

Modeliranje računalniških omrežij
Študijsko leto 2013/2014

Modeliranje IPv6 omrežij

Poročilo seminarske naloge

Nihad Kerić, 63090347
Miha Novak, 63100268
Gregor Bahor, 63090049
Darko Janković, 63100176

Ljubljana, 16. december 2013

Kazalo

1 Naloga

Modelirajte bolj kompleksne primere IPv6 omrežja s pomočjo INET ogrodja v orodju OMNeT++.

2 Opis 3 zgledov za modeliranje IPv6 omrežij v INET ogrodju.

2.1 Pv6NCLients

V datoteki NclientsEth.ned oz. NclientsPPP.ned imamo skonfigurirano omrežje s tremi IPv6 usmerjevalniki ter n IPv6 odjemalcev (komuniciranje preko aplikacije TelnetApp). IPv6 odjemalci so strežnik in n klienti, kar je prikazano na sliki spodaj. Stevilo n-klientov v našem testnem primeru se določi v konfiguraciji: [General]*.n=10, lahko tudi spremenimo čas izvajanja simulacije v našem primeru 'sim-time-limit=168h' v datoteki omnetpp.ini. Vsi klienti so vezani na en usmerjevalnik r1. Med klienti in strežnikom so med seboj zaporedno vezani trije usmerjevalniki, strežnik je vezan na usmerjevalnik r3. Ob zagonu simulacije se vzpostavi stanje omrežja, nato pa se prične seja med strežnikom in klienti. Seje se izmenjujejo med različnimi klienti. Pri testiranju različno velikem številu klientov n=2,10,100,200 in pri simulacijskem času 168h ni prišlo do napak. Simulacija NclientsPPP je identična po zgradbi omrežja NclientsEth. Razlika med omrežjema se pojavi v načinu povezave fiberline ali ethernetline in stem se spremeni čas potovanja paketov in propustnost kanalov. Simulacija NclientsEth ima definirano hitrost prenosa podatkov datarate=100Mbps in je počasnejša od NclientsPPP, katera ima hitrost prenosa podatkov datarate=1Gbps.

2.2 IPv6Bulk

Omrežje sestavljajo strežnik, usmerjevalnik in trije odjemalci. Strežnik in vsi trije odjemalci so povezani z usmerjevalnikom, obstaja pa tudi direktna povezava med strežnikom in enim izmed odjemalcev. Vse povezave so tipa in/out, hitrost prenosa podatkov po kanalu pa je 10Mbps z zakasnitvijo 0.1us. Ves promet v omrežju usmerja usmerjevalnik, ki usmerja tudi promet med strežnikom in odjemalcem.

Pred zagonom simulacije lahko izbiramo med različnimi implementacijami TCP (Transmission Control Protocol):

- TCP, je protokol za nadzor prenosa podatkov, ki zagotavlja, da se informacije med prenosom ne izgubijo, ne spremenijo in da se informacije vnovič dostavijo, če je prišlo med prenosom do napake.
- TCP_lwIP, TCP lightweight IP, je široko uporabljen odprtokodni TCP/IP protokolni sklad oblikovan za uporabo v vgrajenih sistemih.
- TCP_NSC, implementacija TCP, ki je bila razvita v okviru NSC projekta (Network Simulation Cradle project)

Opazujemo lahko izvajanje NDP (Neighbor Discovery Protocol) in TCP seje (trosmerno rokovanje, prenos podatkov).

Paketi, ki se prenašajo pri NDP (Neighbor Discovery Protocol):

- RSpacket (Router Solicitation)
- RApacket (Router Advertisement)
- NSpacket (Neighbour Solicitation)
- NApacket (Neighbour Advertisement)

