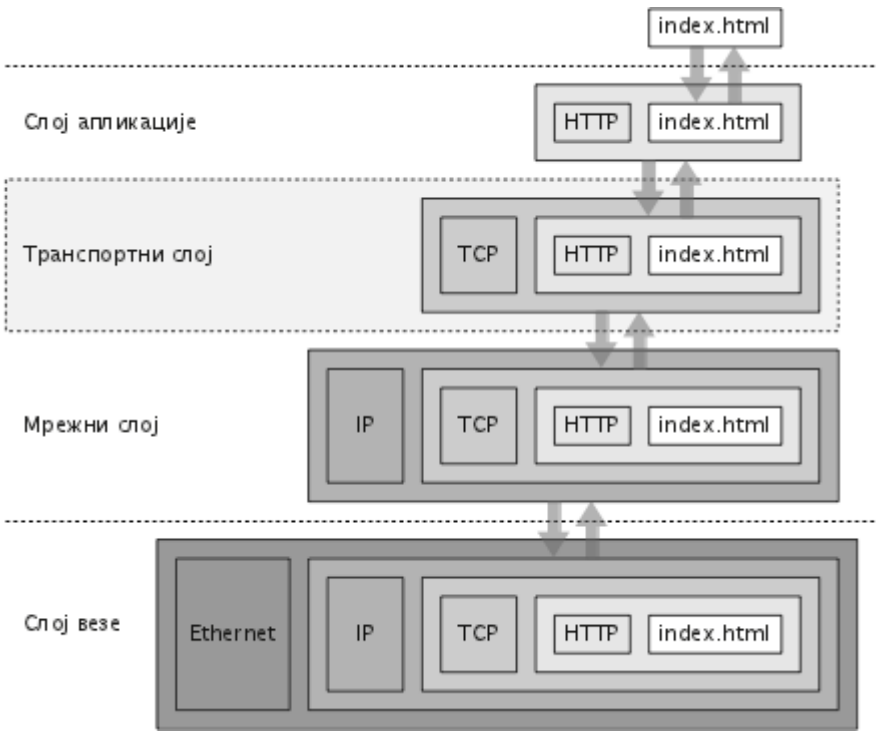
У савременим рачунарским мрежама постоје два основна типа података у зависности од критичног захтева при њиховом преносу:

1. подаци осетљиви на грешке
2. подаци осетљиви на кашњење

У складу са овим захтевима на транспортном слоју постоје два основна протокола за пренос података:

1. протокол за контролу преноса (енгл. *Transmission Control Protocol*, *TCP*)
2. протокол корисничких датаграма (енгл. *User Datagram Protocol*, *UDP*)

Конкретна улога транспортног слоја јесте да прихвати податке од апликације на страни пошиљаоца и достави их апликацији одредишта старајући се о преносу, контроли и исправљању грешака при преносу и о гарантовању испоруке.



Слика 4-1. Пример комуникације протокола транспортног и суседних слојева

Након прихватања података од слоја апликације транспортни слој има задатак да их преведе у οблик погодан за транспорт. Функционалности које се углавном очекују од транспортног слоја су:

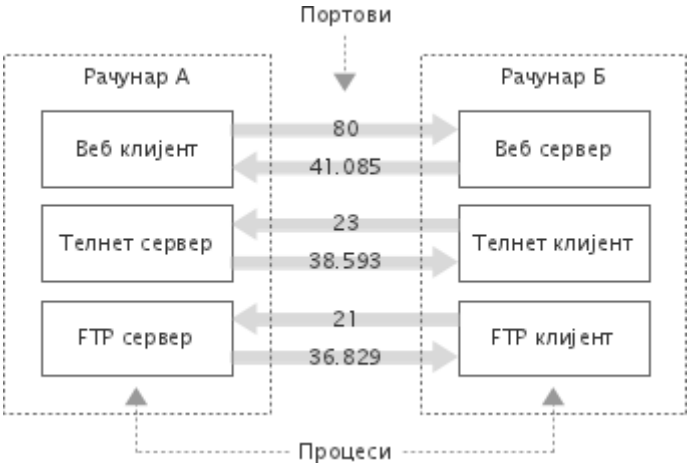
1. Остваривање виртуелне везе за пренос података.
2. Превођење података у (углавном бинарни) формат погодан за пренос.

