МОНГОЛ УЛСЫН ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ

МЭДЭЭЛЭЛ ХОЛБОО ТЕХНОЛОГИЙН СУРГУУЛЬ

Шарав Мөнхтулга

J.SA13D023

Пакет үүсгэгч програмыг хөгжүүлэх нь

Мэргэжил: Компьютерийн Системийн Хамгаалал

Систем хамгааллын төсөл

Улаанбаатар хот

2017он

Гарчиг

Хүснэгтийн жагсаалт

Зургийн жагсаалт

Бүлэг 1

Ерөнхий зүйл

1.1 Зорилго

Энэхүү төслийн ажлын зорилго нь интернетийн сүлжээн дээгүүр өнгөрч буй өгөгдөл дамжуулалт,тэдгээрийн мэдээлэл,custom пакет үүсгэх ажиллагааг судлан үүн дээр тулгуурлан хэрэглэгчийн интерфэйстэй програмыг нэмж хөгжүүлэх юм.

1.2 Зорилт

1. Интернет сүлжээн дээгүүр өгөгдөл хэрхэн дамждаг мөн тэдгээр өгөгдлүүдийг сүлжээний түвшин бүр дээр ямар хэлбэртэй дамждаг болохыг судлах.
2. Сокет болон протокол дата юнит ажиллагааг судлах.
3. Пакет үүсгэгч програмуудыг судлах, давуу болон сул талуудыг тодорхойлох.
4. Хэрэглэгчийн интерфэйсийг зохиомжлон пакет үүсгэгч програмыг бүтээх.
5. Өмнөх хөгжүүлэгчийн хийсэн бүтээлийг сайжруулах.

1.3 Удиртгал

Одоо ихэнх сүлжээнүүд сувгийн-холболттой болсон бөгөөд энэ нь илүү үр ашигтай болсон.Үүгээр дамжин өнгөрч байгаа мэдээллүүд пакет хэлбэртэй болж харагддаг бөгөөд энэ нь хүний судсаар өнгөрч буй улаан эстэй тун их төстэй үүрэгтэй юм.

Бүлэг 2

Онолын хэсэг

2.1 Сүлжээний тухай

Сүлжээ нь өөртөө холбогдсон төхөөрөмжүүдийг өөр хоорондоо өгөгдлөө солилцох боломж олгодог. Сүлжээнд багтаж байгаа гол цэгүүд нь утастай мөн утасгүй гэсэн орчинд холбогдоно.

Өгөгдлийг үүсгэх, чиглүүлэх мөн төгсгөх үйлдэл хийдэг сүлжээний төхөөрөмжүүдийг гол цэгүүд гэж нэрлэдэг. Гол цэгүүд нь хувийн компьютерүүд, утаснууд, серверүүд гэх мэт сүлжээний техник хангамжууд буюу хостууд байж болно. Хоёр төхөөрөмж нэг нь нөгөө төхөөрөмжрүүгээ мэдээлэл дамжуулах боломжтой болсон бол үүнийг сүлжээ тогтлоо гэж хэлж болно.

Компьютерийн сүлжээнд дохиог зөөхдөө өөр өөр дундын дамжууллын орчныг ашигладаг байна. Үүнд хатуу, шингэн, хий мөн плазм зэрэг олон янзын төлөв байж болно. Энэхүү сүлжээ нь WWW, видео, тоон аудио хандалт мөн хэрэглээний болон хадгалалтын серверүүд, принтерүүд, имэйл, мессеж програм гэх мэт өөр асар их хэрэглээний боломжийг хүмүүс бидэнд олгодог.

Ихэнхи тохиолдолд програмын-тусгай харилцааны протоколууд нь бусад ерөнхий протоколуудтай давхарга болон угсрагддаг. Энэ нь пакет свичинг сүлжээний өгөгдөл дамжих үндэс нь болдог.

Сүлжээ нь дараах үүргүүдтэй:

- Харилцаа холбоог бий болгох. Сүлжээг ашиглан хүмүүс нь и-мэйл, мессенжэр, чаат өрөө, телефон, видео телефон яриа, видео хурал зэргийн хэрэгслүүдээр бие биентэйгээ үр ашигтайгаар, хялбараар харилцаж болно.

- Техник хангамжийг хамтарч хэрэглэх. Сүлжээнд хамрагдах компьютер бүр нь тус сүлжээнд холбоотой техник хангамжийг хэрэглэх боломжтой. Жишээлбэл, хамт хэрэглэх байдлаар тохируулсан принтерээр документ хэвлэх.

- Файл, өгөгдөл, мэдээллийг хамтарч хэрэглэх (шэйр). Сүлжээний орчинд баталгаажуулсан хэрэглэгч нь тус сүлжээний бусад компьютерүүд дахь өгөгдөл, мэдээлэлд хандах боломжтой. Хамт хэрэглэх байдлаар тохируулсан хадгалах төхөөрөмжүүд дээрх өгөгдөл, мэдээлэлд хандах явдал нь олон сүлжээний чухал онцлог болдог.

- Програм хангамжийг хамтарч хэрэглэх. Сүлжээнд холбогдсон хэрэглэгчид нь холын зайд орших компьютерүүд дээр програм ажиллуулах боломжтой.

- Мэдээллийг хадгалах.

- Аюулгүй байдал.

- Хурд.

2.2 Сувгийн холболттой сүлжээ

Пакет свичинг сүлжээ нь сүлжээний өөр нэгэн үндсэн загвар болох хэлхээ свичинг сүлжээнээс өөр арга юм. Хэлхээ свичинг сүлжээ нь холболтын session ашиглах үед гол цэгүүдийн дундах тогтмол хүлээлт мөн тогтмол бит хэмжээ бүхий тусгай холболт шаардсан хязгаарлагдмал холболтын арга юм. Мөн өгөгдөл дамжаагүй нөхцөлд хэлхээ свичинг сүлжээг холболтын хугацааны нэгж бүр дээр төлбөр төлдөг гэж үздэг бол пакет свичинг сүлжээг мэдээллийн нэгж бүр дээр төлбөр төлдөг гэж тодорхойлж болно. Энэ нь сүлжээг үр ашигтайгаар ашиглах ойлголттой холбогдох болно.

Пакет свичинг сүлжээний нэвтрүүлэх чадамжийг нэмэгдүүлэх маш олон янзын протоколууд, алгоритмууд болон бодлогууд байдаг. Өнөө үед бид үндсэн хоёр пакет свичинг протоколуудыг хэрэглэж байна. Эхнийх нь холболтгүй пакет свичинг буюу бидний мэдэхээр датаграм свичинг, харин дараагийнх нь холболт-хандлагатай пакет свичинг буюу бидний мэдэхээр виртуал хэлхээ свичинг юм.

Датаграмын үед пакет бүр дээр бүрэн гүйцэт хаяглалт болон чиглүүлэлтүүдийг нэг бүрчлэн боловсруулсан байна. Эдгээр пакетууд нь өөр өөр замаар дамжигдах эсвэл ямар нэгэн боломжит чиглүүлэлтүүд хийгдэх боломжтой. Виртуал хэлхээ свичинг нь

пакет свичинг сүлжээ яг л хэлхээ свичинг сүлжээ шиг ажиллах үеийг хэлдэг. Төхөөрөмжүүдийн дундах холболтууд нь пакетыг дамжуулахдаа тусгай зориулсан гол цэгүүд эсвэл чиглүүлэлтүүдийг ашигладаг.

Пакет горимын холболт нь завсрын илгээгч гол цэгүүд(пакет свичүүд эсвэл рүүтерүүд)гүйгээр ч хэрэгжиж болно. Бүхий л пакет горимын холболтонд сүлжээний нөөцүүд нь статистикийн мультипекс эсвэл динамик сүлжээний өргөний хуваарилалтаар удирдагдаж байдаг. Статистикийн мультиплекс нь пакет свичинг болон бусад хадгалах-дамжуулах буферын нэвтрүүлэлтийн өөр өөр хоцролтоууд мөн дамжуулалтын үеийн ролт/гаралтын хэмжээг тооцоолох зэрэг дээр хэрэглэгддэг.Энгийнээр дундын сүлжээний гол цэгүүд болон мультиплекс хийгчээр илгээгдсэн пакетууд бүр эхлэж-ор, эхлэжгар зарчимтай буферыг ашигладаг.

Өөр нэг арга нь пакетууд fair queuing(шударга дараалал), traffic shaping(хөдөлгөөн засах), weighted fair queueing(жигнэсэн шударга дараалал) мөн leaky bucket(нэвтрүүлдэг хувин) гэх мэт зарим алгоритмын хуваарийн дагуу дамжуулагдаж болно. Мөн дундын физик дамжих орчин(радио эсвэл 10BASE5 шиг)-ы тохиолдолд пакетууд нь multiple access(олон хандалт) схемийн дагуу дамжиж болно.

Өгөгдлийг тусдаа жижиг, жижиг хэсгүүд буюу пакетууд болгоод тухайн пакет бүр дээрх эцсийн цэгийн хаяг дээр үндэслэн дамжуулдаг. Хүлээн авсан үед илгээмжийг бүрдүүлхийн тулд пакетуудыг таарсан дарааллын дагуу цуглуулдаг. Харин хэлхээ-свичинг сүлжээний холболтын үед тусгай зориулсан цэгээс-цэгт шугам хэрэгтэй байдаг.

Пакетыг ашигласнаар сүлжээний үр ашгийг илүү ихэсгэж, ашиглалтын найдвартай байдал мөн ижил сүлжээн дээр олон програм зохицож ажиллах зэрэг боломж олгодог. Пакетууд нь payload болон header гэсэн хэсгүүдээс бүрддэг. Толгой хэсэг буюу header нь payload ачаа хэсгийг зөөх үед сүлжээний төхөөрөмжүүдийн ашиглах мэдээллүүд байна.

Практикт хэрэглэгдэж буй системүүд нь OSI моделээс өөр байх тохиолдол их бий. Жишээ нь, Интернэтийн хэрэглэдэг TCP/IP модел нь арай өөр бүтэцтэй. Мөн OSI систем доторх дэд давхаргууд академи болон үйлдвэрлэлийн нийтлэлүүдэд өөрийн багтах давхаргаасаа илүүтэйгээр яригдах нь бий. Жишээ нь, MAC (Media access control) буюу орчны хандалт удирдлагын давхарга нь data-link layer буюу өгөгдөл холболтын давхаргын дэд давхарга боловч тусдаа давхарга болон яригдах нь элбэг. Сүлжээний давхаргуудыг ашиглан програм зохиогч өөрийн програмыг бүтээх бө- гөөд програмын энэхүү ажиллагаа нь ихэвчлэн OSI загвар дээр хэрэгждэг. Учир нь OSI загвар нь үзүү-

лэнгийн шинж чанартай бол үүний үйл ажиллагааг хэрэгжүүлдэг загвар бол TCP/IP загвар юм.