Paketi, ki se prenašajo pri TCP seji

- SYN
- SYN + ACK

2.3 Nclients

Nclient ima dve mreži : 1) NClientsEth.ned 2) NClientsPPP.ned

2.3.1 NClientsEth

Pri tistem omrežju imamo komunikacijo med odjemalcem in strežnikom, ali pač z n odjemalcov pa enim strežnikom preko 3 usmerjevalnika. Kateri so povezani preko ipv6 protokola, naslovi so razdelni na 8 naslovo. V NClientsEth.ned fielu imam dve vrsti kanalov (channel): - fiberline -ethernetline Kanali imata iste attribute z različnimi nastavitvi. fiberline (delay= 1us in datarate= 512 Mbps) ethernetline (delay= 0.1us in datarate= 10 Mbps) Usmerjevalniki komunicirajo preko tistih kanalov ,prvo uspostave povezavo

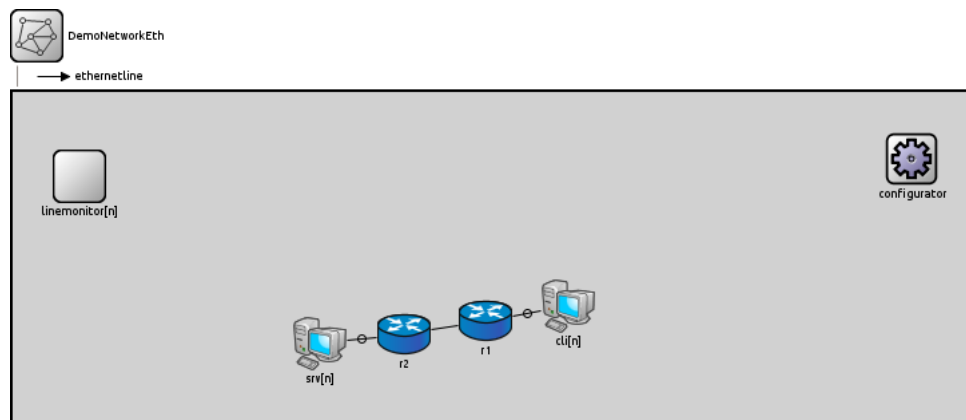
pošiljanjem različnih paketov kot so: NSpacket , RApacket, RSpacket , SYN, SYN+ACK. Po vzpostavljenoj povezavi se začnejo pošiljati paketov. Strežnik pošlje paket proti odjemalcu kateri pol odgovori z pošiljanjem paketa ACK. Isto se zgodi pri pošiljanu paketov z strani odjemalca.

2.3.2 NClientPPP

Tudi v tem omrežju imamo komunikacijo med n odjemalcev in strežnikom preko tri usmerjevalnika, ipv6 naslov je razdeljen na 8 naslovo kateri so krajši od naslovo prve konfiguracije v temu se razlikujeta. V NClientsEth.ned fielu imam kanal (channel): -fiberline z nastavitvemi: (delay= 1us in datarate= 512 Mbps)

3 Podrobna analiza enega od zgledov

Za analizo smo izbrali zgled *demonetworketh*.



Omrežje je sestavljeno iz naslednjih gradnikov:

- configurator tipa FlatNetworkConfigurator6
- r1 tipa Router6
- r2 tipa Router6
- cli[n] tipa StandardHost6
- srv[n] tipa StandardHost6

- `linemonitor[n]` tipa `TCPDump`

FlatNetworkConfigurator6

Konfigurira Ipv6 naslove in posredovalne tabele.

Router6

Predstavlja Ipv6 usmerjevalnik.

StandardHost6

Ipv6 gostitelj s TCP, SCTP in UDP plastmi in aplikacijami.

TCPDump

Pregledovanje vsebine paketov.

4 Podroben opis razpoložljivih gradnikov INET ogrodja

Opisali smo gradnike iz primera *demonetworketh*.

