
“”

Гарчиг

	i
1 Эхлэл	1
1.1 Удиртгал	1
1.2 Зорилго	1
1.3 Зорилт	1
2 Онолын хэсэг	2
2.1 Сүлжээний тухай	2
2.2 TCP болон UDP протокол	3
2.2.1 TCP	3
2.2.2 Давуу тал	4
2.2.3 TCP холболт	5
2.2.4 TCP сегментийн бүтэц	5
2.3 UDP	8
3 Судалгааны хэсэг	10
3.1 Wireshark	10
3.2 PCATTCP	11
3.3 iPerf	16

Бүлэг 1

ЭХЛЭЛ

1.1 Удиртгал

TCP болон UDP пакет дамжуулах хүлээн авах програмуудыг өөр өөр хөгжүүлэгчид нээлттэй байдлаар хөгжүүлдэг. Төгсөлтийн ажлын судалгааны бүлэгт пакет дамжуулах хүлээн авах хэрэгслүүд болон нээлттэй эх кодын функц алгоритм функцуудыг ашиглан үр дүнг харуулсан болно.

1.2 Зорилго

Энэхүү төгсөлтийн ажлын зорилго нь TCP болон UDP протокол ашиглана хүссэн хэмжээ тоогоор пакетийг дамжуулах хүлээн авах хэрэглэгчийн интерфэйстэй програм хөгжүүлэхэд оршино. Судалгааны ажлаар сүлжээн дээгүүр пакет дамжуулах хүлээн авах хэрэгслүүд түүний алгоритмийн бүтэц мөн хэрхэн хөгжиж байгаа тухай судлах юм.

1.3 Зорилт

- Интернет сүлжээн дээгүүр өгөгдөл хэрхэн дамждаг тухай судлах.
- TCP болон UDP протоколоор өгөгдөл дамжуулах, хүлээн авах тухай судлах.
- Ижил төстэй програмын ажиллагаа алгоритмийн бүтцийг судлах.
- Хэрэглэгчийн интерфэйсийг зохиомжлон TCP болон UDP протоколоор пакет дамжуулах хүлээн авах програмыг хөгжүүлэх.

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.

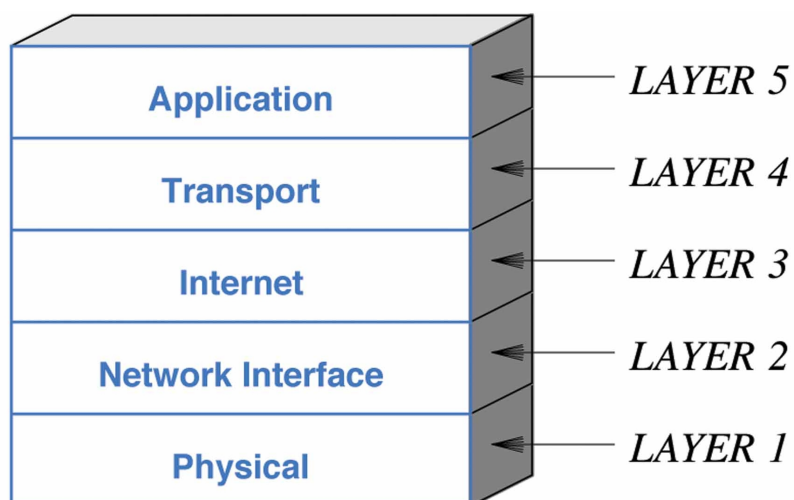
Компьютерийн сүлжээнд дохиог зөөх дөө өөрөөр дундын дамжууллын орчныг ашигладаг байна. Үүнд хатуу, шингэн, хий мөн плазм зэрэг олон янзын төлөв байж болно. Энэхүү сүлжээ нь WWW, видео, тоон аудио хандалт мөн хэрэглээний болон хадгалалтын серверүүд, принтерүүд, имэйл, мессеж програм гэх мэт өөр асар их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

2.2 TCP болон UDP протокол

2.2.1 TCP

TCP протокол нь холболтонд түшиглэсэн, өгөгдлийн дарааллыг хадгалдаг, алдааг шалгадаг, найдвартай байдлыг хангаж өгсөн протокол юм. TCP нь урсгалд түшиглэсэн протокол юм. IP нь пакетуудыг тус тусын зүйлс гэж үздэг бол TCP-г ашиглан дамжуулагч процесс нь өгөгдлөө байтын урсгал болгон дамжуулдаг. Дамжуулагч, хүлээн авагч процессууд нь ижил хурдтайгаар өгөгдөл бичиж, уншиж чадахгүй тул буффер ашиглан хадгалдаг. Хүлээн авагч, дамжуулагч гэсэн 2 буффер байдаг. TCP протокол нь таван давхаргаас бүрдэнэ.



Зураг 2.1: TCP протокол 5 давхарга

- Application(Хэрэглээний)
- Transport (Тээврийн)
- Network(Сүлжээний)
- Data-Link(Өгөгдлийн сувгийн)
- Physical(Физик)

APPLICATION LAYER

Хэрэглээний давхарга нь процесс хооронд мессэж дамжуулах үүрэгтэй ба хэрэглэгчийг үйлчилгээгээр хангадаг. Жишээ нь: e-mail илгээх, файл дамжуулах, веб хуудас гэх мэт.

TRANSPORT LAYER

Тээврийн давхаргад логик холболт нь end-to-end байдаг. Transport давхаргад нь application давхаргаас segment-ийг хүлээн авч үүнийг packet болгон өөрчлөн network давхаргад хүргэдэг.

NETWORK LAYER

Сүлжээний давхарга нь интернетийн үндсэн протокол буюу internet protocol(IP)-ийг агуулдаг. Уг давхаргад packet ийн төрлийг datagram гэж нэрлэдэг. Сүлжээний давхаргын гол үүрэг нь datagram ийг үүсвэрээс хүлээн авагч хооронд дамжуулах юм.