2.2.1 Сувгийн холболттой сүлжээний үүсэл

Өгөгдлийг жижиг блокуудад шилжүүлэх санааг анх 1960 оны эхэн үед Paul Baran гаргаж ирсэн. Англид байрлах Үндэсний Физик Лаборатор(NPL)-д Donald Davies мөн хамааралгүйгээр ойролцоо санааг хөгжүүлж байсан. Baran аюулгүй харилцааны сүлжээг бий болгохоор Америкийн агаарын хүчин дахь RAND корпорацид хийж байсан судалгааны үедээ мессеж блок шилжүүлэлтийн ойл- голтоо хөгжүүлсэн. Эхлээд 1961 оны зун агаарын хүчинд B-265[1] товч зөвлөмжийг танилцуулсан. Дараа нь 1962 онд P-2626 баримт бичгийг нийтэлсэн. Baran-ий P-2626 бичиг баримт нь томоохон хэмжээний тархсан харилцааны сүлжээний үндсэн архи- тектурыг тодоройлсон юм. Энэхүү бичиг баримт нь гурван түлхүүр санаанууд дээр төвлөрч байсан: Эхнийх нь, хоёр цэгүүдийн хоорондох олон замууд бүхий төвлөрсөн бус сүлжээний ашиглалт; хоёр дахь нь, хэрэглэгчийн бүрэн мессежүүд нь түүний хэлс- нээр мессеж блокууд(дараа нь пакетууд гэж хэлдэг болсон)руу хуваагдах; гурав дахьнь эдгээр хадгалах болон илгээх свичинг хийгдсэн мессежүүдийг эцсийн цэгт хүргэх. Baran-ий ажил нь Англид байрлах Үндэсний Физик Лаборатор дахь Donald Davies- ээр бие биендээ хамааралгүйгээр гүйцэтгэгдсэн судалгаатай төсөөтэй байсан. 1965 онд Davies пакет-свич сүлжээний ойлголтыг хөгжүүлж Английн өргөн сүлжээний саналыг тавьж байсан. Дараа нь Батлан Хамгаалах Яамнаас түүнд Baran-ий ажлын тухай ярьж өгсөн. Davies багийн гишүүн 1967 онд Үйлдлийн Системийн зарчмууд сэдэвт ACM зөвлөгөөн дээр Lawrence Robertstэй уулзаж хоёр грүүпийг цугтаа ажиллахыг ятгасан. Сонирхолтой нь Davies түүний Baran-ийхтай төстэй анхны сүлжээний загварын за- рим параметрүүдийг сонгосон бөгөөд үүнд нь пакетын хэмжээ нь 1024 бит байх зэрэг орж байсан. 1966 онд Davies УФЛ-ийн хэрэгцээг ашиглахын тулд сүлжээг лабораторт байгуулах мөн пакет свичингийн боломжтойг нотлох санал гаргаж байсан. Тэрхүү УФЛ өгөгдлийн харилцаа холбооны сүлжээ нь 1970 онд үйлчилгээнд нэвтэрсэн. Roberts болон ARPANET-ийн баг Davies-ийн ажил нэртэй байсныг "пакет свичинг"болгож өөрчилсөн. Анхны компьютерийн сүлжээ болон пакет свичинг сүлжээ нь 1968 онд цөөн хэдэн дундын хадгалах төхөөрөмж болон цөөн хэдэн Teletype Model 33 ASR терминалуу- дыг дундаа ашиглахаар дөрвөн ширхэг Control data 6600 компьютерүүдийг Lawrence Livermore National

Laboratory-д холбосноор эхлэж байжээ. 1973 онд Vint Cerf болон Bob Kahn Дамжуулын Удирдлагын Протокол(TCP) про- токолыг бичсэн. Энэ нь гол цэгүүд дунд пакет-свичинг сүлжээг ашиглан холбогдох боломж олгодог протокол юм. Үүнээс хойш өнөө үед бидний хамгийн том хэрэглээ болж чадсан интернет хөгжжээ.

2.2.2 Пакет Свичинг сүлжээнд

Пакет свичинг нь компьютерийн сүлжээнүүд гэх мэт тоон харилцаа холбооны сүлжээнүүдэд дамжууллын хоцролтыг багасгахын тулд сувгийн боломжит багтаамжийг оновчтой болгоход ашиглагдаж байна. Мөн энэ нь харилцааны тогтвортой байдлыг нэмэгдүүлэхэд маш чухал юм.

Нэг гол цэгээс нөгөө нэг рүү өгөгдлийг дамжуулхын тулд сүлжээний холболтонд долоон давхаргууд байдаг. Эцсийн хэрэглэгчийн компьютерийн буферүүд ямар нэгэн урттай өгөгдлийг дамжуулна. Хэрвээ тэрхүү өгөгдөл нь хэтэрхий том байвал жижиг жижиг хэсгүүд(сегментчлэл)рүү хуваагдана. Сегмент бүр өгөгдлийг илгээх болон бусад холбогдолтой мэдээллүүдийг агуулсан толгой хэсэгтэй байна. Дараа нь сегмент нь дараагийн давхарга руу дамжуулагдана. Энэхүү үйл явц нь өгөгдөл эцсийн давхаргад хүрэх хүртэл давтагдах бөгөөд дараа нь хүлээн авагч гол цэг рүү илгээгдэнэ. Хүлээн авсан гол цэг нь өгөгдлийг толгой хэсгээс нь салгадаг.

Эдгээр давхаргууд нь харилцаа холбооны нарийн төвөгтэй байдлыг хялбаршуулсан. Хамгийн дээд давхарга(давхарга 7) нь хэрэглэгчийн төвшний давхарга бөгөөд давхарга доошлох тусам улам командын шинжтэй болдог. Давхаргууд нь эцсийн цэг рүү дамжуулахад хоёртын тоо руу шилжүүлэхээр бэлтгэж байгаа хамгийн боломжит шийдэл юм.

2.2.3 Холболтгүй мөн холболт-хандлагатай горимууд

Пакет свичинг нь холболтгүй пакет свичинг буюу бидний мэдэхээр датаграм свичинг мөн холболт-суурилсан буюу виртуал хэлхээний свичинг гэж ангилагддаг. Холболтгүй протоколуудын жишээ гэвэл Ethernet, Internet Protocol(IP) мөн User Datagram Protocol(UDP). Харин холболт-суурилсан протоколуудад X25, Frame Relay, Multiprotocol Label Switching(MPLS), Transmission Control Protocol(TCP) гэх мэт багтдаг байна.

Холболтгүй горимын үед пакет бүр гүйцэт хаяглалтын мэдээллийг агуулна. Тэрхүү пакетууд нь тус тусдаа чиглүүлэгдэнэ. Пакет бүр эцсийн цэгийн хаяг, эх үүсвэрийн хаяг

мөн портын дугааруудаар хаяглагдсан байна. Мөн пакетын дарааллын дугаараар хаяглагдсан байна. Пакет бүр өөр өөр чиглүүлэлтүүдээр явсан мөн бүгд холболт-суурилсан бол систем холболтыг тохируулж болно. Гэвч пакет нь програмын шаардсан мэдээллийг илүү бага агуулсан байна. Эцсийн цэг дээр анхны мессеж, өгөгдөл пакетын дарааллын дугаар дээр үндэслэн зөв дарааллаар дахин угсрагдана. Ийм маягаар виртуал холболт буюу virtual circuit нь дамжууллын төвшний протоколоор эцсийн хэрэглэгч рүү заагддаг.

Холболт-суурилсан дамжуулал нь ямар нэгэн өгөгдлийн пакет дамжуулагдахаас өмнө оролцсон гол цэг бүр дээр тохиргооны үе шат хийгдэж байхыг шаарддаг. Пакетууд нь холболт тодорхойлогчийг агуулдаг бөгөөд эцсийн цэгүүдийн хооронд тохиролцдог. Ингэснээр алдааг шалгаж дамжуулдаг.

2.2.4 Пакет свичингийн давуу талууд

Пакет свичинг нь хэлхээ свичингтэй харьцуулахад олон давуу талуудтай:

- Хэлхээ свичинг сүлжээг бодвол илүү тогтвортой мөн Пакет свичинг сүлжээ нь хоёртын өгөгдлийг дамжуулахад илүү тохиромжтой.
- Одоо үеийн хамгийн дэвшилтэт технологи. Энэ нь дууг шифрлээд пакет хэлбэрт оруулж маш бага эрсдэлтэйгээр дамжуулах боломжтой.
- Гэмтсэн пакет дахин илгээгдэх боломжтой. Учир нь зөвхөн тэр л хэсэг гэмтсэн болохоор нийт файлыг дахин илгээх хэрэггүй.
- Мультиплекс хийх боломжтой. Өөр өөр хэрэглэгчид эсвэл нэг хэрэглэгчийн өөр өөр процессууд хугацааны нэг агшинд зэрэг холбогдож чадна.
- Зориулсан хэлхээ ашиглаагүй терминалуудын хоорондох трафик нийлбэр нь хувийн шугамуудыг ашигласнаас эдийн засгийн хувьд илүү ашигтай.
- Телефоны дуудлагын хамгийн бага төлбөрт хугацааны нэгжээс өгөгдлийн холболтын сешнүүд нь бага тохиолдолд телефоны залгасан өгөгдлийн үеийнхээс илүү эдийн засгийн хувьд үр ашигтай.
- Эцсийн цэгийн мэдээлэл нь пакет бүр дээр агуулагддаг. Тиймээс олон тооны мессежүүдийг олон янз бүрийн эцсийн цэгүүд рүү хурдан илгээж чадна

- Гол цэгүүд дээр байрлах компьютерүүд динамик өгөгдлийн чиглүүлэлтүүдийг зөвшөөрдөг. Энэхүү сүлжээн дахь энэ төрөлхийн онцлог нь тодорхой хугацаанд сүлжээн дээгүүр дамжиж байгаа пакетууд нь хамгийн сайн боломжит чиглүүлэлтийг сонгож авах боломжтой болгодог.

- Сүлжээний пакетын төрөлхийн онцлог нь мөн зам(линк) эсвэл гол цэгийн алдааны үед сүлжээний уналтыг засах боломж олгодог.

- Хэлхээ свич сүлжээтэй адилаар зохион байгуулах боломжтой. Жишээ нь X.25 болон ATM сүлжээнд тухайн аргыг хэрэглэдэг бөгөөд үүнийг виртуал хэлхээ гэж нэрлэдэг. Эдгээр виртуал хэлхээнүүд нь хэлхээ свич шиг ижил замууд дээр хэрэгжүүлэгддэг. Гэвч нэг үндсэн ялгаа байдаг нь Виртуал хэлхээнүүд нь өөр виртуал хэлхээнүүдтэй ижил линкийг ашиглахыг зөвшөөрдөг. Энэ ойлголт нь олон гол цэгүүдийн дунд нэг линк ашиглан нэгэн зэрэг холболт хийх боломж олгодог(хоёр цэгүүдийн хооронд хэлхээ свичинг холболт хийхээс илүү дээр).

2.3 Интернет протокол

Интернет протокол гэдэг нь Интернет болон ижил төстэй компьютер сүлжээнүүд дээр ашиглагддаг харилцааны протоколуудын бүрдэлийн компьютерийн сүлжээний загварыг хэлнэ. Үүнийг ихэнхидээ TCP/IP гэдгээр нь мэддэг. Учир нь TCP болон IP протоколууд нь маш чухал протоколууд бөгөөд анхны сүлжээний протоко- лууд нь энэ стандарт дээр тодорхойлогдсон. Заримдаа DoD гэдгээр нь ч мэддэг бөгөөд учир нь Америкийн Нэгдсэн улсын хамгаалалтын хэлтэс(USDoD)-ийн агентлаг буюу DARPA-аар санхүүжсэн сүлжээний загварын хөгжил юм.

TCP/IP нь төгсгөлөөс төгсгөлийн хооронд өгөгдөл хэрхэн пакет болон угсрагдах, хаяглагдах, дамжих, чиглүүлэгдэх мөн эцсийн цэг дээр ирэх зэрэг холболтын тодорхойлолтуудаар хангадаг. Энэхүү ажиллагаа нь сүлжээний адил төстэй протоколууд бүхий дөрвөн хийсвэр давхаргын зохион байгуулалттай.

TCP/IP протокол нь OSI моделийн өмнө бий болсон. TCP/IP протоколын төвшнүүд OSI моделийхтэй яг ч тохирдоггүй. Жинхэнэ TCP/IP протокол нь 4н төвшнээс тогтдог гэж тодорхойлогдсон. Үүнд: толгой компьютерээс сүлжээнд холбогдох, интернет, тээвэрлэлтийн мөн хэрэглээний төвшин. TCP/IP протокол OSI модельтэй харьцуулахад

хувьд толгой компьютерээс сүлжээнд холбогдох төвшнийг нь физикийн болон өгөгдөл холболтын төвшнүүдтэй адилтгаж болно.