3. Сегментација података ради ефикаснијег искоришћења комуникационог канала.
4. Испорука података апликацији на страни примаоца у идентичном облику у ком су послати.
5. Омогућавање оптималне брзине преноса података у складу са пропусном моћи и учесталошћу грешака на комуникационом каналу и прихватној моћи примаоца.

Ентитете апликативног слоја је могуће адресовати путем портова по сопственом избору с тим да је за стандардизацију у овој области задужена организација *IANA*. Ово тело на захтев произвођача софтвера анализује оправданост за званичним додељивањем слободних портова (у складу са распрострањеношћу сервиса за који се порт захтева) и додељује захтевани порт уколико је доступан. Пре званичног додељивања порта сервису потребно је остварити одређену општост, односно популарност сервиса.

5.1. Портови и утичнице

Иако је адресовање подразумевана улога мрежног слоја, транспортни слој поседује интерни систем адресовања чија је адресна јединица порт. Порт је одређен 16-битним нумеричким параметром и његова је улога да одреди изворни/одредишни ентитет апликативног слоја (апликацију) од кога потичу подаци, односно коме треба испоручити податке.



Портови се могу поделити на привилеговане, регистроване и динамичке (или краткотрајне). Привилеговани портови се налазе у опсегу бројева од 0 до 1023 и право на њихово отварање углавном има само оперативни систем или процеси које је покренуо администратор система. На привилегованим портовима се налазе најчешће коришћени сервиси (*FTP, SSH, Telnet, DNS* и други). Регистровани портови се крећу у опсегу од 1024 до 49151 и на њима се подразумевано користе сервиси новијег датума. Динамички или краткотрајни портови крећу се у опсегу од 49152 до 65535 и њих није могуће регистровати а углавном служе за клијентске компоненте клијент/сервер софтвера.

Појам и функционалност утичница (енгл. *socket*) први пут се појављују у верзији 4.2 *BSD UNIX* оперативног система, а данас су оне де факто стандард за све популарне оперативне

системе који подржавају мрежне функционалности. Утичнице представљају композитне адресне јединице на нивоу транспортног и мрежног слоја. Саставни делови утичница су:

1. *IP* адреса изворишта
2. Порт изворишта
3. Протокол транспортног слоја
4. Порт одредишта
5. *IP* адреса одредишта

Подршка за утичнице у оперативним системима најчешће се реализује помоћу готових системских библиотека. Неке од најпопуларнијих библиотека овог типа су *Berkeley socket* за *UNIX* оперативне системе и *Winsock* за оперативне системе компаније Мајкрософт. Осим употребе у рачунарским мрежама утичнице се могу користити и код апликација које се извршавају на локалном рачунару. На пример, *X Window System* графички систем на *UNIX* платформи захтева коришћење утичница да би функционисао.

5.2. TCP - протокол за контролу преноса

Протокол за контролу преноса (енгл. *Transmission Control Protocol, TCP*) података је протокол задужен за рад са подацима у транспортном слоју. У питању је протокол са успоставом везе дизајниран да за податке користи низове бајтова и да обезбеди поуздан пренос података у оба смера (*full-duplex*). Овај протокол је погодан за рад на комуникационим каналима високе поузданости а показује слабије перформансе на комуникационим каналима са честим оштећењем података током преноса.

Протокол за контролу преноса један је од најчешће коришћених протокола на транспортном слоју када су у питању Интернет и класичне локалне мреже. Овај протокол је већ годинама у употреби а разлог томе је пре свега оптималан рад на Етернет технологији. Иницијално је дефинисан у документу *RFC793* а касније је његова спецификација неколико пута мењана у складу са новим потребама и могућностима рачунарских мрежа. Неки параметри протокола за контролу преноса одређени су у складу са ограничењима комуникационих канала (на пример, *MTU* параметар је ограничен на 536 бајтова за везе са странама које нису у локалној мрежи) али су се временом показали неадекватним услед значајних унапређења брзина и поузданости комуникационих канала.

Савремена окружења за развој дистрибуираних апликација најчешће имају развијену подршку за коришћење протокола за контролу преноса или протокола вишег нивоа који користе његове услуге. То значи да овај протокол представља *de facto* стандард за развој дистрибуираног софтвера који захтева поуздан пренос података. Неки од најпопуларнијих сервиса интернета (Веб, електронска пошта и други) подразумевају коришћење овог протокола.