DATA-LINK LAYER

Өгөгдлийн сувгийн давхарга нь фреймийг нэг төхөөрөмжөөс нөгөөд дамжуулах үүрэгтэй. Өөрөөр хэлвэл шууд хоорондоо холбогдсон 2 төхөөрөмжийн хооронд өгөгдлийг дамжуулна.

PHYSICAL LAYER

Физик давхарга нь битүүдийг дохионд хувиргаж нэг төхөөрөмжөөс нөгөөд дамжуулах үүрэгтэй.

IP түвшин нь өгөгдлийг урсгал байдлаар биш пакет болгон дамжуулна. Иймд тээвэрлэлтийн түвшинд TCP нь байтуудыг багцлаад сегмент нэртэй пакет болгоно. Сегментийг IP пакетад хийж дамжуулагддаг. Сегментэд толгой хэсгийг нэмдэг. Сегментүүд нь ижил хэмжээтэй байх албагүй. TCP-д өгөгдөл нь нэгэн зэрэг, хоёр зүгт дамжих боломжтой. TCP бүр нь дамжуулагч, хүлээн авагч буффер байдаг. TCP нь найдвартай тээвэрлэлтийн протокол. Acknowledgement механизм ашиглаж өгөгдлийг бүрэн бүтэн ирсэн эсэхийг шалгадаг.

2.2.2 Давуу тал

- TCP програм нь хүлээн авсан, дамжуулсан сегментүүдийг бүртгэдэг ч сегментийн дугаарын талбар нь толгой хэсэгт л байдаггүй. Оронд нь sequence number/дарааллын дугаар, acknowledgement number гэсэн хоёр талбар бий. Энэ хоёр нь сегментийн биш байтын дугаарыг заана.
- Дамжуулж буй байт бүрийг TCP дугаарладаг. Эхний дугаар нь 0-ээс 2-ийн (31-1)

зэрэг хүртэлх тоонуудаас санамсаргүй авсан тоо байна.

- Байтуудыг дугаарласны дараа сегмент бүрд дарааллын дугаар оноодог. Үүний утга нь тус сегмент дахь эхний өгөгдлийг байтын дугаар болно. Сегментэд хэрэглэгчийн өгөгдөл байхгүй бол логикийн хувьд дарааллын дугаар байхгүй. Талбар байгаа ч утга нь хүчингүй гэсэн үг. Гэхдээ зарим үед acknowledgement-г хүлээж авахын тулд дарааллын дугаар шаардагдана. Ийм сегментийг холболтыг үүсгэлт, таслалт, дуусгалтын үед ашигладаг.
- Acknowledgment Number-н утга нь дараа хүлээж авах ёстой байтын дугаар юм. Acknowledgement number нь нийлбэр байдалтай.
- TCP нь урсгалын, алдааны, бөглөрөлтийн удирдлагатай. Урсгалын удирдлагад өгөгдлийг хүлээн авагч нь ирж буй өгөгдлийн хэмжээг зохицуулна. Байт дээр түшиглэсэн байна. Найдвартай байдлыг хангахын тулд алдааны удирдлагыг хэрэглэнэ. Бөглөрөлтийн удирдлагын хувьд дамжуулагч, сүлжээнээс хамаардаг.

2.2.3 TCP холболт

TCP-д холболтыг үүсгэхдээ three-way handshaking гэсэн аргачлалыг хэрэглэнэ. Эхлээд сервер нь өөрийн TCP-дээ холболт хийхэд бэлэн байгаагаа мэдэгдэнэ. Үүнийг passive open хийх хүсэлт гэнэ. Дараа нь клиент active open хийх хүсэлт гаргана. Энэ нь гурван үе шаттай байна:

- Клиент нь SYN сегментийг дамжуулна. Дарааллын дугаарыг синхрончлох үүрэгтэй. Өгөгдөл байхгүй ч дарааллын нэг дугаарыг эзлэнэ.
- Сервер нь SYN+ACK сегментийг дамжуулна. Нөгөө тийш дамжуулах зориулалттай SYN сегмент, мөн өмнөх SYN-г авсан гэж acknowledge хийх үүрэгтэй. Өгөгдөл байж болохгүй ч нэг дарааллын дугаарыг эзлэнэ.
- Клиент гурав дахь ACK сегментийг дамжуулна. Өгөгдөл байхгүй бол дарааллын дугаарыг эзлэхгүй.

2.2.4 TCP сегментийн бүтэц

TCP нь өгөгдлийн урсгалаас өгөгдөл хүлээн авч жижиглэн хуваагаад TCP толгой хэсгийг нэмж TCP сегмент үүсгэдэг. Дараа нь TCP сегментийг IP пакет болгон багцлаад

цааш дамжуулдаг. TCP-гийн сегментийг албан бусаар TCP пакет гэж хэлдэг ч албан ёсоор сегмент гэнэ.

TCP сегмент нь толгой болон өгөгдлийн хэсгээс тогтоно. TCP толгой хэсэг 10 зайлшгүй талбартай ба нэг туслах өргөтгөл талбартай.

Толгой хэсгийн дараа өгөгдлийн хэсэг орж ирнэ. Өгөгдлийн хэсгийн уртыг толгойд зааж өгдөггүй бөгөөд нийт IP датаграммын уртаас (IP толгойн хэсэгт заалттай байдаг) TCP толгой ба IP пакетийн толгойн уртуудыг хасч олдог.

TCP Segment Header Format								
Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Sequence Number							
64	Acknowledgment Number							
96	Data Offset	Res	Flags			Window Size		
128	Header and Data Checksum				Urgent Pointer			
160...	Options							

Зураг 2.2:

Acknowledgment Number-н утга нь дараа хүлээж авах ёстой байтын дугаар юм. Acknowledgement number нь нийлбэр байдалтай.