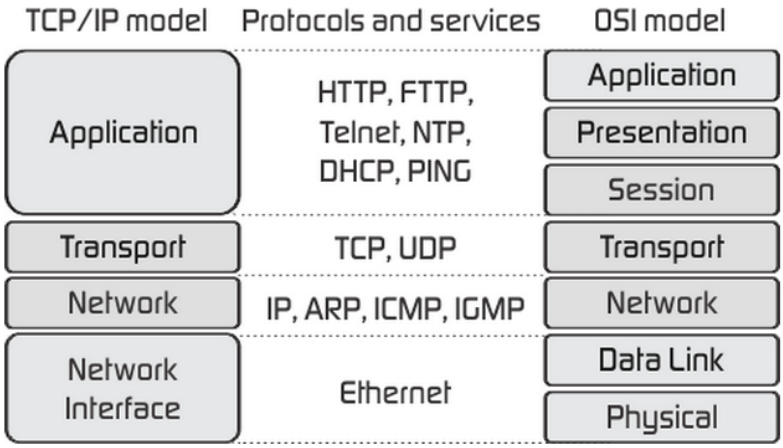
Интернет төвшин нь сүлжээний төвшинтэй ижил. Хэрэглээний төвшин нь орчин, танилцуулах, хэрэглээний төвшнүүдийн үүргийг гүйцэтгэнэ. Тиймээс бид энэ номонд TCP/IP протоколыг таван төвшнээс тогтсон гэж үзнэ: физикийн, өгөгдөл холболтын, сүлжээний, тээвэрлэлтийн, хэрэглээний. OSI моделийн төвшин бүрийн үүргийг зааж өгдөг бол TCP/IP протокол нь хэрэгцээнээсээ шалтгаалаад үүргүүд нь солигдож болно.

2.3.1 TCP/IP

'TCP/IP' -ийн үндсэн протокол гэж хэлэх бөгөөд 'TCP' дамжуулалт хянах протокол (transmission control protocol) гэж нэрлэнэ. Интернэт дээр ажиллах програм хангамжаас өгөгдлийг цуглуулан багц багцаар бэлтгэж зохих газрууд руу найдвартай дамжуулалтыг хангах үйл ажиллагааг хариуцан гүйцэтгэнэ. Энд дамжуулалт хянах протокол болон интернэт протокол (IP) хамтран ажилладаг.Дамжуулалт хянах протокол мэдээлэл солилцоонд гарах алдаа, мэдээлэл хүргэх дэг журам, удирдлагыг хянаж байх үүрэгтэй. TCP/IP эдгээр үүргийг гүйцэтгэнэ. Энэ 2 протокол бусад протоколтой харилцан ажилладаг гэдгийг мартаж болохгүй. Өөрөөр хэлбэл протоколууд тус тусдаа өөрийн гэсэн үүрэгтэй.

Үндсэн ялгаа: TCP/IP болон OSI-ийн хоорондох үндсэн ялгаа нь OSI загвар зөвлөмж загвар бөгөөд TCP/IP нь OSI загварын хэрэгжүүлэлт гэж үздэг.

Доорх зүйлсүүд нь OSI болон TCP/IP-ийн хоорондох түлхүүр тэмдэглүүштэй ялгаанууд юм.



Зураг 2.1: TCP/IP болон OSI

- OSI загвар нь "ерөнхий протокол - бие даасан стандарт". TCP/IP бол интернетэд зориулж хөгжүүлсэн стандарт гэж үзнэ.
- TCP/IP нь энгийн загвар учир нь долоон давхарга бүхий OSI-харьцуулахад илүү бага(дөрөв) давхаргатай.
- TCP/IP илүү найдвартай загвар учир нь интернет эргэн тойронд хөгжүүлэгдэж байдаг.
- TCP/IP нь OSI-ийн дата линк болон физик давхаргуудыг нэгтгэж сүлжээний хандалтын давхарга болгосон.
- OSI загвар зөвлөмж загвар бөгөөд TCP/IP нь OSI загварын хэрэгжүүлэл юм.

TCP/IP протокол нь шаталсан протокол юм. Шаталсан гэдгийн учир нь дээд төвшний пр отоколуудыг нь арай доод төвшнийхүүд нь дэмжинэ.

- Тээвэрлэлтийн төвшинд TCP/IP протокол нь 3 протоколыг тодорхойлно: тээвэрлэлтийг хянах протокол, хэрэглэгчийн өгөгдлийн протокол, урсгал хянах протокол. Сүлжээний төвшний үндсэн интернет ажлын протокол нь TCP/IP протоколоор тодорхойлогдоно

2.3.2 Порт хаяг

- Физик болон логик хаягууд нь дамжиж буй өгөгдлийн тоо хэмжээний хувьд чухал үүрэгтэй. Өнөөдөр компьютер нь олон үйлдлийг зэрэг хийх чадвартай. Жишээ нь А компьютер С компьютертэй TELNET-ээр холбогдсон. Мөн тэр үедээ В компьютертэй файл дамжуулах протоколоор холбогдсон гэж үзье. Энэ үед процессууд мэдээллээ зэрэг хүлээж авахын тулд хаяглах хэрэгтэй. Энэ процессд өгч буй хаягийг порт хаяг гэнэ. TCP/IP дахь порт хаяг нь 16 битийн урттай байна.

2.3.3 IP хаяг

IP хаяг нь компьютерийн сүлжээнд холбоотой, IP буюу Internet Protocol-ийг ашиглан холбогдож байгаа бүх төхөөрөмжинд байдаг тоон хаяг юм. IP буюу Интернэт протокол нь сүлжээнд байгаа компьютерүүд хэрхэн яаж хоорондоо холбогдохыг үзүүлдэг дүрэм юм. Өөрөөр хэлбэл өөр хоорондоо ярилцдаг хэл юм. Интернэт протоколын IPv4 ба IPv6 гэсэн хоёр хувилбар байдаг бөгөөд тус бүрийн хаяглалтын схем өөр.

IP хаягууд нь хоёртын тооллоор хадгалагддаг боловч дэлгэцэнд харуулахдаа хүнд илүү ойлгомжтой 192.168.0.1 (IPv4) эсвэл 2001:db8:0:1234:0:567:1:1 (IPv6) гэх мэт хэлбэрээр гаргадаг. IP хаягийн үүргийг дараах хэлбэрээр тодорхойлжээ: "Нэр нь бидний хайж байгаа зүйлийг тодорхойлдог. Хаяг (address) нь хаана байгааг нь харуулдаг. Зам (route) нь яаж тэнд очихыг харуулдаг юм."

TCP/IP-г анх зохион бүтээхдээ IP хаягийг 32-бит тоогоор тодорхойлсон бөгөөд Internet Protocol Version 4 (IPv4) буюу IPv4 хувилбар нэрээр одоо ч гэсэн хэрэглэгдсээр байгаа болно. Гэсэн хэдий ч интернетийн хурдацтай өсөлтийн улмаас IPv4 хаягийн нөөц хүрэлцээгүй болж байгаа учраас шинэ хаягийн систем - IPv6-г 128 бит хэрэглэн боловсруулжээ.

Интернет протокол нь сүлжээнүүдийн хооронд мэдээллийн багцуудыг чиглүүлэх үүрэгтэй бөгөөд IP хаягууд нь сүлжээний бүтэц, зам дээр мэдээллийн эх үүсвэр болон хүрэх төхөөрөмжийн байрлалыг тодорхойлж байдаг. Энэ зорилгоор IP хаягны зарим битүүд нь сабнэт буюу дэд сүлжээг тодорхойлоход хэрэглэгддэг. CIDR (Classless Inter-Domain Routing) бичлэгээр бол дэд сүлжээг тодорхойлогч битийг IP хаягны ард 192.168.100.1/16 байдлаар тэмдэглэдэг. IP хаяг нь дотоод (дотоод сүлжээнд хэрэглэх) болон гадаад (интернет, WAN сүлжээнд) байж болно.

IP хаягны эхэн үеийн стандартуудад IP хаягийг үүнийг компьютер эсвэл сүлжээний

төхөөрөмж тус бүрд онцгойгоор зааж өгсөн байхаар төлөвлөж байжээ. Гэсэн хэдий ч хувийн буюу дотоод сүлжээнүүд олноор гарч ирэхэд заавал тус тусдаа IP хаягтай байх нь хүрэлцээгүй болохоор байсан тул RFC 1918 стандарт гаргасан бөгөөд энэ стандартын дагуу хэн ч, хаана ч дотоод сүлжээндээ хэрэглэж болохоор дотоод IP хаягуудыг тодорхойлж өгсөн. Харин эдгээр дотоод сүлжээнүүд нь интернетэд хандахдаа NAT буюу Network Address Translation ашиглан нэгдсэн нэг гадаад хаягаар хандаж болдог байна.

2.3.4 OSI загвар

OSI модел буюу Нээлттэй системүүд харилцан холбогдох загвар нь Олон Улсын Стандартын Байгууллагын (ISO) зүгээс компьютерийн сүлжээний ажиллагааг давхаргуудад хуваан стандартжуулах гэсэн оролдлого юм. Давхарга болгон нь өөрийн дээд болон доод давхаргуудтай тодорхой интерфэйс ашиглан харилцах ба хоёр өөр элементийн ижил давхаргууд хоорондоо протоколуудын тусламжтайгаар харилцана.

Практикт хэрэглэгдэж буй системүүд нь OSI моделээс өөр байх тохиолдол их бий. Жишээ нь, Интернэтийн хэрэглэдэг TCP/IP модел нь арай өөр бүтэцтэй. Мөн OSI систем доторх дэд давхаргууд академи болон үйлдвэрлэлийн нийтлэлүүдэд өөрийн багтах давхаргаасаа илүүтэйгээр яригдах нь бий. Жишээ нь, MAC (Media access control) буюу орчны хандалт удирдлагын давхарга нь data-link layer буюу өгөгдөл холболтын давхаргын дэд давхарга боловч тусдаа давхарга болон яригдах нь элбэг.

80-аад оны эхээр ISO (International Standards Organization), ITU-T (International Telecommunications Union – Telecommunication sector) болон бусад стандартчлалын байгууллагууд сүлжээний хөгжилд түлхэц өгсөн загварыг боловсруулжээ. Энэ загварыг нээлттэй системүүдийн харилцан үйлчлэлийн загвар (Open System Interconnection, OSI) буюу OSI загвар гэж нэрлэнэ. OSI загвар нь системүүдийн харилцан үйлчлэлийн төрөл бүрийн төвшнийг тодорхойлж, стандарт нэр өгч төвшин бүрийн гүйцэтгэх үүргийг зааж өгсөн. Энэ загвар нь 70-аад оны үед компьютерийн сүлжээ байгуулж байх үед хуримтлагдсан туршлагын үндсэн дээр бүтээгджээ.

OSI загварт үйлдлийн систем, системийн программын болон аппарат хэрэгслүүдээр хийгдэх системийн харилцан үйлчлэлийн хэрэгслүүдийг тодорхойлдог. Уг загварт хэрэглэгчдийн программуудын харилцан үйлчлэлийн хэрэгслүүд ордоггүй. Программууд нь өөрийн харилцан үйлчлэлийн протоколоо системийн хэрэгслүүдэд хандан гүйцэтгэ-

дэг. Иймд хэрэглэгчийн төвшин ба программуудын харилцан үйлчлэлийг ялгаж ойлгох хэрэгтэй. Мөн программ нь OSI загварын дээд төвшнүүдийн үүргийг биелүүлж болно. Жишээ нь өгөгдлийн санг удирдах зарим системүүд файлд алсаас хандах өөрийн гэсэн хэрэгсэлтэй байдаг. Энэ тохиолдолд программ нь алслагдсан системд хандахдаа системийн файлын албыг ашиглалгүйгээр OSI моделийн дээд төвшнүүдийг алгасч, доод төвшинд байрлах сүлжээгээр мэдэгдэл дамжуулах төвшний системийн хэрэгслийг ашигладаг. Жишээ нь ямар программ хэрэглэгчийн төвшинд хүсэлт гаргаж, файлын албанд хандаж байг. Энэ хүсэлтийг үндэслэн хэрэглэгчийн төвшний программ хангамж стандарт хэлбэртэй мэдэгдлийг үүсгэнэ. Энгийн мэдэгдэл нь толгой хэсэг ба өгөгдлийн талбараас бүрдэнэ. Толгой хэсэгт хүлээн авагч компьютерт сүлжээгээр дамжуулж, юу хийх ёстойг нь тодорхойлсон албан мэдээллийг агуулна. Уг тохиолдолд толгой хэсэгт файлын байрлал, түүнд хийгдэх үйлдлийн тухай мэдээлэл агуулагдах ёстой. Өгөгдлийн талбар нь хоосон байх буюу файлд бичих өгөгдлийг агуулсан байж болно. Энэ өгөгдлийг хаягаар нь хүргэхийн тулд доод төвшнүүд олон асуудал шийдэх ёстой.