4.1 NetworkLayer6

Omrežje vsebuje elemente, ki so ključnega pomena pri klientih, usmerjevalnikih in strežnikih. Modul `NetworkLayer6.ned` predstavlja Ipv6 omrežno plast in je sestavljen iz elementov: `-import inet.networklayer.ipv6.IPv6ErrorHandling;`
`-import inet.networklayer.ipv6.IPv6;` `-import inet.networklayer.icmpv6.IPv6NeighbourDiscovery;`
`-import inet.networklayer.icmpv6.ICMPv6;`
 SLIKA

-Ipv6: modul setavlja klasifikacijski obrejekt (modul) `IPv6Datagram`, ki predstavlja glavo paketa IPv6 protokola. Ko modul `ipv6` pošlje paket višjemu nivoju (TCP ali UDP protokol) po ISO/OSI omrežnem modelu ga opremi z izvornim in ponornim naslovom. Opisani elementi povezujejo 3. omrežni (IPv6) in 4. transportni (TCP/UDP) nivo po OSI/ISO omrežnem modelu.
 -`IPv6ErrorHandling`: Napake pridejo v obliki sporočila, modul se uporablja za beleženje napak na omrežnem nivoju.
 -`NeighbourDiscovery`: modul se uporablja za izvajanje vseh naloge, povezanih z odkritje sosedov in brez naslovno auto konfiguracijo. Neighbour discovery paketi so sami poslani in obdelani stem modulom. Ko Ipv6 prejme enega, posreduje paket naprej k IPv6 Neighbor Discovery.

-Icmpv6: modul, ki služi za pošiljanje zahtev »echo request« na omrežnem nivoju. Zahteva bo poslana na vrat pingIn z proloženo IPv6ControlInfo. Odgovor »echo reply« bo sprejet, ko sporočilo bo poslano skozi vrata pingOut.

4.2 RoutingTable6

Ta gradnik predstavlja IPv6 usmerjevalno tabelo. Vsak gostitelj ali usmerjevalnik v omrežju ima natanko en primerek tega gradnika. Je preprosti (simple) modul brez vrat in se aktivira s klicem funkcije. Ima funkcije za branje in posodabljanje tabele, ter za usmerjanje *unicast* in *multicast* prometa. Usmerjevalna tabela se inicializira na vrednosti, ki jih prebere iz datoteke. Med simulacijo se seveda iz nje bere ter se jo spreminja, skladno z uporabljenimi usmerjevalnimi protokoli.

Modul *RoutingTable6.ned* ima 3 parametre:

- xml routingTable = default(xml(«routingTable/>")); // datoteka z začetno usmerjevalno tabelo v obliki XML
- bool isRouter; // če je nastavljena na *true*, je omogočeno posredovanje IP
- bool forwardMulticast = default(false); // posredovanje multicast prometa

4.3 InterfaceTable

Gradnik predstavlja tabelo omrežnih vmesnikov. Omrežne vmesnike kot so PPPInterface in drugi, dinamično registrirajo ustrezni moduli. Tabela vsebuje lastnosti vmesnikov, ki se ne navezujejo na lastnosti protokolov. Specifične lastnosti protokolov, IPv4 in IPv6 se hranijo v gradniku RoutingTable oziroma v zgoraj opisanemu gradniku RoutingTable6. Modul InterfaceTable nima vrat in se aktivira s klici funkcij, ki se nahajajo v pripadajočem C++ razredu. Modul je opisan samo z enim parametrom displayAddresses, ki je tipa boolean in pove ali naj prikaže IP naslove na povezavah ali ne.

4.4 Notification Board

Z uporabo Notification Boarda, lahko moduli sedaj med seboj obvestita o dogodkih, kot je sprememba usmerjevalnih tabel, statusa vmesnika, konfigu-

racije vmesnik, brezžične predajanja itd. Notification Board ima natanko en primerek v klientu ali strežniku model in deluje kot posrednik med modulom. Lahko se pojavijo in moduli, ki so zainteresirani za učenje o tistih spremembe. Sej morajo biti imenovan kot "notificationBoard", da deluje pravilno. Moduli lahko "naročite" na kategorij sprememb (npr. "usmerjevalne table spremeniti ali radijska postaja postala prazna"). Ko pride do take spremembe, ki ustrezajo modul (npr. RoutingTable ali modul fizični sloj) bo pustil NotificationBoard. Obveščanja dogodki so združene v "kategoriji". Vsaka kategorija je določena s celim številom (zdaj je to pripisano v izvorno kodo preko ENUM). Stranke, ki želijo prejemanje obvestila je treba izvajati (podrazred od) v INotifiable vmesnik, dobimo kazalec na NotificationBoard, in se naročite na kategorije jih zanima s klicom subscribe() metodo v NotificationBoard. Obvestila bodo dostavljeni na receiveChangeNotification() metodo stranke (re definirani od INotifiable).