- Үүсвэр портын хаяг

16 битийн урттай, дамжуулагч програмын порт хаягийг тодорхойлно.

- Зорьсон портын хаяг

16 битийн урттай, дамжуулагч програмын порт хаягийг тодорхойлно

- Дарааллын дугаар

Сегмент 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.

- Зөвшөөрлийн дугаар

Сегментийн 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 илгээж болно.

- Толгойн урт

TCP толгой дахь 4 байтын word-н тоог илтгэх 4 битийн талбар. 20-с 60 байт байж болно. Иймээс тус талбарын 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.

- Цонхны хэмжээ

Нөгөө үзүүрт байх ёстой цонхны хэмжээг байтаар тодорхойлно. Хэмжээ нь 16 бит тул цонхны дээд утга 65,536 байт юм. Үүнийг receiving window (rwnd) гэдэг 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.

- Боломжууд

TCP-н толгой хэсэгт зайлшгүй биш 40 байт мэдээлэл байж болно.

- Доторлогоо

Өгөгдөл, толгой хэсгийг 32 битийн заагтай болгох зориулалттай, 0 битүүдээс тогтоно.

- Flags

9 ширхэг 1 битийн флагийг агуулна.

- NS (1 бит) – ECN-nonce concealment protection

- CWR (1 бит) – Congestion Window Reduced (CWR) флагийг илгээгч хост тавьж өгсөн, өөрөөр хэлбэл ECE флагтай TCP сегментийг хүлээж авч congestion control механизмд хариу өгөхийг
- ECE (1 бит) – ECN-Echo indicates
SYN флаг нь 1 бол TCP peer нь ECN-г дэмждэг
SYN флаг нь 0 бол Congestion Experienced флагтай пакетийг хүлээж авсан
- 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

2.3 UDP

User Datagram Protocol буюу UDP нь компьютерийн сүлжээний тээвэрлэлтийн түвшний найдваргүй, холболтгүй нөхцөлд хэрэглэгддэг протокол юм. IP протоколын үйлчилгээг өргөжүүлж процессоос процесс руух 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 IP давхарга заасан нь хоёроос үйлчилгээг үзүүлж байна. Энэ нэмэлт нь шалгалтын чадавхи мэдээллийн бүрэн бүтэн ирсэн гэдгийг нотлохын тулд өөр өөр хэрэглэгчийн хүсэлтийг ялгаж болон туслах портын дугаарыг олгодог. Харин UDP л пакетуудыг, энэ нь их бага зурвасын өргөн нэмэлт зардал болон хоцрогдол байна гэсэн үг илгээдэг. Харин пакетууд нь алдсан, хувь пакетууд нь илгээгч болон хүлээн авагчийн хооронд дундуур өөр өөр замыг улмаас, үр дүнд тулд гарч хүлээн авч болно. UDP нь хожимдол нь ийм тоглоомын, дуу, видео харилцаа холбоо, сөрөг ойлголт чанарт нөлөөлж байгаа ч зарим өгөгдлийн

алдагдлаас зовох болно гэж чухал юм үзэж сүлжээний програмууд нь хамгийн тохиромжтой протокол юм. Зарим тохиолдолд, форвард алдаа залруулах арга нь зарим алдсан хэдий ч, аудио, видео чанарыг сайжруулахын тулд ашигладаг. Найдвартай байдал, байнгын холболт шаардлагатай үед TCP протоколыг ашигладаг. TCP нь бие даасан багц руу их хэмжээний өгөгдлийг багц зөрчсөн шалгаж ба алдагдсан пакетуудыг resending зөв дараалалд оруулах пакетуудыг жагсаах зэрэг үйлдлүүдийг үйлчилгээний улмаас интернэт холболт ихэнх нь ашиглаж зонхилох протокол болоод байна. Гэхдээ эдгээр нэмэлт үйлчилгээ нэмэлт өгөгдөл нэмэлт ачааллын хувьд нь зардлаар ирж, саатал хоцрогдол гэж нэрлэдэг.

UDP толгойн талбарууд

- Source port (16 bits)

Дамжуулагчийн процессийн дугаар.

- Destination port (16 bits)

Хүлээн авагчийн процессийн дугаар.

- Message length (16 bits)

Байтаар багцын уртыг заана. (UDP толгойн мэдээлэл болон дата)

- Checksum (16 bits)

TCP – тэй адил зарчмаар ажиллана.

Гэхдээ сонголт байж болно. Θ/x 0 утгатай байвал шалгалт хийдэггүй

-

Berkeley Socket Interface (BSI)

Сүлжээний программыг хэрхэн хөгжүүлэх вэ ?

- Хамгийн зөв зам бол стандартуудыг мэдэх ба хамгийн их тархсан протоколуудыг ашиглах явдал юм.
- Дата линк үед Ethernet ашиглах.
- Сүлжээний үед IP ашиглах.
- Транспорт үед IP ашиглах.
- Хэрэглээний программын үед Berkeley Socket Interface стандарт API ашиглах.

Бүлэг 3

Судалгааны хэсэг

3.1 Wireshark

Вайршарк (Wireshark) нь сүлжээний пакет-д дүн шинжилгээ хийх зориулалт бүхий програм юм. Сүлжээний пакет-д дүн шинжилгээ хийхдээ энэхүү програм нь сүлжээн дээгүүр дамжигдаж буй пакетуудыг чагнаж, цуглуулаад тэдгээр пакет өгөгдөл (packet data)-ийг боломжит хамгийн дэлгэрэнгүй байдлаар задлан харуулдаг.

Сүлжээний пакет шинжлэгч (packet analyzer) нь сүлжээний кабел дээгүүр дамжигдаж буй дээд түвшинд харуулах, хэмжих зориулалттай багаж юм. Вайршарк (wireshark) нь сүлжээний пакет шинжлэгч (packet analyzer) програмуудын дундаас шилдэг програмуудынх нь нэгд зүй ёсоор багтдаг юм.