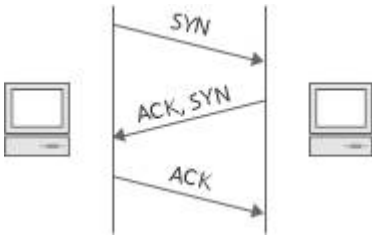
5.2.1. Управљање везом

При коришћењу протокола за управљање преносом две стране морају да успоставе везу између себе као предуслов за даљу размену података. За успостављање везе, као и за њен раскид, користе се сегменти без корисничког садржаја али са укљученим индикаторима *SYN*, *FIN* и *ACK*.

Успостављање везе се врши путем следећих корака:

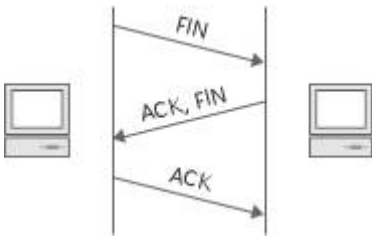
1. Клијент серверу шаље сегмент са *SYN* индикатором који садржи број порта сервера на који клијент жели да се повеже и са *ISN*-ом клијента.
2. Сервер одговара на *SYN* захтев клијента сегментом који садржи *ACK* индикатор са *ISN*-ом клијента увећаним за један. Сегмент такође садржи *SYN* индикатор сервера са његовим *ISN*.
3. Клијент одговара на *SYN* захтев сервера шаљући сегмент са *ACK* индикатором који садржи *ISN* сервера увећан за један.

Ова три корака се називају “руковање” (енгл. *handshake*) и уколико не дође до грешке у њима, веза је успостављена. Страна која иницира успостављање везе извршава активно успостављање везе (енгл. *active open*) док страна која прихвата успостављање везе извршава пасивно успостављање везе (енгл. *passive open*).



Слика 5.2.1.-1. Пренос сегмената за успостављање везе

Једном успостављена веза остаје активна док год се не захтева њен раскид или док једна од страна не изгуби евиденцију о њој (нпр. ресетовањем рачунара). То значи да су у периодима када се веза не користи за пренос података могући прекиди на свим нижим слојевима (укључујући и физички).



Слика 5.2.1.-2. Пренос сегмената за раскид везе

Код раскида успостављене везе потребно је да обе стране добију информацију да је веза раскинута и да даљи пренос података није могућ. Раскид везе може иницирати свака од страна, без обзира на то која страна је иницирала успостављање везе. За раскид везе се користе *FIN* индикатори у *TCP* заглављу а редослед сегмената је следећи:

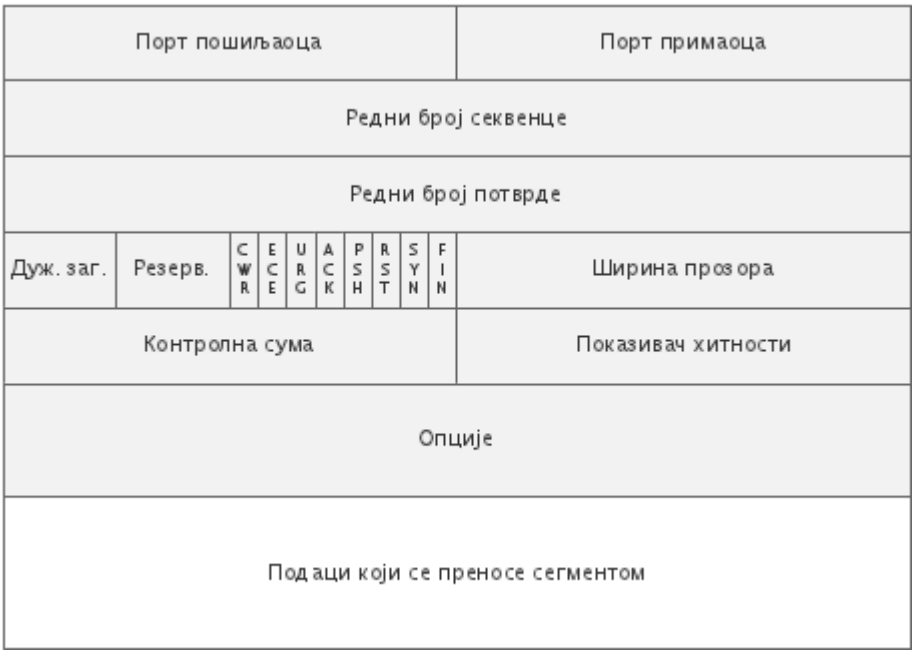
1. Страна која иницира раскид шаље *TCP* сегмент са укљученим *FIN* индикатором.
2. Прималац одговара сегментом са укљученим *ACK* индикатором.
3. За потпуни раскид везе прималац шаље сегмент са укљученим *FIN* индикатором.
4. Иницијатор раскида шаље одговор у виду сегмента са укљученим *ACK* индикатором.