Мэдэгдлийг үүсгэсний дараа хэрэглэгчийн төвшин түүнийг стекийн дагуу дүрслэлтийн төвшинд хүргэнэ. Дүрслэлтийн төвшний протокол хэрэглэгчийн төвшний толгой хэсгээс авсан мэдээллийн дагуу зохих үйлдлүүдийг гүйцэтгээд уг мэдэгдэлд өөрийн албан мэдээлэл болох дүрслэлтийн төвшний толгой хэсгийг нэмэх ба тэнд хүлээн авах компьютерын дүрслэлтийн төвшинд зориулсан заалтууд агуулагдана. Энэ мэдэгдэл нь доош синхрончлох төвшинд шилжих ба уг төвшин нь өөрийн толгой хэсгээ нэмэх гэх мэтээр үргэлжлэнэ.(Зарим протоколууд нь мэдэгдлийн эхэнд бус төгсгөлд нь “сүүл” байдлаар албан мэдээллээ байрлуулдаг.) Эцэст нь мэдэгдэл физик төвшинд хүрэх ба тэндээс нь холболтын шугамаар хүлээн авах компьютер руу нь дамжуулна.

Мэдэгдэл (message) гэсэн ойлголтоос гадна сүлжээний мэргэжилтнүүдийн өгөгдөл солилцох өөр нэгжүүд байдаг. ISO стандартад төрөл бүрийн төвшний протоколуудын дамжуулж байгаа өгөгдлийн нэгжийг протоколын өгөгдлийн блок (Protovol Data Unit - PDU) гэж нэрлэдэг. Тодорхой төвшний өгөгдлийн блокуудыг тусгай нэрээр: кадр (frame), багц (packet), дейтаграмм (datagram), сегмент (segment) гэж нэрлэнэ.

OSI загварт протоколын хоёр үндсэн төрөл байдаг. Холболттой (connection-oriented) протоколд өгөгдөл дамжуулахын өмнө илгээгч ба хүлээн авагч нь холбоо тогтоож, өгөгдөл солилцох үедээ ашиглах зарим параметрүүдээ сонгодоно. Харилцаа дууссаны дараа энэ холболтоо тасалдаг. Холболттой харилцан үйлчлэлийн нэг жишээ нь телефон

утас юм.

Хоёр дахь төрлийн протоколууд нь урдчилан холболт хийдэггүй. Ийм протоколуудыг дейтаграмын протоколууд ч гэж нэрлэдэг. Илгээгч нь мэдэгдлийг бэлэн болохоор нь явуулна. Захианы хайрцаг руу захиаг хийх нь холболтгүй ажиллагааны жишээ юм. Компьютеруудын харилцан үйлчлэлд эдгээр протоколууд хоёулаа ашиглагддаг.

2.3.4.1 OSI загварын төвшнүүд

1. **Физик төвшин** (Physical layer) Агаар, кабель гэх мэт мэдээлэл дамжуулах физик орчинтой харилцана.Энэ төвшин нь Сэргийллийн төвшний фреймүүдийг бит битээр нь нэг зангилаанаас нөгөө зангилаа руу нийлүүлэх үүрэг гүйцэтгэнэ. Энэ төвшний протоколууд нь сэргийллийн төвшинтэй адил сүлжээний төрөл дээр нэмээд дамжуулах хэрэгслээс хамаарна.

2. **Өгөгдөл холболтын төвшин** (Data-link layer) Физик орчноос ирсэн мэдээлэлтэй ажиллах давхарга (Жишээ нь, Этэрнэт).Энэ төвшин датаграмуудыг хост тус бүрийн залгах төвшин рүү дамжуулна. Эхээс товлосон хоост уруу биш замд таарах бүх зангилаа (хоост эсвэл рүүтэр) бүр дээр энэ төвшин зам заах үүрэг гүйцэтгэнэ.

3. **Сүлжээний төвшин** (Network layer) Өгөгдлийн дамжих замыг заана (Жишээ нь, IP).Энэ төвшин нь илгээсэн талын зөөврийн төвшний сегментүүдийг товлосон хаягийн залгах төвшний элементүүдэд нийлүүлэх үйлчилгээг үзүүлнэ.

4. **Тээвэрлэлтийн төвшин** (Transport layer) Мэдээллийн бодит агуулгыг дамжуулна (Жишээ нь, TCP).Интернетийн сүлжээний хувьд энэ төвшинд TCP, UDP хоёр байна.Энэ төвшин нь илгээсэн талын зөөврийн төвшний сегментүүдийг товлосон хаягийн залгах төвшний элементүүдэд нийлүүлэх үйлчилгээг үзүүлнэ.

5. **Холболтын төвшин** (Session layer)

6. **Үзүүлэх төвшин** (Presentation layer)

7. **Хэрэглээний төвшин** (Application layer)Энэхүү төвшинд HTTP, HTTPS, FTP, SMTP, LDAP ... гэх мэт олон протоколууд хамаарна. Тэдгээр протоколууд нь програмуудад хэрэглэгддэг дээд төвшний протоколууд. Мэдээллийн бодит агуулгыг ашиглах давхарга (Жишээ нь, HTTP)

Давхарга	Нэр	Зориулалт
7	Хэрэглээний давхарга (Application Layer)	Хэрэглээний програм хангамжуудыг сүлжээнд холбодог. Энэ давхарга нь удирдлагын функцүүдийг агуулдаг.
6	Үзүүлэнгийн давхарга (Presentation Layer)	Энэ давхарга нь хэрэглээний төвшинг бусад давхаргуудтай холбодог. Өгөгдлийн формат болон синтаксууд нь хэрэглээний давхаргад зориулагдсан байна.
5	Сешн давхарга (Session Layer)	Төгсгөлийн цэгүүд дээр ажиллаж байгаа програм хангамжуудын хоорондох харилцаа холбооны удирдлагаар хангадаг.
4	Тээвэрлэлтийн давхарга (Transportation Layer)	Төгсгөлийн гол цэгүүдийн хоорондох холболтыг хангадаг.
3	Сүлжээний давхарга (Network Layer)	Сүлжээн дээрх өгөгдлийн харилцааг зохицуулна. Энд хаяглалтууд, чиглүүлэлтийн мэдээллүүд гэх мэт сүлжээний мэдээллүүд байна.
2	Сувгийн давхарга (Data Link Layer)	Энэ төвшинд физик хаяглалт хийгдэх бөгөөд Медиа Хандалтын Удирдлага (MAC) хаяг ашиглана.
1	Физик давхарга (Physical Layer)	Сүлжээнд гол цэгүүдийн хооронд физик холболтыг хийдэг давхарга. Өгөгдөл 2 –тын битийн урсгал хэлбэртэйгээр дүрслэгдсэн байдлаар түүн дээгүүр дамжуулагддаг.

Хүснэгт 2.1: Нээлттэй системүүд харилцан холбогдох загварын(OSI) 7 давхарга

2.3.5 TCP

TCP/IP-ийн хамгийн чухал протоколуудын нэг бөгөөд процесс хоорондын дамжууллыг гүйцэтгэдэг. Холболтод түшиглэсэн, өгөгдлийн дарааллыг хадгалдаг, алдааг шалгадаг, найдвартай байдлыг хангаж өгсөн протокол юм. OSI загварын тээвэрлэлтийн төвшинд байрладаг.

TCP-г ашигладаг дээд төвшний протоколуудын жишээ гэвэл HTTP, HTTPS, SMTP, POP3, IMAP, SSH, FTP, Telnet зэрэг юм. TCP-ийн гол онцлог бол найвартай үйлчилгээ.

2.3.6 UDP

UDP протоколын connectionless буюу холболт үүсгэдэггүй үйлчилгээний гол санаа нь UDP-гээр илгээгдсэн пакет бүр нь бусад пакетуудаасаа үл хамаарна. Ингэхдээ илгээж буй, хүлээн авч буй талууд нь ижилхэн байсан ч хоорондоо хамааралгүй байдаг. Хэрэглэгчийн datagram-уудыг дугаарладаггүй.

Түүнчлэн холболт үүсгэх, салгах процесс байдаггүй бөгөөд пакет бүр ялгаатай замуудаар дамжиж болно. Холболт үүсгэхгүй байхын тэг сул тал нь холболтын үед дамжуулж байгаа мэдээллийг урсгал байдлаар дамжуулж болохгүй бөгөөд пакет бүрд таарах байдлаар жижиг хэсгүүдэд задлан дамжуулах ёстой.

2.3.7 HTTP

HTTP /Hyper Text Transfer Protocol/ - Вэб хуудаснуудад ашиглагддаг

2.3.8 FTP

FTP /File Transfer Protocol/ - Файл оруулах, татахад ашиглагдана.

2.3.9 ICMP

ICMP /Internet Control Message Protocol/ - өөр чиглүүлэгчтэй мэдээлэл солилцох чиглүүлэгчид ашиглагддаг.

2.3.10 SMTP /Simple Mail Transport Protocol/ - Имэйл захиа буюу текстэн мэдээллийг дамжуулахад хэрэглэгддэг.

2.3.11 SNMP

SNMP /Simple Network Management Protocol/ - Алслагдсан компьютераас мэдээлэл цуглуулахад ашиглагдана.

2.3.12 Telnet

Telnet /Teletype Network/ – Одоо ашиглаж байгаа энэ компьютераасаа алс хол байгаа компьютерт нэвтрэн үйлдэл хийхэд ашиглагддаг. Гэхдээ цаад компьютераасаа зөвшөөрөл авсан байх хэрэгтэй.

2.4 Зурвасын өргөн

Компьютерийн ухаанд зурвасын өргөн нь секундэд дамжиж байгаа битүүдийн хэмжээ юм. Өөрөөр хэлбэл тухайн холболтоор өгөгдлийг нэг секундэд өгөгдлийг татах болон илгээх хэмжээ. Хэмжих нэгж нь bps(bet per second). Өөрөөр зурвасын өргөнийг сүлжээний зурвасын өргөн, өгөгдлийн зурвасын өргөн эсвэл дижитал зурвасын өргөн гэх мэтээр тодорхойлж болно.

2.4.1 Сүлжээний зурвасын багтаамж

Компьютерийн сүлжээнд зурвасын өргөн нь өгөгдсөн хугацааны үед(ихэнхидээ секунд) нэг цэгээс өөр цэг рүү зөөгдөж чадах өгөгдлийн нийт дүн хэмжээ буюу өгөгдлийн дамжуулалтын хурдыг хэлнэ. Орчин үеийн сүлжээнүүд нь секунд бүрд сая сая битүүд(мегабит буюу Mbps) эсвэл олон тэрбум битүүд(гигабитс буюу Gbps)-ийг дамжуулдаг байна.

Зурвасын өргөн нь дан ганц сүлжээний гүйцэтгэлийн нөлөөллийн хүчин зүйл биш юм. Энэ нь мөн сүлжээний нэвтрүүлэх чадамжийн муутгадаг пакетын алдагдал хоцролт мөн доргио зэргийг тодорхойлоход ашиглаггдаг. Сүлжээний зам нь ихэвчлэн тус бүрдээ өөрийн зурвасын өргөн бүхий дараалсан линкүүдийн багц байдаг. Тиймээс тухайн төгсгөлөөс-төгсгөл зурвасын өргөн нь хамгийн бага хурдтай линкийн зурвасын өргөнөөр хязгаарлагддаг.

Зурвасын өргөн гэсэн хэллэг нь заримдаа интернетийн бит хурд, сувгийн багтаамж эсвэл тоон харилцаа холбооны системд физик мөн логик холболтын замын хамгийн их нэвтрүүлэх чадамж зэргийг тодорхойлж болно. Жишээ нь зурвасын өргөний тестүүд нь компьютерийн сүлжээний хамгийн их нэвтрүүлэх чадамжийг хэмждэг.

2.4.2 Сүлжээний зурвасын өргөний хэрэглээ

Ялгаатай програм бүр ялгаатай зурвасын өргөнийг шаарддаг. Эгшин зуурын мессеж харилцаа нь секундэд 1000 бит(bps)-ээс бага хурд шаардлагатай; IP дээр яриа(VoIP) хийх харилцаа нь дуу хоолойг тогтуун мөн цэвэрхэн дамжуулхын тулд секундэд 56 мянган бит(Kbps) хурдыг шаарддаг. Стандарт видео (480p) нь секундэд 1 мегабит(Mbps) хурд дээр ажилладаг. Гэвч HD видео(720p) нь 4 Mbps хурдтай ажилладаг харин HDX(1080p) нь 7 Mbps ээс өндөр хурдтай байх хэрэгтэй.