Вайршарк (wireshark) програмын зарим түгээмэл хэрэглээ

- Сүлжээнд үүссэн асуудлыг оношлох, тодруулахад
- Сүлжээний аюулгүй байдалтай холбоотой асуудлыг хянах, илрүүлэхэд
- Хөгжүүлэгчид шинэ протокол хөгжүүлэх, хэрэгжүүлэх явцдаа шалгах зориулалтаар
- Компьютерийн сүлжээг хэрхэн ажилладаг талаар суралцаж буй хүмүүс сургалтын зориулалтаар гэх мэт.

Вайршарк (wireshark)-ын ажиллагааны онцлог

- Windows болон Unix үйлдлийн системүүд дээр ажиллана.

- Сүлжээний интерфэйс картууд (Network Interface Card – NIC) дээгүүр дамжиж буй пакет өгөгдлийг барьж авах (capture), цуглуулж авна.
- Вайршарк (Wireshark) програмтай ижил үйлдэл хийдэг tcpdump/Windump гэх мэт сүлжээний өгөгдөлд анализ хийх програмуудын цуглуулсан packet өгөгдлүүдийг нээнэ, анализ хийнэ.
- Пакет (Packet) өгөгдлийн 16тын тооллын системээр илэрхийлэгдсэн (hex) файлаас вайршарк (wireshark) програм руу импорт хийнэ.
- Пакет (Packet) өгөгдлийг ашиглагдаж буй протоколоор нь дэлгэрэнгүйгээр харуулна.
- Цуглуулж авсан пакет (packet) өгөгдлийг хадгална.
- Цуглуулсан пакет (packet) өгөгдлөө хэсэгчлэн эсвэл бүтнээр нь олон төрлийн файлын төрлийн (file format) сонголттойгоор экспорт хийнэ.
- Олон төрлийн шалгуур үзүүлэлт, параметр ашиглан пакет (packet) өгөгдлөөс шүүлт (filter) хийнэ.
- Шүүлтүүр (filter) хийсэн пакет (packet) өгөгдлийн үр дүнг өнгөөр ялгаж харуулна.
- Төрөл бүрийн статистик үзүүлэлтүүдийг автоматаар үүсгэнэ гэх мэт олон үйлдлүүдийг нэг дороос хийх боломжтой.

3.2 PCATTCP

Test TCP (TTCP) нь хоёр системийг хооронд нь TCP ба UDP холболтыг хэмжих команд мөрийг суурилуулсан хэрэгсэл юм. Энэ нь анх 1984 онд BSD үйлдлийн системд зориулагдан боловсруулагдаж байсан. Үүнээс TTCP хөгжүүлэн Windows үйлдлийн системд зориулан PCATTCP (Printing Communication Association TCP) хэрэгслийг гаргасан байна.

PCATTCP дамжуулах машинаас өгөгдлийг пакет хэлбэрээр хүлээн авах машинруу илгээж статистик аргаар хэмждэг. Хэрэглэгч дамжуулалтын төгсгөлд илгээсэн пакетийн тоо, тэдгээр пакетийн хэмжээг сонгох боломжтой. Үүгээр пакетийн хэмжээ янз бүрийн холболтыг тестлэх боломжтой.

Хэрэглэгч PCATTCP илгээх болон хүлээн авах тохиргоог өөрийн дураар өөрчлөх боломжтой. Жишээ нь pcattcp сонсох порт нь анхний утгаараа 5001-ийг сонсдог үүнийг хэд портоор ч сольж болно.

```
C:\Users\gggre\Desktop\PCATTCP-0114>pcattcp -t 192.168.1.6
PCAUSA Test TCP Utility V2.01.01.14 (IPv4/IPv6)
  IP Version : IPv4
Started TCP Transmit Test 0...
TCP Transmit Test
  Transmit : TCPv4 0.0.0.0 -> 192.168.1.6:5001
  Buffer Size : 8192; Alignment: 16384/0
  TCP_NODELAY : DISABLED (0)
  Connect : Connected to 192.168.1.6:5001
  Send Mode : Send Pattern; Number of Buffers: 2048
  Statistics : TCPv4 0.0.0.0 -> 192.168.1.6:5001
16777216 bytes in 7.558 real seconds = 2167.68 KB/sec +++
numCalls: 2048; msec/call: 3.779; calls/sec: 270.960
```

Зураг 3.1: PCATTCP илгээх

```
C:\Users\gggre\Desktop\PCATTCP-0114>pcattcp -r
PCAUSA Test TCP Utility V2.01.01.14 (IPv4/IPv6)
  IP Version : IPv4
Started TCP Receive Test 0...
TCP Receive Test
  Local Host : DESKTOP-UHJVIFR
*****
Listening...: On TCPv4 0.0.0.0:5001

Accept : TCPv4 0.0.0.0:5001 <- 192.168.1.6:2168
Buffer Size : 8192; Alignment: 16384/0
Receive Mode: Sinking (discarding) Data
Statistics : TCPv4 0.0.0.0:5001 <- 192.168.1.6:2168
16777216 bytes in 12.198 real seconds = 1343.18 KB/sec +++
numCalls: 3413; msec/call: 3.660; calls/sec: 279.801
```

Зураг 3.2: PCATTCP хүлээн авах

Зураг 3.1, 3.2-д PCATTCP-ийг ашиглана дотоод сүлжээнд TCP протоклоор пакет дамжуулсан.

PCATTCP-ийг анхы утгаар TCP протокол ашиглана пакет дамжуулахад. Хүлээн авагч тал нь 5001 портыг сонсож буфферийн хэмжээ нь 8192 байна. Харин илгээгч тал 5001 портруу 16777216 байт мэдээллийг дамжуулсан.

Зураг 3.6, 3.7-д PCATTCP-ийг ашиглана дотоод сүлжээнд UDP протоклоор пакет дамжуулсан.