5 Postavitev 4 modelov omrežij

5.1 Drevesna topologija (tree topology)

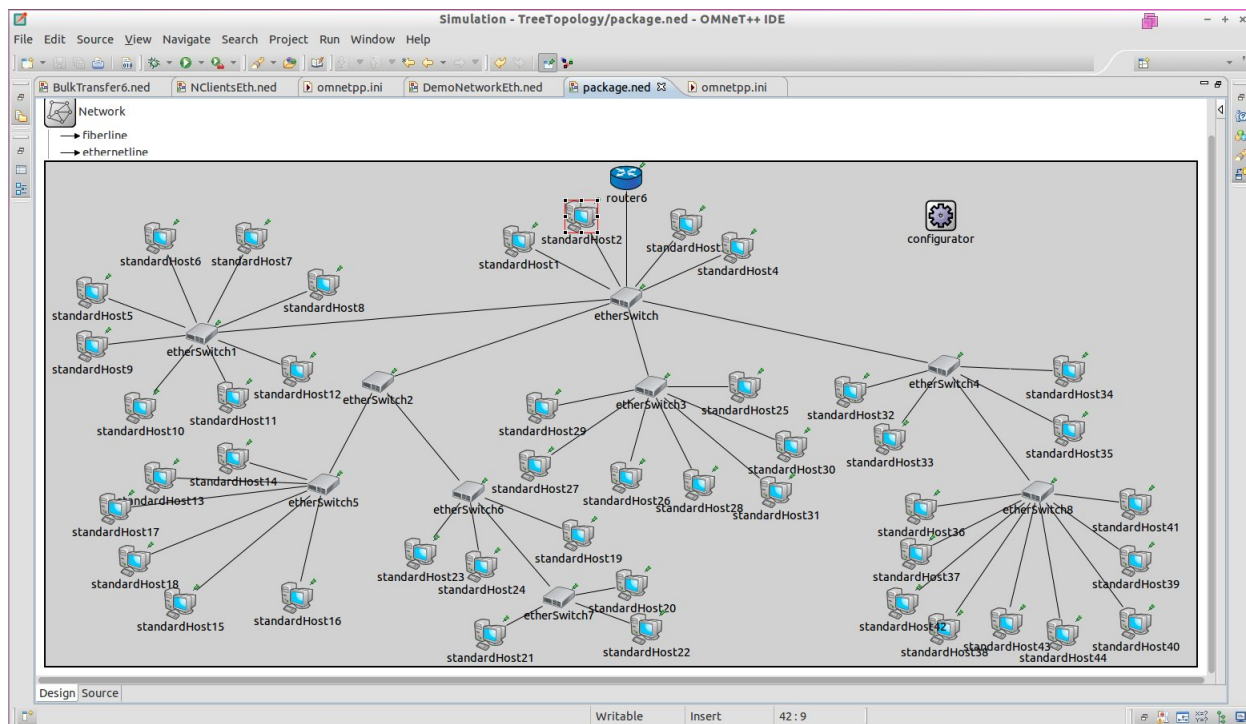
Pri drevesni topologiji gre za kombinacijo *star* in *bus* topologij, kar izboljša skalabilnost omrežja. Ime je dobila po podobnosti z binarnim drevesom, ker imamo eno korensko vozlišče, ki ima lahko več otrok, pri čemer so tudi otroci lahko koreni svojega podomrežja. Primer takšnega omrežja je DNS sistem.