Бодит зурвасын өргөн нь сүлжээний замаар найдвартай дамжиж чадах хамгийн их хурд юм. Энэхүү хурд нь тухайн файлыг өөрийнхөө байгаа цэгийг орхиж амжилттай эцсийн цэг рүү дамжигдахад шаардагдах хугацааг хэмжээг тодорхойлно.

Бит/секунд хэмжүүр бүхий зурвасын өргөн нь мөн холболтын замаар амжилттай дамжсан өгөгдлийн дундаж хурдыг тодорхойлдог. Үүнийг хэрэгжүүлэхдээ зурвасын өргөнийг хэлбэржүүлэх, зурвасын өргөний удирдлага, зурвасын өргөн throttling, зурвасын өргөн малгай (cap), зурвасын өргөний хуваарилалт (жишээ нь зурвасын өргөний хуваарилалтын протокол болон динамик зурвасын өргөний хуваарилалт) гэх мэт технологиуд болон ойлголтуудыг ашигладаг байна. Бит урсгалуудын зурвасын өргөн нь ашиглагдаж буй сигналын дундаж утгатай тэнцүү.

Сувгийн зурвасын өргөн нь өгөгдлийг нэвтрүүлэх чадамжтай андуурагдах тохиолдлууд байдаг. Жишээ нь x bps бүхий суваг нь протоколууд, шифрлэлтийн үед өгөгдлийг х хурдтай дамжуулах шаардлагүй бол үлдсэн хурдыг ашиглан өөр хэрэглээнд зарцуулж болно. Интернетийн ихэнхи урсгал нь холболт бүр дээр гурван-замт гар барилт(three-way handshake) шаарддаг Дамжууллын удирдлагын протокол (TCP) ашигладаг. Хэдийгээр орчин үеийн ихэнхи протоколын хэрэгжүүлэлтүүд нь үр бүтээмжтэй боловч энгийн протоколуудыг бодвол илүү их толгой хэсгийг агуулдаг. Мөн өгөгдлийн пакетууд нь алдагдсан тохиолдолд ашигтай өгөгдлийн нэвтрүүлэх чадамжийг багасгаж байдаг. Ихэнхи тохиолдолд үр бүтээлтэй тоон харилцаа холбоо нь хүрээ(framing) протоколууд хэрэгтэй байдаг бөгөөд мөн хэрэгжүүлэлтээс шалтгаалж толгой хэсэг мөн

бодит нэвтрүүлэх чадамж хэрэгтэй байдаг. Ашигтай нэвтрүүлэх чадамж нь бодит сувгийн зурвасын өргөнтэй тэнцүү эсвэл бага байна.

2.5 Socket

2.5.1 Төрөл

Хэрэглэгчдэд боломжтой сокетын дөрвөн төрөл бий. Эхний хоёр нь ихэвчлэн хэрэглэгддэг бөгөөд сүүлийн хоёр нь ховорхон хэрэглэгддэг.

Процесууд нь ижил төрлийн сокетуудын хооронд харилцдаг гэж таамагладаг. Гэвч өөр өөр төрлийн сокетуудын хооронд харилцахаас сэргийлсэн хязгаарлалтууд байдаггүй.

- **Стрийм сокетууд** - Сүлжээ тогтсон орчин нь баталгаатай болсон тохиолдолд илгээдэг. Хэрвээ "A, b, C"гэсэн гурван тэмдэгтийг стрийм сокетоор дамжуулж илгээвэл эдгээр тэмдэгтүүд нь яг ижил байрлалтайгаар "A, b, C"хүрч ирдэг. Эдгээр сокетууд нь өгөгдлийг дамжуулахдаа TCP протоколыг ашигладаг. Хэрвээ хүргэлт нь боломжгүй болвол илгээгч нь алдаа шалгуур үзүүлэлтийг хүлээн авдаг. Өгөгдлийн бүртгэл нь ямар ч хил хязгаар байхгүй.

- **Датаграм сокетууд** - Сүлжээ тогтсон орчин нь баталгаатай биш байсан ч илгээдэг. Энэ сокет нь холболтгүй учир нь Стрийм сокет шиг нээлттэй холболт байх алба байхгүй. Шууд л пакетаа эцсийн цэгийн мэдээлэлтэй цуг бүтээгээд илгээхэд хангалттай. Хэрэглэгчийн Датаграм Протокол(UDP) ашигладаг.

- **Raw Сокетууд(эсвэл Raw IP сокетууд)**-Эдгээр сокетууд нь хэрэглэгчдийг үндсэн харилцааны протоколууд руу хандах боломж олгодог. Эдгээр нь Датаграм дээр суурилдаг гэхдээ нарийн шинж чанар нь протоколоор хангагдсан интерфэйсээс хамааралтай. Raw сокетууд нь ерөнхий хэрэглэгчид зориулагдаагүй. Шинэ харилцааны протоколууд хөгжүүлэх эсвэл одоо байгаа протоколуудын нуугдмал хэсгийг илүү үр ашигтай болгох зорилготой байгаа хүмүүст зориулж гаргасан гэж үзэж болно.

- **Дараалсан Пакет Сокетууд** - Эдгээр сокетууд нь стрийм сокетуудтай төстэй. Энэхүү интерфэйс нь Сүлжээний Систем(NS)-үүдийн хэсэг шиг хангагддаг ба NS програмууд маш чухал сокет юм. Дараалсан-пакет сокетууд нь Дарааллан Пакет

Протокол (SPP) эсвэл Интернэт Датаграм Протокол (IDP) толгойнуудыг пакет эсвэл пакетуудын групп дээр удирдах боломж олгодог.

Бүлэг 3

Судалгааны хэсэг

3.1 Протокол дата юнит(PDU)

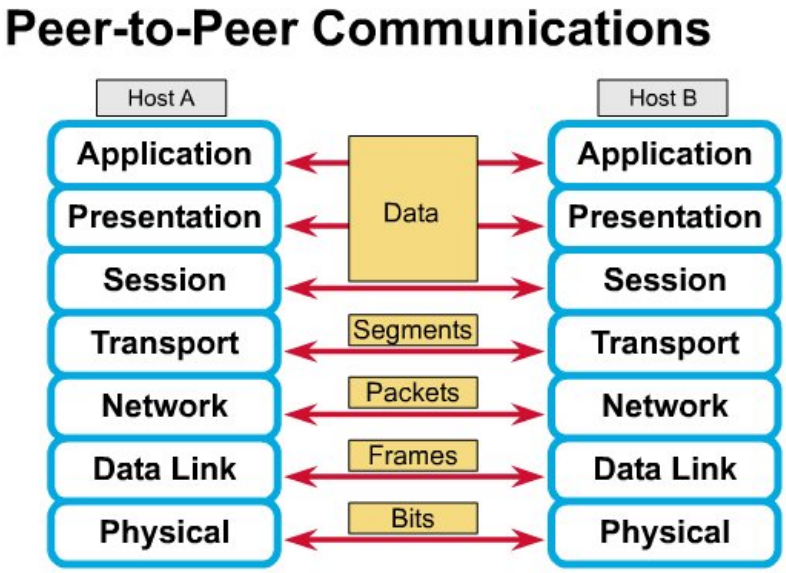
Харилцаа холбооны салбарт Протокол Дата Юнит(PDU) гэдэг нэр нь дор дурдсан утгуудтай:

- Сүлжээний төхөөрөмжүүдийн дунд хүргүүлэгдэх Мэдээлэл бөгөөд хаягийн мэдээллүүд гэх мэт удирдлагын мэдээллүүд эсвэл хэрэглэгчийн өгөгдөл зэргүүдийг агуулж болно.
- Давхаргажсан системд өгөгдсөн давхаргын заасан протоколын өгөгдлийн нэгж.

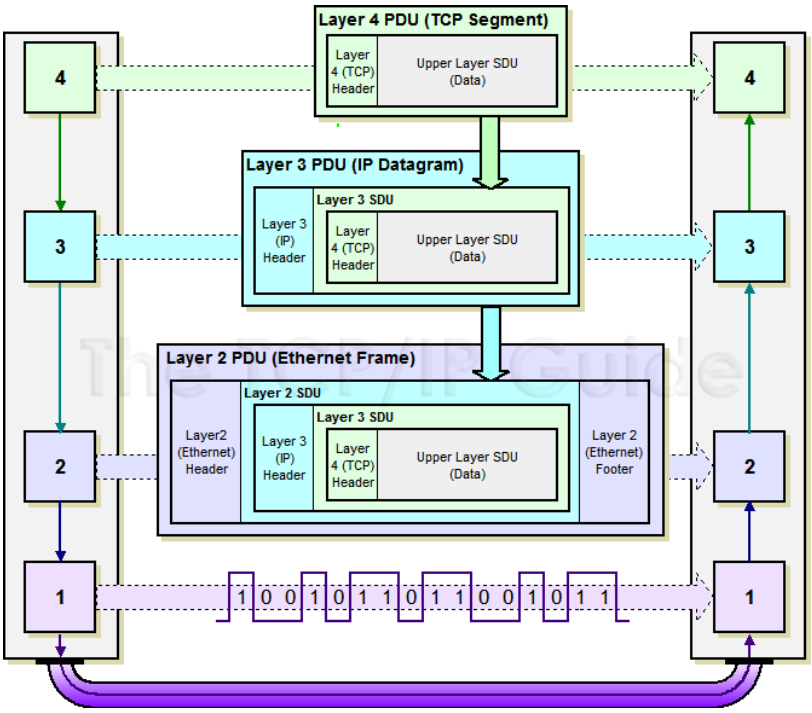
OSI загварт

PDU нь OSI-ийн эхний 4 давхаргууд бүр дээр өөр өөр байна:

- Давхарга 1 (Физик давхарга) дээр PDU нь бит байна.
- Давхарга 2 (Дата линк давхарга) дээр PDU нь фрейм байна.
- Давхарга 3 (Сүлжээний давхарга) дээр PDU нь пакет байна.
- Давхарга 4 (Тээвэрлэлтийн давхарга) дээр PDU нь TCP нь сегмент, UDP нь датаграм байна.
- Давхарга 5-6-7 (Хэрэглээний давхарга) дээр PDU нь ил текст, шифрлэсэн эсвэл шахсан хэлбэртэйгээр байх дата буюу өгөгдөл байна.



Зураг 3.1: PDU болон Давхаргын хаяглалт



Зураг 3.2: Давхарга бүрийн хаяглалт

3.1.1 Сервис Дата Юнит(SDU)

Нээлттэй системүүдийн харилцан холбоо(OSI)-ийн нэр томъёонд үйлчилгээний өгөгдлийн нэгж(SDU) нь OSI давхаргаас доод давхарга руугаа дамжуулагдсан бөгөөд хараахан доод давхаргаар PDU руу хайрцаглагдаагүй өгөгдлийн нэгж юм.

Энэ нь PDU-ээс ялгаатай. Тухайн SDU байгаа давхаргыг 'n' гэвэл PDU нь 'n+1' давхарга юм. Үүнээс дүгнэвэл SDU нь тухайн PDU бүрийн хувьд 'ачаа' байна. Энэ нь өөрөөр тухайн давхарга бүрийн доод давхаргаар хэрэгжүүлэгдэх хайрцаглалт(encapsulation)-ын үйл явц буюу SDU нь PDU болон өөрчлөгдөх үйл явц юм. SDU дотор агуулагдаж байгаа бүх өгөгдөл PDU дотор орж хайрцаглалт хийгддэг. Давхарга n-1 дээр SDU руу толгой эсвэл хөл эсвэл хоёуланг нь нэмдэг ингэснээр PDU болон өөрчлөгдлөө гэж үзнэ. Нэмэгдсэн толгой эсвэл хөл нь соорс хэсгээс эцсийн цэг рүү чиглүүлэгдэхэд хэрэгтэй мэдээллүүд болон бусад мэдээллүүд байна.