[illegible]

Зураг 3.3: Wireshark шүүсэн мэдээлэл

```

▼ Frame 33076: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
  Interface id: 0 (\Device\NPF_{B66F30F2-BB44-4853-90C4-E81A84B45796})
  Encapsulation type: Ethernet (1)
  Arrival Time: Oct 11, 2017 05:26:04.855954000 Pacific Daylight Time
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1507724764.855954000 seconds
  [Time delta from previous captured frame: 0.000084000 seconds]
  [Time delta from previous displayed frame: 0.000084000 seconds]
  [Time since reference or first frame: 25.296414000 seconds]
  Frame Number: 33076
  Frame Length: 54 bytes (432 bits)
  Capture Length: 54 bytes (432 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]
  [Coloring Rule Name: TCP]
  [Coloring Rule String: tcp]
▼ Ethernet II, Src: Azurewaw_63:bc:c0 (48:5d:60:63:bc:c0), Dst: HonHaiPr_4e:9d:9b (c0:18:85:4e:9d:9b)
  ▼ Destination: HonHaiPr_4e:9d:9b (c0:18:85:4e:9d:9b)
    Address: HonHaiPr_4e:9d:9b (c0:18:85:4e:9d:9b)
    .... ..0. .... ..0. .... .. = LG bit: Globally unique address (factory default)
    .... ..0. .... ..0. .... .. = IG bit: Individual address (unicast)
  ▼ Source: Azurewaw_63:bc:c0 (48:5d:60:63:bc:c0)
    Address: Azurewaw_63:bc:c0 (48:5d:60:63:bc:c0)
    .... ..0. .... ..0. .... .. = LG bit: Globally unique address (factory default)
    .... ..0. .... ..0. .... .. = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.6
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
    .... 00.. = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 40
  Identification: 0x5c9c (23708)

```

Зураг 3.4:

PCATTCSP-ийг анхы утгаар UDP протокол ашиглана пакет дамжуулахад. Хүлээн авагч тал нь 5001 портыг сонсож буфферийн хэмжээ нь 8192 байна. Харин илгээгч тал 5001 портруу 16777216 байт мэдээллийг дамжуулсан.

```

  ▾ Flags: 0x02 (Don't Fragment)
    0... .. = Reserved bit: Not set
    .1... .. = Don't fragment: Set
    ..0... .. = More fragments: Not set
  ▾ Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
  ▾ Header checksum: 0x1adb [validation disabled]
    [Good: False]
    [Bad: False]
  Source: 192.168.1.2
  Destination: 192.168.1.6
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
  ▾ Transmission Control Protocol, Src Port: 5001 (5001), Dst Port: 2168 (2168), Seq: 1, Ack: 4381, Len: 0
    Source Port: 5001
    Destination Port: 2168
    [Stream index: 7]
    [TCP Segment Len: 0]
    Sequence number: 1 (relative sequence number)
    Acknowledgment number: 4381 (relative ack number)
    Header Length: 20 bytes
  ▾ Flags: 0x010 (ACK)
    000... .. = Reserved: Not set
    ...0... .. = Nonce: Not set
    ...0... .. = Congestion Window Reduced (CWR): Not set
    ...0... .. = ECN-Echo: Not set
    ...0... .. = Urgent: Not set
    ...1... .. = Acknowledgment: Set
    ...0... .. = Push: Not set
    ...0... .. = Reset: Not set
    ...0... .. = Syn: Not set
    ...0... .. = Fin: Not set
    [TCP Flags: *****A****]
  Window size value: 256
  [Calculated window size: 65536]
  [Window size scaling factor: 256]

```

Зураг 3.5:

```

C:\Users\gggre\Desktop\PCATTCP-0114>pcatttcp -u -r
PCAUSA Test TCP Utility V2.01.01.14 (IPv4/IPv6)
  IP Version   : IPv4
Started UDP Receive Test 0...
UDP Receive Test
  Protocol     : UDPv4
  Port         : 5001
  Buffer Size   : 8192; Alignment: 16384/0
  recvfrom     : UDPv4 <- 192.168.1.6:63481

```

Зураг 3.6: PCATTCP UDP протокол илгээх

```

C:\Users\gggre\Desktop\PCATTCP-0114>pcatttcp -u -t 192.168.1.6
PCAUSA Test TCP Utility V2.01.01.14 (IPv4/IPv6)
  IP Version   : IPv4
Started UDP Transmit Test 0...
UDP Transmit Test
  Transmit     : UDPv4 0.0.0.0 -> 192.168.1.6:5001
  Buffer Size   : 8192; Alignment: 16384/0
  Send Mode     : Send Pattern; Number of Buffers: 2048
  Statistics    : UDPv4 0.0.0.0 -> 192.168.1.6:5001
16777216 bytes in 13.820 real seconds = 1185.50 KB/sec +++
numCalls: 2050; msec/call: 6.903; calls/sec: 148.333

```

Зураг 3.7: PCATTCP UDP протокол хүлээн авах


```

▼ Frame 3743: 834 bytes on wire (6672 bits), 834 bytes captured (6672 bits) on interface 0
  Interface id: 0 (\Device\NPF_{B66F30F2-BB44-4853-90C4-E81A84B45796})
  Encapsulation type: Ethernet (1)
  Arrival Time: Oct 11, 2017 05:15:49.323874000 Pacific Daylight Time
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1507724149.323874000 seconds
  [Time delta from previous captured frame: 0.000013000 seconds]
  [Time delta from previous displayed frame: 0.000016000 seconds]
  [Time since reference or first frame: 2.629604000 seconds]
  Frame Number: 3743
  Frame Length: 834 bytes (6672 bits)
  Capture Length: 834 bytes (6672 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:udp:data]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
▼ Ethernet II, Src: Azurewav_63:bc:c0 (48:5d:60:63:bc:c0), Dst: HonHaiPr_4e:9d:9b (c0:18:85:4e:9d:9b)
  > Destination: HonHaiPr_4e:9d:9b (c0:18:85:4e:9d:9b)
  > Source: Azurewav_63:bc:c0 (48:5d:60:63:bc:c0)
  Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.6
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 820
  Identification: 0x26fc (9980)
  ▼ Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
  Fragment offset: 7400
  Time to live: 128
  Protocol: UDP (17)