Prednosti:

- Skalabilnost - lahko dodajanje naprav na vozlišča
- Možne povezave točka-točka (point to point)
- Zaradi večjih nivojev lažje vzdrževanje omrežja

Slabosti, ki se lahko pojavijo pri takšni topologiji:

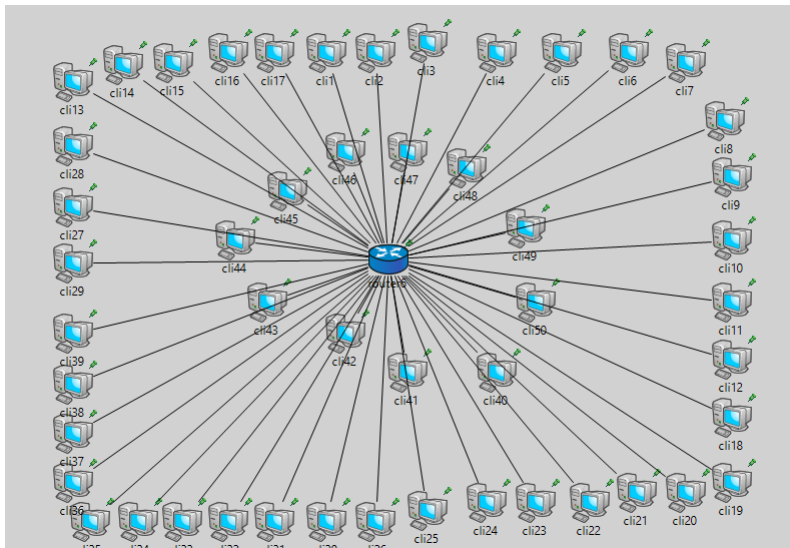
- Vzdrževanje lahko postane težje če se omrežje razprostira na večji fizični površini
- Če hrbtenica omrežja odpove, bo tudi celotno omrežje nedelujoče
- Relativno težja konfiguracija v primerjavi z ostalimi topologijami



Slika 1: Primer omrežja tree

5.2 Zvezna topologija (star topology)

Omrežje star je sestavljeno iz 50 klientov in usmerjevalniku, kateri je v vozlišču omrežja. Vsi klienti so povezani z »fiberline« povezavo in komunicirajo preko tega usmerjevalnika. Fiberline povezava omogoča hitrost prenosa 1Gbps v 1 us. Na začetku usmerjevalnik prvo vzpostavi povezavo z vsakim klientom pošiljanjem paketa NS in pol vsak klient naredi isto pošlje NS paket usmerjevalniku. Zdaj imamo vzpostavljeno povezavo z usmerjevalnikom. Da bi klient »1« komuniciral z klientom »2« prvo pošlje signal SYN na to usmerjevalnik preveri povezavo do klienta »2« kateri mu odgovori pošiljanjem paketa NS. Usmerjevalnik pošlje SYN-ACK klientu kateri je zahteval povezavo v našem primeru je to klient »1«, zdaj imamo povezana klienta »1« in »2«. Zvezda poizkuša povečati stabilnost omrežja z uvedbo centralnega vozla v katerega so povezani vsi odjemalci. V primeru, da izpade zveza med poljubnim odjemalcem in centralnim vozlom, ostanejo ostale zveze nespremenjene. Vsi odjemalci torej komunicirajo direktno le z glavnim vozlom - odgovoren je za prenos podatkov do ostalih uporabnikov omrežja.



Slika 2: Primer omrežja star

Prednosti:

- Preprosta namestitve in razširitve
- Hiter, direkten prenos podatkov

Slabosti:

- Izpad centralnega vozla privede do izpada celotnega omrežja
- Odjemalci potrebujejo dodatno strojno opremo