Страна која иницира раскид везе извршава активан раскид везе (енгл. *active close*) док страна која прихвата раскид везе извршава пасиван раскид везе (енгл. *passive close*). Уколико једна страна иницира раскид везе (пошаље сегмент са *FIN* индикатором и прими сегмент са *ACK* индикатором) а друга задржи везу успостављеном таква веза се назива полу-раскинутом (енгл. *half-close*) и њоме се надаље могу слати само подаци у једном смеру.

5.2.2. Сегментација података

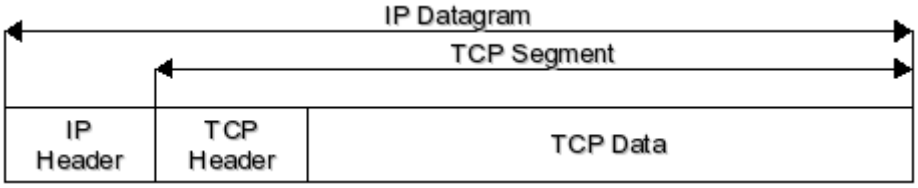
Протокол за контролу преноса на страни пошиљаоца од слоја апликације преузима податке у виду низа битова условно неограничене дужине. Међутим, за потребе преноса добијени низ података се дели у сегменте мање битске дужине. Ови сегменти се преносе појединачно а на пријемној страни се спајају у низ идентичан послатом. Добијени низ битова се испоручује слоју апликације.
Иако се протокол за контролу преноса може користити у спрези са различитим протоколима осталих слојева, он све функције неопходне за поуздан пренос података обезбеђује самостално. За сваки послати сегмент података захтева се потврда примаоца о успешном пријему без грешке. Слање оштећених и изгубљених сегмената се понавља све док се не обави успешан пренос.

Сегменти представљају основну јединицу података протокола за контролу преноса. Они се састоје од заглавља (које генерише и интерпретира сам протокол) и података који се преносе (које генерише/преузима слој апликације). Апликативни подаци нису обавезан део сегмента и изостављени су код контролних сегмената (на пример сегмената задужених за успостављање и прекид везе).



Слика 4.3.2-1. Структура сегмента протокола за контролу преноса

Заглавље сегмента се састоји од поља фиксне дужине која садрже информације везане за протокол. Битска дужина заглавља је пет пута по 32 бита, уколико нису укључене опције. Битска дужина целокупног сегмента једнака је битској дужини заглавља (уколико се сегментом не преносе подаци апликације) или збиру битске дужине заглавља и битске дужине података добијених од слоја апликације.



Слика 4.3.2-2. Сегменти *TCP* протокола се инкапсулирају у *IP* датаграме

Осим основног дела заглавље може садржати и додатне опције везане за протокол. Свака опција заглављу додаје реч од 32 бита. Уколико опција не садржи довољно података да испуни 32 бита, преостали битови се допуњавају нулама. Неке од најважнијих опција *TCP* протокола су *MSS* (*Maximum Segment Size*), *WSOPT* (*Window Scale Option*), *SACK* (*Selective ACK*) и *SACK Permitted*. Садржај опција такође улази у контролну суму сегмената (*TCP checksum*).