```

Зураг 3.8:

```

▼ Header checksum: 0x89c7 [validation disabled]
  [Good: False]
  [Bad: False]
  Source: 192.168.1.2
  Destination: 192.168.1.6
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
▼ [6 IPv4 Fragments (8200 bytes): #3738(1480), #3739(1480), #3740(1480), #3741(1480), #3742(1480), #3743(800)]
  [Frame: 3738, payload: 0-1479 (1480 bytes)]
  [Frame: 3739, payload: 1480-2959 (1480 bytes)]
  [Frame: 3740, payload: 2960-4439 (1480 bytes)]
  [Frame: 3741, payload: 4440-5919 (1480 bytes)]
  [Frame: 3742, payload: 5920-7399 (1480 bytes)]
  [Frame: 3743, payload: 7400-8199 (800 bytes)]
  [Fragment count: 6]
  [Reassembled IPv4 length: 8200]
  [Reassembled IPv4 data: cdb113892008861550434155534120504341545443502050...]
▼ User Datagram Protocol, Src Port: 52657 (52657), Dst Port: 5001 (5001)
  Source Port: 52657
  Destination Port: 5001
  Length: 8200
  ▼ Checksum: 0x8615 [validation disabled]
    [Good Checksum: False]
    [Bad Checksum: False]
  [Stream index: 2]
▼ Data (8192 bytes)
  Data: 5043415553412050434154544350205061747465726e2021...
  [Length: 8192]

```

Зураг 3.9:

3.3 iPerf

iPerf хэрэгслийг сүлжээний өргөн зурвас чадлыг хэмжих болон тохируулахад өргөн хэрэглэдэг.

Энэ нь параметруудийг тааруулах, буфер болон TCP, UDP, SCTP протоколуудыг дэмжих бөгөөд бусад хэрэгслээс илүү ямар ч сүлжээн дахь гүйцэтгэлийн хэмжилтийг хийж чаддагаараа чухал ач холбогдолтой.

Iperf нь клиент болон серверийн функцтай бөгөөд тэдгээрийн хооронд өгөгдлийн урсгал үүсгэж нэвтрүүлэх чадварыг хэмждэг.

Iperf энгийн гаралтын дамжуулсан өгөгдлийн хэмжээ болон нэвтрүүлэх чадварын хэмжилтийн талаар цаг хугацаатайн илэрхийлэх тайлан агуулсан байдаг. UDP: UDP протоколын хүчин чадлыг турших үед Iperf нь хэрэглэгчийн датаграмийн хэмжээг тодорхойлж датаграм дамжуулах чадвар болон пакетийн алдагдлын үр дүнг тодорхойлдог.

TCP: TCP протоколын хүчин чадлыг турших үед чадлыг турших үед iperf нь ашигтай ачааллын бүтээмжийг хэмждэг.

IPerf нь C программын хэл дээр бичигдсэн нээлттэй эх программ бөгөөд Линукс, Юуникс, Виндоус зэрэг төрөл бүрийн платформ дээр ажилладаг. Бэлэн эх код нь хэрэглэгч хэмжилтийн аргачлалыг судлах боломжийг олгодог.

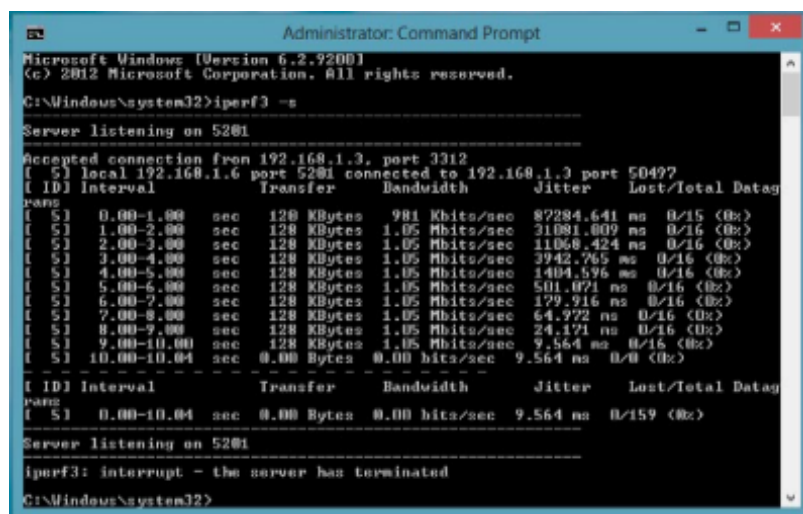
Iperf – ийн анхны хувилбарыг NLANR/DAST академи боловсруулсан.