3.2 Raw Сокет

Эхлээд raw сокетыг энгийн сокетоос юугаараа ялгаатай вэ гэдгийг тодорхойлох хэрэгтэй. Энгийн сокет нь танд бэлдээд өгчихсөн аяга дүүрэн амтлаг шарсан талх, төмс мах мөн хэрчээд бэлдсэн ногоонууд юм. Үүнийг та зөвхөн хазаад ходоод руугаа оруулахад л хангалттай ба танд энэхүү бэлэн хоолны илчлэг буюу сервис л чухал юм. Харин тэрхүү сервис үйлчилгээг хэрхэн ямар үйлдлүүд шаардаж таны ходоод руу орох вэ гэдгийг өөрөө турших боломжоор raw буюу түүхий сокет танд олгоно. Энэ нь нэг талаас анхлан суралцагчид нөгөө талаас хөгжүүлэгчидэд зориулж гаргасан.

Компьютерийн сүлжээнд raw сокет нь тодорхой заасан тээвэрлэлтийн түвшний форматгүйгээр интернет протокол пакетуудыг шууд илгээх болон хүлээн авах боломж олгодог интернет сокет юм. Стандарт сокет дээр пакетын ачаа нь дамжуулагдахын тулд сонгосон тээвэрлэлтийн давхаргын протокол(TCP UDP гэх мэт) хайрцаглагддаг. Эсрэгээрээ raw сокет нь ихэвчлэн пакетын толгой хэсгийг агуулсан raw пакетуудыг хүлээж авдаг. Пакетуудыг дамжуулах үед автоматаар нэмэгддэг сокетын тохируулгын хэсэг нь тохируулах боломжтой байдаг.

Raw сокет нь nmap шиг аюулгүйн програмуудад ашиглагддаг. Учир нь аюулгүйн тусгай програмууд нь сүлжээний урсгалыг тодорхой хэрэглээний дагуу үүсгэх шаардлагатай байдаг тул энгийн сокет ашиглах нь үр дүнгүй байдаг. Харин raw сокетыг ашиг-

ласнаар сүлжээний урсгалыг хүссэнээрээ үүсгэх цаашлаад удирдах боломжтой болно.

Raw сокет нь үндсэн тээвэрлэлтийн хангагч руу хандах боломж олгодог сокетын нэг төрөл юм. Raw сокетыг ашиглахын тулд програм нь ашиглагдах үндсэн протоколын дэлгэрэнгүй мэдээлэлтэй байх ёстой.

3.2.1 Raw сокет үүсгэх

SOCK.RAW төрлийн сокетыг үүсгэхийн тулд af(address family) параметр нь AF.INET эсвэл AF.INET6, type параметр нь SOCK.RAW мөн протокол параметр нь протоколын дугаарыг заасан байх ёстой. Протокол параметр нь IP толгой хэсэгт байдаг(жишээ нь SCTP 132)

Raw сокет нь суурь тээвэрлэлт буюу тээвэрлэлтийн түвшний протоколуудын доод түвшний удидлагын боломжийг олгодог бөгөөд энэ төрлийн сокет нь аюулгүй байдалд заналхийлсэн буруу зорилгоор ашиглагдах боломжтой. Тиймээс Windows 2000 болон үүнээс хойшхи үйлдлийн системүүд дээр зөвхөн Адмминистратор бүлгийн гишүүд л SOCK.RAW төрлийн сокетыг ашиглах боломжтой.

3.2.2 Илгээх болон хүлээн авах үйл ажиллагаа

Програм нь SOCKET.RAW төрлийн сокетыг үүсгэсэн бол тухайн сокет нь өгөгдлийг хүлээн авах болон илгээх боломжтой болно. SOCKET.RAW төрлийн сокет дээр хүлээн авсан болон илгээгдсэн бүхий л пакетууд холболтгүй сокет буюу датаграм шиг харьцана.

SOCK.RAW сокетууд дээрх үйл ажиллагаанууд нь доорх дүрмүүдийн дагуу хийгдэнэ:

- SOCK.RAW төрлийн сокет дээр өгөгдлийг илгээхэд sendto эсвэл WSASendto функц ашиглагдана. Эцсийн цэгийн хаяг нь дурын сокетын хаягын бүлэг(address family)-ийн броадгаст эсвэл мултигаст хаяг байна. Броадгаст хаяг руу илгээхийн тулд програм нь SO.BROADCAST-ийг зөвшөөрсөн байх хэрэгтэй бөгөөд setsockopt-ийг ашиглана. Үгүй бол sendto, WSASendTo нь WSAEACCESS алдааны кодыг өгч амжилтгүй болно.

- IPv4 өгөгдлийг илгээхэд програм нь гарч байгаа датаграм пакетын IPv4 толгойг зааж өгөх эсэх талаар сонголт байна. Хэрвээ IP.HDRINCL сокет тохируулга нь

зөв IPv4 сокет(AF.INET-ийн address family)-ыг заасан бол програм нь илгээхдээ гарч байгаа өгөгдөлд IPv4 толгойг зааж өгөх ёстой. Хэрвээ энэ тохируулга нь заагдаагүй(default-аараа заагдаагүй байдаг) бол илгээх удирдлагаар гарч байгаа өгөгдөлд IPv4 толгой заагдаагүй байна.

- IPv6 өгөгдлийг илгээхэд програм нь гарч байгаа датаграм пакетын IPv4 толгойг зааж өгөх эсэх талаар сонголт байна. Хэрвээ IPV6.HDRINCL сокет тохируулга нь зөв IPv6 сокет(AF.INET6-ийн address family)-ыг заасан бол програм нь илгээхдээ гарч байгаа өгөгдөлд IPv4 толгойг зааж өгөх ёстой. Хэрвээ энэ тохируулга нь заагдаагүй(default-аараа заагдаагүй байдаг) бол илгээх удирдлагаар гарч байгаа өгөгдөлд IPv6 толгой заагдаагүй байна.

- SOCK.RAW төрлийн сокет дээр өгөгдлийг хүлэн авахад recvfrom эсвэл WSARecvFrom функц ашиглана. Эдгээр функцүүд нь хоёулаа пакетыг илгээсэн соорс IP хаягийг буцаах тохируулгатай байдаг. Тухайн хүлээн авсан өгөгдөл нь холболтгүй сокетоос ирсэн датаграм байна.

- IPv4(address family AF.INET) дээр програм нь IP.HDRINCL сокет тохируулгаас үл хамааран хүлээн авсан датаграм бүр дээр IP толгойг мөн хүлээн авна.

- IPv61(address family AF.INET6) дээр програм нь IPC6.HDRINCL сокет тохируулгаас үл хамааран хүлээн авсан датаграм бүр дээр сүүлийн IPv6 толгойг хүлээн авсны дараа бүх өгөгдлийг хүлээн авна. Програм нь Raw сокетыг ашиглаж байхдаа бүх IPv6 толгойг хүлээн авахгүй.

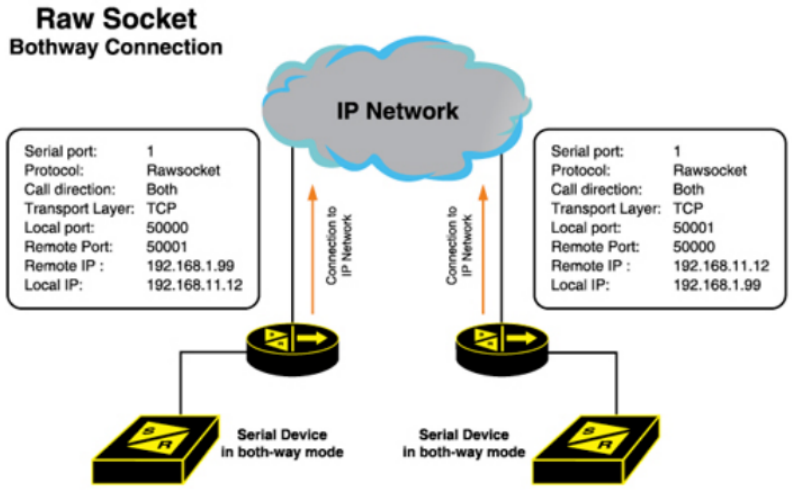
- Хүлээн авсан датаграмууд нь бүгд SOCK.RAW сокет руу хуулагдана. Ингэхдээ доорх нөхцөлүүдийг хангасан байх ёстой.

  - Сокет нь үүсэхдээ хүлээн авсан датаграмын IP толгой дахь протоколын дугаартай түүний протокол параметр дахь протоколын дугаар нь таарч байх ёстой.

  - Хэрвээ тухайн сокет дээр дотоод IP хаяг заагдсан бол энэ хаяг нь хүлээн авсан датаграмын IP толгой дахь эцсийн цэгийн хаягтай таарч байх ёстой. Програм нь bind функцийг ашиглан дотоод IP хаягийг зааж өгч болно. Хэрвээ тухайн сокетод дотоод IP хаяг заагдаагүй байвал датаграмууд нь хүлээн

авсан датаграмын IP толгой дахь эцсийн цэгийн хаягаас үл хамааран сокет руу бүгд хуулагдах болно.

– Хэрвээ тухайн сокет дээр гадаад хаяг заагдсан байвал энэ хаяг нь хүлээн авсан датаграмын IP толгойд заагдсан соорс хаягтай таарч байх ёстой. Програм нь connect эсвэл WSAConnect функцүүдийг ашиглан гадаад IP хаягийг зааж өгч болно. Хэрвээ тухайн сокетод гадаад IP хаяг заагдаагүй байвал датаграмууд нь хүлээн авсан датаграмын IP толгой дахь соорс IP хаягаас үл хамааран сокет руу бүгд хуулагдах болно.



Зураг 3.3: Raw сокет

Зарим SOCK.RAW төрлийн сокетууд нь гэнэтийн маш олон датаграмуудыг хүлээн авж болно. Жишээ нь PING програм нь ICMP echo хүсэлтүүдийг илгээх болон хариултуудыг хүлээн авахдаа SOCK.RAW төрлийн сокетыг үүсгэнэ. Програм нь ICMP echo хариултуудыг хүлээж байх зуур ICMP-ийн бусад бүх мессежүүд (ICMP HOST.UNREACHABLE гэх мэт) програм руу хүргэгдэж болно. Үүнээс гадна хэрвээ хэд хэдэн RAW.SOCKET сокетууд ижил хугацаанд компьютер дээр нээлттэй байвал ижил датаграмууд бүх нээлттэй сокетууд руу хүргэгдэнэ. Үүнийг шийдэхийн тулд програм нь датаграмуудыг зөвшөөрөх болон татгалзах гэсэн механизмтай байх хэрэгтэй. PING програмд энэхүү механизм нь хүлээн авсан ICMP толгойн цорын ганц тодорхойлогч дахь IP толгойг шалгах(жишээ нь програмын процесын дугаар) гэх мэт хэрэгжиж

болно.

3.2.3 Raw сокетын нийтлэг хэрэглээ

SOCK.RAW төрлийн сокетыг ашиглахын тулд администратор давуу эрх шаардлагатай. Хэрэглэгчид raw сокетыг ашигласан Winsock програмыг ажиллуулхын тулд компьютерийн администраторуудын групийн гишүүн байх ёстой бөгөөд хэрвээ ийм эрхтэй биш бол raw сокет нь WSAEACCESS алдааны кодыг өгч амжилтгүй болно. Виндовсын Виста хувилбараас хойш raw сокетын хандалт нь сокет үүсгэх үед хийгддэг болсон. Харин Виндовсын өмнөх хувилбаруудад нь raw сокетын хандалт нь өөр сокетын үйл ажиллагааны үеэр хийгддэг байсан.

Raw сокетын нэг үндсэн хэрэглээ нь IP пакетууд болон толгой хэсгийг дэлгэрэнгүй шалгах хэрэгтэй үед мөн програмын алдааг олж илрүүлэх зорилгоор ашиглах явдал юм. Жишээ нь raw сокет нь сүлжээний интерфэйсээр дайран өнгөрөх бүх IPv4 болон IPv6 пакетуудыг хүлээн авах сокетийг идэвхжүүлэхийн тулд SIO.RCVALL IOCTL ашиглаж болно.