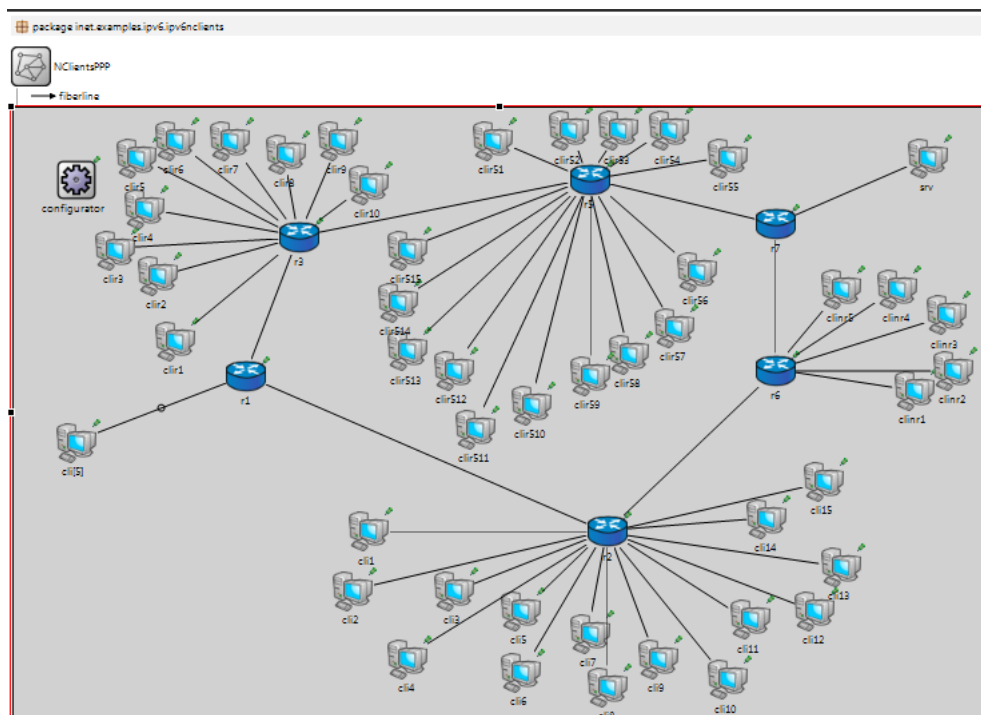
5.3 Obročna topologija (star topology)

Omrežje NClientsPPP (Slika spodaj) je sestavljeno iz pet usmerjevalnikov (r1,r2,r3,r5,r6,r7; so moduli tipa router6), ki so skupaj povezani v obliki topologije OBROČ. Na vsakem usmerjevalniku imamo določeno število računalnikov(clientov) StandardHost6, na koncu imamo eden strežnik.

V omrežje NClientsPPP so moduli povezani med seboj z povezavo fiberline, katera omogoča hitrost prenosa podatkov $\text{datarate}=512\text{Mbps}$. Ko zaženemo simulacijo najprej se inicializirajo vsi moduli nato usmerjevalniki razpošlejo NS pakete vsem povezanim usmerjevalnikom in nato še vsak usmerjevalnik pošlje NS paket svojim klientom, kateri so povezani na njega.

Nato se vzpostavi three-way handshake med enim računalnikom in strežnikom. Ko želi kient cli1 na eni strani našega omrežja vzpostaviti povezavo s strežnikom srv na drugi strani, najprej pošlje sporočilo SYN. Sporočilo (SYN) potuje do prevega usmerjevalnika R2. Nato usmerjevalnik R2 preveri povezljivost do usmerjevalnika R6 s sporočilom NS. Ko mu ta odgovori s sporočilom NA lahko usmerjevalnik R2 z NextHop se sklicuje na naslednji najbližji usmerjevalnik in posreduje prejeto sporočilo SYN od klienta cli1 dalje. Enak postopek izmenjave sporočil NS in NA se zgodi pri prenosu sporočila SYN od usmerjevalnika R6 in R7 do strežnika srv. Tako strežnik srv končno prejme sporočilo SYN in lahko nanj odgovori s sporočilom SYN-ACK. Sporočilo SYN-ACK potuje nazaj po prej zgrajeni sejni poti (izmenjava sporočil NS in NA) preko usmerjevalnika R7 in R6 do usmerjevalnika R2. Usmerjevalnik R2 mora izgraditi še zadnji del celotne poti do klienta cli1. To naredi tako, da zopet pošlje sporočilo NS klientu cli1 in prejme kot odgovor sporočilo NA. Končno lahko usmerjevalnik R2 posreduje sporočilo SYN-ACK klientu cli1. Ker komunikacija med klientom in strežnikom poteka po protokolu TCP mora klient odgovoriti na prejeto sporočilo SYN-ACK od strežnika srv s sporočilom ACK, ki potuje po vspostavljeni povezavi preko usmerjevalnikov R2, R6 in R7 do ponora srv.