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>iperf3 -c 192.168.1.6
Connecting to host 192.168.1.6, port 5201
[ 4] local 192.168.1.3 port 3269 connected to 192.168.1.6 port 5201
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-1.00 sec  2.00 MBytes 16.7 Mbits/sec
[ 4] 1.00-2.00 sec  1.88 MBytes 15.7 Mbits/sec
[ 4] 2.00-3.00 sec  2.12 MBytes 17.9 Mbits/sec
[ 4] 3.00-4.00 sec  2.12 MBytes 17.8 Mbits/sec
[ 4] 4.00-5.00 sec  2.25 MBytes 18.9 Mbits/sec
[ 4] 5.00-6.00 sec  2.12 MBytes 17.8 Mbits/sec
[ 4] 6.00-7.01 sec  2.12 MBytes 17.8 Mbits/sec
[ 4] 7.01-8.00 sec  2.12 MBytes 17.9 Mbits/sec
[ 4] 8.00-9.01 sec  2.00 MBytes 16.6 Mbits/sec
[ 4] 9.01-10.00 sec 2.25 MBytes 19.1 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-10.00 sec 21.0 MBytes 17.6 Mbits/sec
[ 4] 0.00-10.00 sec 20.8 MBytes 17.4 Mbits/sec
iperf Done.
  
```

Зураг 3.10: iPerf клиент IPv4



```

Administrator: Command Prompt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Windows\system32>iperf3 -s

Server listening on 5201

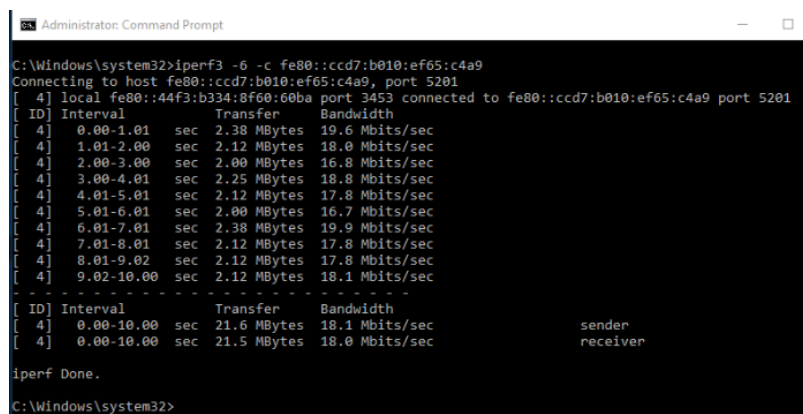
Accepted connection from 192.168.1.3, port 3312
[ 5] local 192.168.1.6 port 5201 connected to 192.168.1.3 port 50497
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Data
[ 5] 0.00-1.00 sec  128 KBytes    981 Kbits/sec   87284.641 ns  0/15 (0%)
[ 5] 1.00-2.00 sec  128 KBytes    1.05 Mbits/sec  31081.009 ns  0/16 (0%)
[ 5] 2.00-3.00 sec  128 KBytes    1.05 Mbits/sec  11068.424 ns  0/16 (0%)
[ 5] 3.00-4.00 sec  128 KBytes    1.05 Mbits/sec  3942.765 ns   0/16 (0%)
[ 5] 4.00-5.00 sec  128 KBytes    1.05 Mbits/sec  1404.596 ns   0/16 (0%)
[ 5] 5.00-6.00 sec  128 KBytes    1.05 Mbits/sec  501.071 ns    0/16 (0%)
[ 5] 6.00-7.00 sec  128 KBytes    1.05 Mbits/sec  179.916 ns    0/16 (0%)
[ 5] 7.00-8.00 sec  128 KBytes    1.05 Mbits/sec  64.772 ns     0/16 (0%)
[ 5] 8.00-9.00 sec  128 KBytes    1.05 Mbits/sec  24.171 ns     0/16 (0%)
[ 5] 9.00-10.00 sec 128 KBytes    1.05 Mbits/sec  9.564 ns      0/16 (0%)
[ 5] 10.00-10.04 sec 0.00 Bytes    0.00 bits/sec   9.564 ns      0/0 (0%)
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Data
[ 5] 0.00-10.04 sec 0.00 Bytes    0.00 bits/sec   9.564 ns      0/159 (0%)

Server listening on 5201

iperf3: interrupt - the server has terminated
C:\Windows\system32>

```

Зураг 3.11: iPerf сервер IPv4



```

Administrator: Command Prompt

C:\Windows\system32>iperf3 -c fe80::ccd7:b010:ef65:c4a9
Connecting to host fe80::ccd7:b010:ef65:c4a9, port 5201
[ 4] local fe80::44f3:b334:8f60:60ba port 3453 connected to fe80::ccd7:b010:ef65:c4a9 port 5201
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.00-1.01 sec  2.38 MBytes   19.6 Mbits/sec
[ 4] 1.01-2.00 sec  2.12 MBytes   18.0 Mbits/sec
[ 4] 2.00-3.00 sec  2.00 MBytes   16.8 Mbits/sec
[ 4] 3.00-4.01 sec  2.25 MBytes   18.8 Mbits/sec
[ 4] 4.01-5.01 sec  2.12 MBytes   17.8 Mbits/sec
[ 4] 5.01-6.01 sec  2.00 MBytes   16.7 Mbits/sec
[ 4] 6.01-7.01 sec  2.38 MBytes   19.9 Mbits/sec
[ 4] 7.01-8.01 sec  2.12 MBytes   17.8 Mbits/sec
[ 4] 8.01-9.02 sec  2.12 MBytes   17.8 Mbits/sec
[ 4] 9.02-10.00 sec  2.12 MBytes   18.1 Mbits/sec
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.00-10.00 sec  21.6 MBytes   18.1 Mbits/sec
[ 4] 0.00-10.00 sec  21.5 MBytes   18.0 Mbits/sec

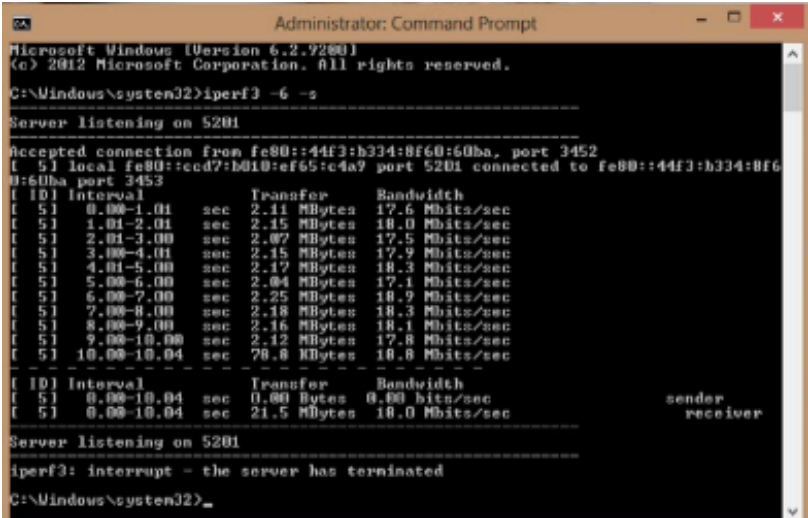
iperf Done.