3.2.4 Raw сокетын хязгаарлалтууд

Виндовс 7, Виндовс Виста, Виндовс XP SP2, Виндовс XP SP3 үйлдлийн системүүд дээр raw сокет ашиглан урсгалыг илгээх боломж нь хэд хэдэн аргаар хязгаарлагдсан байна:

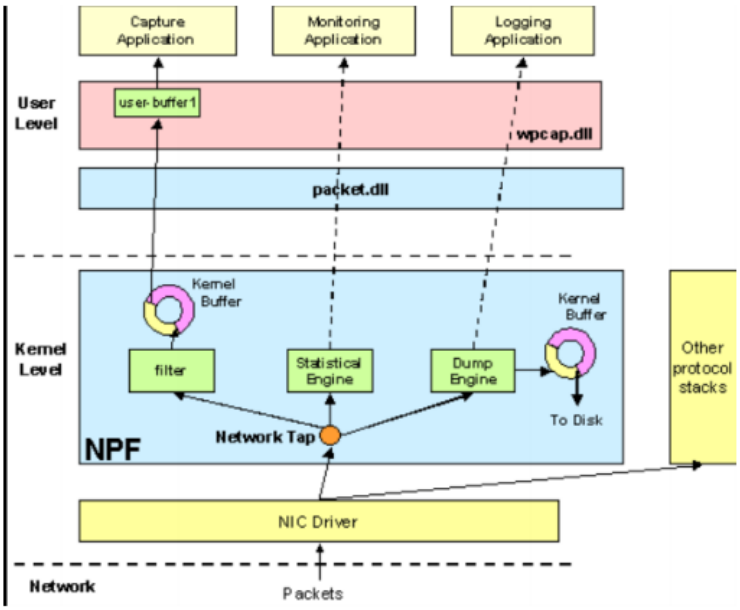
- TCP өгөгдөл нь raw сокетоор илгээгдэх боломжгүй.
- Хүчин төгөлдөр бус соорс хаягтай UDP датаграмууд raw сокетыг ашиглан илгээгдэх боломжгүй. Гарж буй датаграмын соорс IP хаяг нь сүлжээний интерфэйс дээр байх ёстой. Үгүй бол датаграм нь устгагдах болно. Энэ өөрчлөлт нь тархсан үйлчилгээ зогсоох халдлага хийхийн тулд хортон код явуулах боломжийг хязгаарладаг. Мөн хуурамч пакет(хуурамч соорс IP хаяг бүхий TCP/IP пакетууд)-уудыг илгээх боломжийг хязгаарлана.
- Raw сокет дахь bind функц нь IPPROTO.TCP протоколд зөвшөөрөгдөөгүй. Харин энэ функц нь бусад протокол(жишээ нь IPPROTO.IP, IPPROTO.UDP эсвэл IPPROTO.SCTP гэх мэт)-уудад зөвшөөрөгддөг.

3.3 Pcap

Компьютерийн сүлжээний удирдлагын салбарт pcap(пакет барих) нь хэрэглээний програмын интерфэйс(API)-үүдээс бүрддэг. Юникс-төст системүүд libpcap сангаар pcap-ийг хэрэгжүүлдэг. Харин Виндовс WinPcap-ийг ашигладаг.

Сүлжээний мониторинг програм хангамжууд сүлжээн дээр дамжиж байгаа пакетуудыг барьж авахын тулд libpcap эсвэл WinPcap ашигладаг. Холболтын давхаргын сүлжээнд пакетуудыг өөрчилж дамжуулах боломжтой.

Pcap API нь C дээр бичигдсэн, тиймээс өөр програмчлалын хэлнүүд болох java, .NET хэлнүүд мөн скриптинг хэлнүүд нь ихэнхидээ wrapper-ийг ашигладаг. C++ програмууд нь C API руу шууд холбогдох эсвэл обьект хандалтат wrapper ашигладаг.



Зураг 3.4: Pcap функц

libpcap болон WinPcap нь маш олон протокол анализ хийгч (пакет чагнагч), сүлжээний монитор, сүлжээний халдлага илрүүлэлтийн системүүд(NIDS), трафик үүсгэгч мөн сүлжээний тест гэх мэт нээлттэй эхийн мөн төлбөртэй сүлжээний хэрэгслүүдийг пакет-барих мөн шүүлтүүрийн хөдөлгүүрүүдээр хангадаг.

libpcap болон WinPcap нь мөн барьж авсан пакетуудыг файл руу хадгалах мөн тухайн хадгалсан пакетууд бүхий файлаа унших, анализ хийх боломжуудыг олгодог. Тэр-

хүү libpcap болон WinPcap форматын дагуу хадгалсан файл нь tcpdump, Wireshark, CA NetMaster мөн Microsoft Network Monitor 3.x зэрэг програмуудаар уншигдах боломж бүхий форматтай байна. Politecnico di Torino-д програмчид WinPcap-ийн анхны кодыг бичсэн.

3.3.1 libcap

libpcap нь анх "Lawrence Berkeley Laboratory"дахь сүлжээний судлаачдын грүүпийн tcpdump хөгжүүлэгчдээр хөгжүүлэгдэж байсан. Доод-түвшинд пакет барих, барьсан пакетын файлыг унших нь энэхүү сангуудаар хэрэгждэг. Одоо ч гэсэн libpcap нь tcpdump.org буюу tcpdump хөгжүүлэгчдээр хөгжүүлэгдэж байна.

- Pcap.findalldevs() – бүх идэвхитэй байгаа интерфэйсүүдийг гаргана.
- Pcap.lookupdev() – дефаулт интерфэйсийг гаргана
- Pcap.lookupnet() – IPv4 хаяг болон сабнетмаск гаргана
- Pcap.open.live() – пакет барих интерфэйсийг нээнэ
- Pcap.compile() – шүүлтүүрийг хөрвүүлнэ.
- Pcap.setfilter() – шүүлтүүрийг тааруулна.
- Pcap.loop() – барьсан пакетуудыг хадгална.

3.3.2 WinPcap

WinPcap нь дараах зүйлсүүдээс бүрддэг:

- Сүлжээний Драйвер Интерфэйс Тодорхойлолт(NDIS)-ийг сүлжээний адаптертай шууд холбогдон пакетууд уншихад ашигладаг Windows NT нэгдэл(Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7 гэх мэт)-ийн x86 мөн x86-64 драйверууд.
- Заагдсан үйлдлийн системүүд дахь доод-түвшний сангийн хэрэгжүүлэлтүүд.
- libpcap-ийн порт

Технологи	**Технологи wrapper сангууд**
C++	Libtins, Libcrafter
Perl	Net::Pcap
Python	python-libpcap, Pcapy
Ruby	PacketFu
Rust	pcap
Tcl	tclpcap, tcap, pktsrc
Java	jpcap, jNetPcap, Jpcap, Pcap4j
.Net	WinPcapNet, SharpPcap, Pcap.Net
Haskell	pcap
OCaml	mlpcap
Chicken Scheme	pcap
Common Lisp	PLOKAMI
Racket	SPeaCAP
Go	pcap

Хүснэгт 3.1: lipcap/WinPcap-ын wrapper сангууд

3.3.2.1 Pcap.Net

Pcap.Net бол C++/CLI болон Csharp дээр бичигдсэн WinPcap-ийн .net wrapper юм. Энэ нь бараг WinPcap-ийн бүх шинжүүдийг агуулна. Мөн пакет боловсруулалтын фрейм-воркыг агуулдаг. өмнө нь байсан WinPcap-ийн wrapper-уудаас ялгаатай давуу талуудтай мөн шинэ боломжууд болон алдааны засваруудад зориулж зөвхөн хөгжүүлэлтийн хэрэглээнд л ашиглагдаж байгаа. Бас кодчилол нь маш өндөр стандарттай байдаг.

Pcap.Net нь дараах онцлогтой:

Агуулна:

- Дотоод хост дээрх Live төхөөрөмжүүдийн жагсаалтыг авах.
- Live төхөөрөмжүүд(Сүлжээний төхөөрөмжүүд)-ээс пакетуудыг унших.
- Нийт сүлжээний урсгал барилтаас статистик мэдээллүүдийг хүлээн авах.
- Бүхэл пакетуудын статистик мэдээллүүдийг хүлээн авах.
- Өөр өөр sampling аргуудыг ашиглах.
- Berkley Пакет Шүүлтүүрүүдийг хэрэглэх.
- • Live төхөөрөмжүүд рүү пакетуудыг шууд илгээх эсвэл WinPcap-ийн илгээлтийн дарааллуудыг ашиглах.
- Pcap файлууд руу пакет демпинг хийх.

Агуулахгүй:

- AirPcap шинжүүд.
- Алсын Pcap шинжүүд.

Пакет боловсруулалт:

- Ethernet
- IPv4
- UDP
- TCP

- ARP
- IPv6
- GRE
- ICMP
- IGMP
- HTTP
- DNS

3.4 Протоколууд

3.4.1 ICMP протокол

RFC 792 стандартад тодорхойлогдсон Интернет Удирдлагын Мессеж протокол(ICMP) протокол нь хүсэлт явуулсан үйлчилгээ бэлэн биш байх эсвэл тухайн төхөөрөмж хариу өгөхгүй байна зэрэг алдааны мессеж илгээх байдлаар ажилладаг. TCP, IP сүлжээний холболтон дээгүүр хэд хэдэн төрлийн мессежүүдийг илгээдэг.

ICMP(Internet control message protocol) нь датаграммын дамжууллын алдааг үүсгүүрийн чиглэлд буцааж дамжуулна.ICMP протокол нь хост ба рутерийн үндсэн протоколд хамаарна.IP холболтгүй, бодит биш дамжууллыг үүсгэдэг.Интернэт сүлжээний мэдээллийг хянах ICMP протокол нь сүлжээний түвшний интернэт протоколын хэрэгжилт ба IP багцыг боловсруулах явцад гарах алдааны мэдээлэл ба бусад мэдээллүүдийг эх үүсгүүр тал руу мэдээлэх зорилготой богино мэдээний багцын дамжууллыг хангаж өгнө.Датаграмм дамжигдаагүй бол үүсгүүрт мэдээлэл ирнэ.

Үүсгүүрт алдааны болон бусад хяналтын мэдээлэл илгээх боловч алдааг засвар- ладдаггүй.Алдааг засах процесс үүсгүүрт хийгддэг.Датаграмм эх үүсгүүрийн хаяг ба зөвхөн хамгийн эцсийн хүлээн авуурын хаягийг өөрийн багцад тэмдэглэдэг.Энэ нь засврын төхөөрөмжүүдийн хаягийг агуулдаггүй учир ICMP протоколийн зарчмаар мэдээллийг зөвхөн эх үүсгүүр луу дамжуулна.ICMP протоколыг RFC792 стандартаар зөвлөмжилсөн.Хэрэв рутерээс ICMP протоколоор хүрэх төгсгөл хүртэмжгүй (destination unreachable) богино багцаар дамжигдах мэдээг үүсгүүрт өгсөн бол энэ нь рутер багцуу-

дыг төгсгөлийн очих талд дамжуулж чадахгүй.Ингээд буфер эх багцыг устгадаг. Очих талд хүртэмжгүй байх хоёр шалтгаан байдаг.

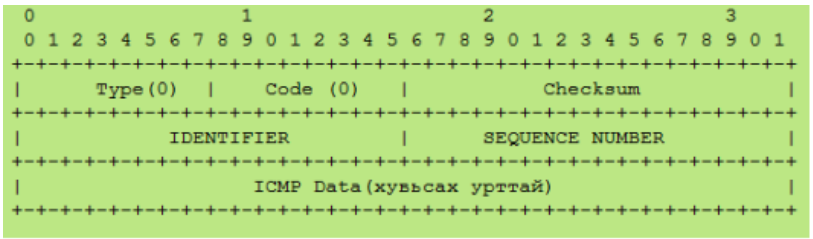
1. Ихэнх тохиолдолд эх үүсгүүрт хост байхгүй хаягийг зааж өгсөн
2. Зарим тохиолдолд рутер очих төгсгөлийн төхөөрөмж хүртэл замчлалыг алдаатай хийдэг

Хүрэх төгсгөлийн хүртэмжгүй мэдээнүүдийн үндсэн 4 төрөл байна.