5.2.2.1. Поузданост и перформансе

Перформансе протокола за контролу преноса углавном су знатно слабије у поређењу са протоколима који раде без успостављања везе и не нуде поуздан пренос. Такође, овај протокол не подржава *broadcasting* и *multicasting* већ омогућава комуникацију између искључиво две стране. Међутим, главна карактеристика *TCP* протокола је поузданост. С обзиром на то да *IP* протокол (протокол мрежног слоја) не гарантује поздан пренос података, поузданост *TCP* протокола се остварује путем следећих правила:

- Податке које апликација доставља транспортном слоју *TCP* дели у сегменте које шаље појединачно. На тај начин се смањује јединица над којом се врши контрола и на тај начин смањује могућност и цена исправљања грешке.
- *TCP* захтева потврду да је сваки од послатих сегмената испоручен. Подаци се након слања чувају у излазном баферу до добијања потврде о пријему. Уколико потврда о испоручивању не стигне у одређеном временском року, сегмент се шаље поново.
- *TCP* елиминише исте сегменте које је мрежни слој грешком доставио два или више пута.
- *TCP* реорганизује примљене сегменте по изворном редоследу без обзира на редослед којим их мрежни слој доставља.
- *TCP* приликом слања сегмената генерише контролне параметре везане за заглавље и садржај. Уколико приликом преноса сегмента дође до измена заглавља или садржаја, ове суме указују на њих и захтева се поновно слање сегмента.
- *TCP* прилагођава фреквенцију слања сегмената прихватној моћи примаоца чиме спречава одбацивање сегмената (и непотребно оптерећење комуникационог канала) и смањује могућност грешке. Добијене корисничке податке (тј. податке које доставља апликативни слој) *TCP* протокол третира као низ бајтова. Уколико добијени низ бајтова прелази највећу дозвољену величину сегмента (*MSS*, *Maximum Segment Size*), низ се дели и шаље са више прихватљивих сегмената.

Параметар *MSS* свака страна доставља супротној у оквиру сегмента са *SYN* индикатором (при упостављању везе) а, уколико та опција сегмента није дата експлицитно, користи се подразумевана системска вредност од 536 бајтова. Укупна величина *TCP* сегмента (укључујући заглавље и податке) са *MSS*-ом од 536 бајтова износи 556 бајтова. Величина *IP* датаграма оваквог пакета би износила 576 бајтова (556 бајтова *TCP* сегмента + 20 бајтова *IP* заглавља). Дакле, *IP* датаграм са оваквом величином *MSS*-а садржи 536 бајтова (93%) корисничких података и 40 бајтова (7%) података везаних за *TCP* и *IP* протоколе. То значи да је у оваквом случају комуникационим каналом пропусне моћи 100Mb/s у једној секунди могуће пренети 93Mb корисничких података. Уколико вредност *MSS*-а повећамо на 3.960 бајтова укупна величина *IP* датаграма би износила 4.000 бајтова (величине *TCP* и *IP* заглавља остају исте по сегметну и датаграму). У том случају би однос корисничких података и података везаних за протоколе износио 99:1. Таква вредност *MSS*-а би на комуникационом каналу пропусне моћи 100Mb/s омогућила пренос 99Mb корисничких податак у секунди. Из овога би се могло закључити да већа вредност *MSS*-а омогућава веће брзине преноса тј. ефикасније коришћење комуникационог канала. Међутим, овакав закључак је тачан само у случају када комуникациони канал и интерфејси ка њему омогућавају пренос података без грешака. У случајевима када се јављају грешке при преносу (што захтева поновно слање оштећених сегмената) високе вредности *MSS*-а могу имати супротан ефекат или чак у потпуности онемогућити комуникацију. Опција *MSS* стоји у блиској вези са перформансама *TCP* протокола. Негативан утицај на перформансе овог протокола може имати и потреба да се за сваки достављени сегмент засебно достави потврда о испоруци.