C:\Windows\system32>

```

Зураг 3.12: iPerf клиент IPv6

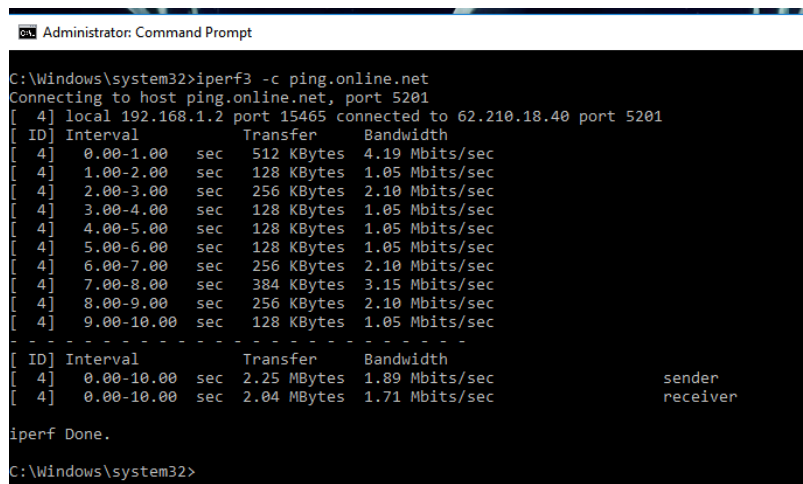
iPerf-д зориулагдсан интернет серверүүд байх бөгөөд серверүүд тус тусдаа Линукс, Юуникс, Виндоус зэрэг платформуудыг дэмждэг.



Зураг 3.13: iPerf сервер IPv6

Europe								
iPerf3 server	Characteristics	Localization	Datacenter	Hosting	Speed	Port	IP version	Contact
bouygues.iperf.fr	Linux 4.10	France Île-de-France	Telehouse 2 Paris Voltaire		10 Gbit/s	5200 TCP to 5209 TCP	IPv4 and IPv6	@labreinfo
ping.online.net ping6.online.net ping-90ms.online.net ping6-90ms.online.net	IPv4 only IPv6 only IPv4 + 90ms latency IPv6 + 90ms latency	France Île-de-France	Online Vitry DC3		10 Gbit/s	5200 TCP/UDP to 5209 TCP/UDP	IPv4 or IPv6	@mikmak
speedtest.serverius.net (Port 5002: add -p 5002)		Netherlands	Serverius datacenter		10 Gbit/s	5002 TCP/UDP	IPv4 and IPv6	@serveriusby
iperf.eenet.ee		Estonia	EENet Tartu			5201 TCP/UDP	IPv4 only	@EENet_HITSA
iperf.volia.net		Ukraine	Volia Kiev			5201 TCP/UDP	IPv4 only	@voliaofficial
Asia								
iPerf3 server	Characteristics	Localization	Datacenter	Hosting	Speed	Port	IP version	Contact
iperf.it-north.net	Linux 3.16 Debian 8	Kazakhstan	Petropavl		1 Gbit/s	5200 TCP/UDP to 5209 TCP/UDP	IPv4 only	@brauninger.AE
iperf.biznetnetworks.com	Linux 2.6.32 CentOS 6	Indonesia	Biznet - Midplaza Cimanggis		1 Gbit/s	5201 TCP to 5203 TCP	IPv4 and IPv6	@Biznet Networks
Americas								
iPerf3 server	Characteristics	Localization	Datacenter	Hosting	Speed	Port	IP version	Contact
iperf.scottlinux.com		USA California	Hurricane Fremont 2		1 Gbit/s	5201 TCP/UDP	IPv4 and IPv6	@scottlinux
iperf.the.net		USA California	Hurricane Fremont 1			5201 TCP/UDP	IPv4 and IPv6	HE forums

Зураг 3.14: iPerf интернет серверүүд



```
Administrator: Command Prompt

C:\Windows\system32>iperf3 -c ping.online.net
Connecting to host ping.online.net, port 5201
[ 4] local 192.168.1.2 port 15465 connected to 62.210.18.40 port 5201
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-1.00 sec    512 KBytes  4.19 Mbits/sec
[ 4] 1.00-2.00 sec    128 KBytes  1.05 Mbits/sec
[ 4] 2.00-3.00 sec    256 KBytes  2.10 Mbits/sec
[ 4] 3.00-4.00 sec    128 KBytes  1.05 Mbits/sec
[ 4] 4.00-5.00 sec    128 KBytes  1.05 Mbits/sec
[ 4] 5.00-6.00 sec    128 KBytes  1.05 Mbits/sec
[ 4] 6.00-7.00 sec    256 KBytes  2.10 Mbits/sec
[ 4] 7.00-8.00 sec    384 KBytes  3.15 Mbits/sec
[ 4] 8.00-9.00 sec    256 KBytes  2.10 Mbits/sec
[ 4] 9.00-10.00 sec   128 KBytes  1.05 Mbits/sec
-----
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-10.00 sec  2.25 MBytes  1.89 Mbits/sec
[ 4] 0.00-10.00 sec  2.04 MBytes  1.71 Mbits/sec
iperf Done.

C:\Windows\system32>
```

Зураг 3.15: iPerf сервертэй холболт тогтоосон