1. Сүлжээний хүртэмжгүй: Багцын хаяглал эсвэл замчлалд алдаа гарсан
2. Хост хүртэмжгүй: дамжууллын алдаа жишээлбэл дэд сүлжээний маск алдаатай
3. Хост хүртэмжгүй: дамжууллын алдаа жишээлбэл дэд сүлжээний маск алдаатай
4. Порт хүртэмжгүй: Техник хангамжтай холбоотой. TCP үзүүр эсвэл порт залгаг-даагүй.

3.4.1.1 ICMP пакет

ICMP мессежүүд нь RFC792 ба RFC950-д тодорхойлсон ба албадмал STD5-д хамаарна.ICMP мессежүүд нь IP datagram-үүдыг илгээдэг.IP header нь үргэлж 1-ийн протокол дугаартай. ICMP мессеж-ээр datagram-ийг бичихийн тулд алдааны кодыг багтаасан.Checksum энэ нь 16 битийг агуулах бөгөөд ICMP мессеж эхлэхдээ ICMP төрлийн талбарыг өөртөө агуулдаг.Data ICMP мессеж-д мэдээлэл багтдаг.Ердийн энэ нь үүсгэгдсэн ICMP мессеж-д жинхэнэ Ip мессеж-ийн хэсэг багтана.Өгөгдлийн урт нь цөөн мессежтэй IP header length-ийг агуулсан IP datagram-ийн уртаар тодорхойлогдоно.



Зураг 3.5: ICMP протоколын бүтэц

3.4.2 UDP протокол

User Datagram Protocol буюу UDP нь компьютерийн сүлжээний тээвэрлэлтийн түвшний найдваргүй, холболтгүй нөхцөлд хэрэглэгддэг протокол юм. IP протоколын үйлчилгээг өргөжүүлж процессоос процесс руу холболт, бага зэргийн алдаа шалгалтыг гүйцэтгэдэг.

UDP нь маш энгийн протокол бөгөөд процессууд хоорондоо найдваргүйгээр зурвас явуулахыг хүсвэл UDP протоколыг ихэвчлэн ашигладаг

Найдвартай байдал, байнгын холболт шаардлагатай үед TCP гэх мэт протоколыг ашигладаг. UDP протоколын connectionless буюу холболт үүсгэдэггүй үйлчилгээний гол санаа нь UDP-гээр илгээгдсэн пакет бүр нь бусад пакетуудаасаа үл хамаарна. Ингэхдээ илгээж буй, хүлээн авч буй талууд нь ижилхэн байсан ч хоорондоо хамааралгүй байдаг. Хэрэглэгчийн datagram-уудыг дугаарладаггүй.

Түүнчлэн холболт үүсгэх, салгах процесс байдаггүй бөгөөд пакет бүр ялгаатай замуудаар дамжиж болно. Холболт үүсгэхгүй байхын тэг сул тал нь холболтын үед дамжуулж байгаа мэдээллийг урсгал байдлаар дамжуулж болохгүй бөгөөд пакет бүрд таарах байдлаар жижиг хэсгүүдэд задлан дамжуулах ёстой.

Source port number

Илгээгч дээр ажиллаж буй процессын портын хаяг. Урт нь 16 бит ба авах утгын хязгаар нь 0-65535 байна. Хэрвээ илгээгч нь клиент бол (клиент хүсэлт явуулахад) портын дугаар нь ихэнх тохиолдолд ephemeral port number байна. Илгээгч дээр ачаалж байгаа UDP нь энэ дугаарыг сонгосон байдаг. Хэрвээ илгээгч нь сервер бол портын дугаар нь ихэнх тохиолдолд well-known port number байна.

Destination port number

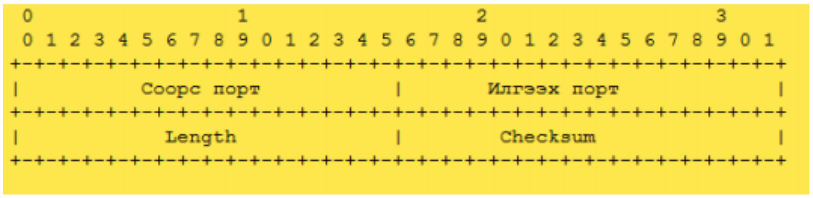
Хүлээн авагч дээр ачаалж байгаа процесын портын дугаар. Мөн 16 бит урттай. Хэрвээ хүлээн авагч нь сервер бол well-known port number, клиент бол ephemeral port number байна.

Length

Хэрэглэгчийн пакетын толгой хэсэг, өгөгдлийн хэсгийн нийлбэр. Хамгийн багадаа 8 байт (өгөгдөл байхгүй, зөвхөн толгой хэсэг).

Checksum

Checksum нь хэрэглэгчийн пакетийн (толгой хэсэг болон өгөгдлийн хэсэг хоёуланг-ийн) алдааг илрүүлдэг.



Зураг 3.6: UDP протокол

3.4.3 TCP протокол

Transmission Control Protocol нь TCP/IP-ийн хамгийн чухал протоколуудын нэг бөгөөд процесс хоорондын дамжууллыг гүйцэтгэдэг. Холболтод түшиглэсэн, өгөгдлийн дарааллыг хадгалдаг, алдааг шалгадаг, найвартай байдлыг хангаж өгсөн протокол юм. OSI загварын тээвэрлэлтийн түвшинд байрладаг.

TCP-г ашигладаг дээд түвшний протоколуудын жишээ гэвэл HTTP, HTTPS, SMTP, POP3, IMAP, SSH, FTP, Telnet зэрэг юм. TCP-ийн гол онцлог болох найвартай үйлчилгээ хэрэггүй гэсэн програм хангамжууд UDP-г хэрэглэх боломжтой.

Урсгалаар дамжуулж хүргэх TCP нь урсгалд түшиглэсэн протокол юм. IP нь пакетуудыг тус тусын зүйлс гэж үздэг бол TCP-г ашиглан дамжуулагч проецсс нь өгөгдлөө байтын урсгал болгон дам- жуулдаг. Ингэснээр 2 проецсс нь хийсвэр хоолойгоор холбогдсон мэт болдог.

Дамжуулагч, хүлээн авагч буфферууд

Дамжуулагч, хүлээн авагч процессууд нь ижил хурдтайгаар өгөгдөл бичиж, уншиж чадахгүй тул буффер ашиглан хадгалдаг. Хүлээн авагч, дамжуулагч гэсэн 2 буффер байдаг.

Сегмэнтүүд

Буфферийн тусламжтайгаар 2 үзүүрийн хурдыг тохиромжтой болгодог ч өгөгдөл дамжуулахаас өмнө нэг зүйлийг хийх хэрэгтэй. IP түвшин нь өгөгдлийг урсгал байдлаар биш пакет болгон дамжуулна. Иймд тээвэрлэлтийн түвшинд TCP нь байтуудыг багцлаад сегмэнт нэртэй пакет болгоно. Сегмэнтийг IP пакетад хийж дамжуулагддаг. Сегмэнтэд толгой хэсгийг нэмдэг. Сегмэнтүүд нь ижил хэмжээтэй байх албагүй.

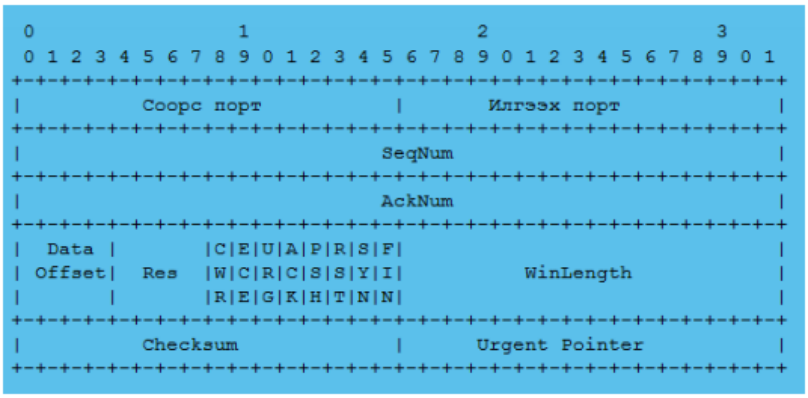
Full duplex дамжуулал

TCP-д өгөгдөл нь нэгэн зэрэг, хоёр зүгт дамжих боломжтой. TCP бүр нь дамжуулагч, хүлээн авагч баффер байдаг. Холболтод түшиглэсэн үйлчилгээ.

TCP нь холболтод түшиглэсэн үйлчилгээтэй. Дараах зүй тогтолтойгоор холбоо явагдана: Хоёр төхөөрөмжүүд дээр ажиллах TCP-нүүд нь холболт үүсгэнэ. Өгөгдлийг хоёр тийш дамжуулна.

Холболтыг таслана.

Энэ нь физик биш виртуаль холбоо болохыг анхаарна уу. Найдвартай үйлчилгээ TCP нь найдаратй тээвэрлэлтийн протокол. Acknowledgement механизм ашиглаж өгөгдлийг бүрэн бүтэн ирсэн эсэхийг шалгадаг.



Зураг 3.7: TCP протокол

Соорс портын хаяг

16 битийн урттай, дамжуулагч програмын порт хаягийг тодорхойлно. Илгээх портын хаяг

SeqNum

Сегмэнт дахь эхний байтын дугаарыг тодорхойлох 32 битийн талбар. Холболтыг үүсгэх үед initial sequence number (ISN)-г санамсаргүй тооны үүсгүүрээр гаргадаг. Тоо нь хоёр зүгт өөр өөр байж болно.

AckNum

Сегмэнтийн хүлээн авагчийн дараа нь ирнэ гэж буй байтын дугаарыг илтгэх 32 битийн талбар. x-р байтыг хүлээж авсан бол x+1 нь acknowledgement number байна. Өгөгдлийг үүнтэй хамт илтгээж болно.

Толгойн урт TCP толгой дахь 4 байтын word-н тоог илтгэх 4 битийн талбар. 20-с 60 байт байж болно. Иймээс тус талбарийн утга нь 5-аас 15 байж болно.

Нөөцөлсөн

Ирээдүйд хэрэглэхээр нөөцлөгдсөн, 6 битийн талбар.

Хяналт

6 өөр удирдлагын бит/флагийг тодорхойлно. Нэг эсвэл олныг тавьж болно.

Flags

9 ширхэг 1 битийн флагийг агуулна.

NS (1 бит) – ECN-nonce concealment protection (RFC 3540).
CWR (1 бит) – Congestion Window Reduced (CWR) флагийг илгээгч хост тавьж өгсөн, өөрөөр хэлбэл ECE флагтай TCP сегментийг хүлээж авч congestion control ме- ханизмд хариу өгөхийг (RFC 3168).

ECE (1 бит) – ECN-Echo indicates SYN флаг нь 1 бол TCP peer нь ECN-г джмэдэг SYN флаг нь 0 бол Congestion Experienced флагтай пакетийг хүлээж авсан (RFC 3168).

URG (1 бит) – Urgent pointer талбарын утга хүчинтэй болохыг

ACK (1 бит) – Acknowledgment талбарын утга хүчинтэй болохыг

PSH (1 бит) – Push буюу Хүлээж авч буй програм руу өгөгдлийг шууд "түлхэх"

RST (1 бит) – Reset буюу Холболтыг дахин эхлүүлэх

SYN (1 бит) – Synchronize буюу Sequence number-ийг синхрончлох.

FIN (1 бит) – Finish буюу Илгээгчээс ирж буй өгөгдөл дууссан

winLength

Нөгөө үзүүрт байх ёстой цонхны хэмжээг байтаар тодорхойлно. Хэмжээ нь 16 бит тул цонхны дээд утга 65,536 байт юм. Үүнийг receiving window (rwnd) гэлэг ба хүлээн авагч тодорхойлно. Хяналтын нийлбэр Тус 16 битийн талбар нь хяналтын нийлбэрийг агуулна. UDP-ээс ялгаатай нь TCP-д зайлшгүй байх ёстой. TCP pseudoheader-н хувьд протоколын талбарын утга 6 байна. Яаралтайн заагуур Urgent flag тавигдсан үед л хүчинтэй 16 битийн талбар. Urgent өгөгдөл байхад хэрэглэгдэнэ.

Бүлэг 4

Хэрэгжүүлэлтийн хэсэг

Бүлэг 5

Хавсралт

Номзүй