Слање *ACK* сегмента за сваки успешно примљени сегмент у одређеним случајевима може представљати непотребно оптерећење комуникационог канала. Међутим, пошиљалац може слањем пакета са *SACK permitted* опцијом овластити клијента за периодично слање сегмента

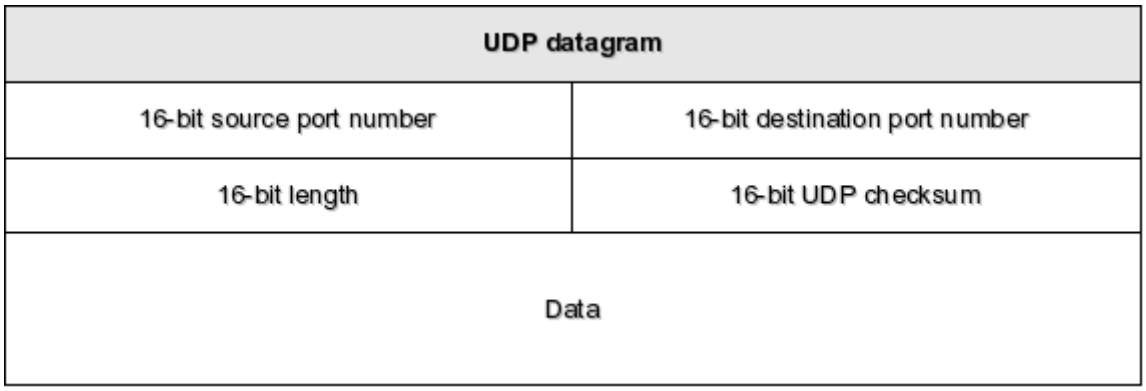
са *SACK* опцијом - сегмента који садржи обавештење о више успешно примљених сегмената. На тај начин се растерећује комуникациони канал на рачун излазног бафера пошиљаоца. Основна потврда о испоруци се шаље у виду сегмента са укљученим АЦК индикатором. Помоћу опције *SACK* (*Selective ACK*) могуће је послати потврду за више примљених пакета у оквиру једног сегмента. За ову опцију је потребно да и друга страна дозволи њено коришћење слањем пакета са *SACK Permitted* опцијом.

5.3. UDP - протокол корисничких датаграма

Протокол корисничких датаграма (енгл. *User Datagram Protocol, UDP*) поред протокола за контролу преноса представља један од најчешће коришћених транспортних протокола Интернета и локалних рачунарских мрежа. Насупрот протоколу за контролу преноса, протокол корисничких датаграма не омогућава поуздан пренос података путем остваривања виртуелне везе, контроле грешака, контроле редоследа сегмената и не прилагођава брзину слања података пријемној моћи одредишта. Недостатак ових функционалности чини протокол корисничких датаграма једноставнијим од протокола за контролу преноса, али и протоколом који не гарантује поуздан пренос података.

Међутим, намена протокола корисничких датаграма није поуздан пренос података, већ пренос са што мањим временским неслагањима између генерисања података на страни извориште и пријема података на одредишту. Главне примене овог протокола су код проточног преноса гласа и видео материјала (Интернет телефонија, видео конференције, рачунарске игрице и слично). Предност протокола корисничких датаграма у односу на протокол за контролу преноса је могућност емисионог слања података, односно истовременог слања података свим члановима мреже. Јединица за пренос података протокола корисничких датаграма је датаграм. Структура датаграма овог протокола је знатно једноставнија од структуре сегмената протокола за контролу преноса јер је у њима изостављена већина контролних информација.

Недостатак контролних информација чини *UDP* протокол знатно ефикаснијим у смислу мањег оптерећења комуникационог канала контролним подацима и мањег оптерећења примаоца датаграма у смислу његове једноставније обраде.



Слика 4.4-1. Структура датаграма протокола корисничких датаграма

Сва поља у заглављу датаграма *UDP* протокола имају дужину од 16 битова а у њих спадају:

- Број порта на изворишту (енгл. *Source port number*) - одређује порт преко кога се комуникација врши на страни пошиљаоца.
- Број порта на одредишту (енгл. *Destination port number*) - одређује порт преко кога се комуникација врши на страни примаца.
- Дужина (енгл. *Length*) - δитска дужина података које датаграм носи.
- Контролна сума (енгл. *Checksum*) - контролна сума заглавља и пакета.

Неотпорност *UDP* протокола на грешке при преносу, вишеструко достављање истих датаграма или гуδитак података, достављање података у измењеном редоследу и слично могуће је надоместити функционалностима у апликативном слоју. Неке апликације које користе *UDP* протокол примењују овакав приступ (на пример *TFTP* сервис). Међутим, коришћењем *UDP* протокола апликације угавном очекују максималне перформансе преноса δез оδзира на грешке и додатни системи за исправљање грешака δи угрозили нормалан рад поменутих апликација.