Oba komunicirata omejen čas, nato pa rušita povezavo. Konfiguracija je nastavljena tako, da postopek traja $\text{sim-time-limit}=168\text{h}$. Usmerjevalne tabele = routingTable6 modul se uporablja za hranjene podatkovnih struktur, kot so lista destinacij, lista sosedov in lista predpon.



Slika 3: Primer omrežja ring

5.4 Topologija vodila (bus topology)

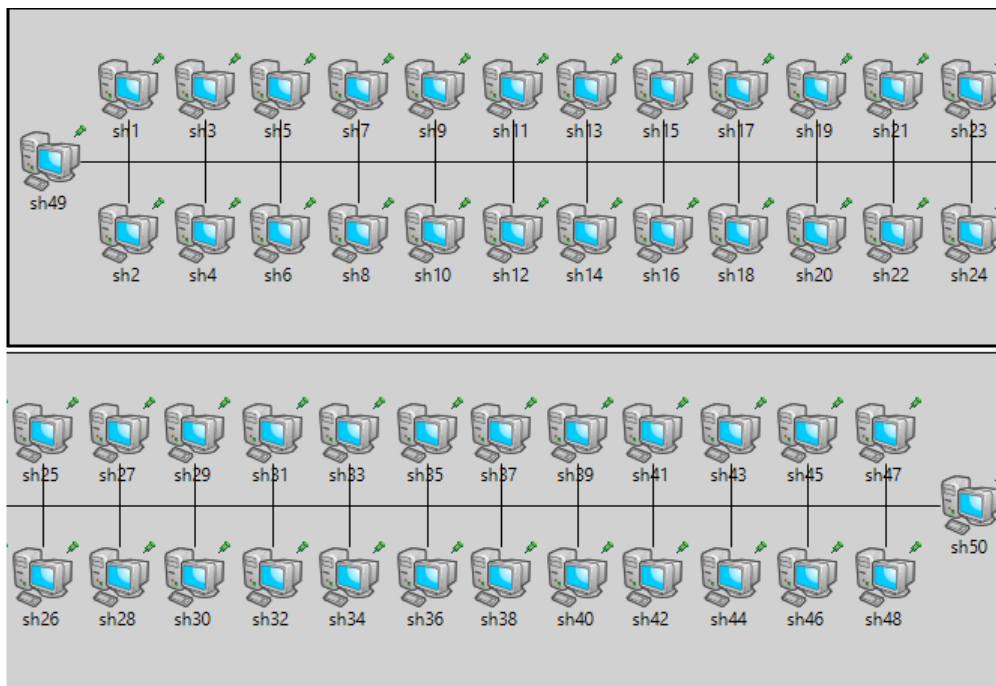
Topologija vodila je najpreprostejša omrežna arhitektura za katero je značilno, da so vse naprave priključene na enoten skupen medij po katerem poteka prenos podatkov. Ponavadi je to ethernet. Topologija omrežja je večkrat prikazana linearno saj so naprave pogosto povezane v ravni liniji. V celotnem omrežju lahko samo ena postaja naenkrat oddaja pakete, dobijo pa jih vse postaje priključene na vodilo. Ta topologija omrežja se danes vedno manj uporablja.

Prednosti:

- Preprosta postavitvev in razširitev omrežja
- Zahteva najmanj omrežnega kabla
- Poceni realizacija

Slabosti:

- Okvara vodila posledično pomeni izpad celotnega omrežja
- Število možnih odjemalcev je omejeno z dolžino kabla
- Težko je odkriti vzrok težav, ker si vse naprave delijo isti medij in posledično dobijo vse pakete
- Na vodilo nastajajo trki (kolizija)



Slika 4: Primer omrežja, ki temelji na topologiji vodila

6 Literatura