

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Praktické cvičenie č. 3

Roman Panenka

Študijný program: Počítačové a komunikačné systémy a siete

Predmet: Bezdrôtové komunikačné systémy

Ročník: Ing. 1.

Akademický rok: 2010/2011

Obsah

1. Cieľ zadania	1
2. Technické a programové prostriedky	1
3. Význam použitých skratiek	1
4. Smerovacie protokoly v ad-hoc bezdrôtových sieťach.....	2
4.1. Smerovací protokol AODV	3
4.2. Smerovací protokol OLSR	3
5. Postup cvičenia	4
6. Výsledky skúmania.....	5
6.1. Experimentovanie s AODV.....	5
6.2. Experimentovanie s OLSR.....	7
7. Záver.....	10

1. Cieľ zadania

Tretie praktické cvičenie je zamerané na oboznámenie sa a experimentovanie s niektorými smerovacími protokolmi v ad-hoc bezdrôtových sieťach. Jedná sa o dva z najčastejšie používaných protokolov:

- AODV (Ad hoc On-Demand Distance Vector)
- OLSR (Optimized Link State Routing).

Študent sa prakticky oboznámi s protokolmi AODV a OLSR a zo základným princípom týchto protokolov. Na základe praktického cvičenia odskúša aj základný rozdiel v princípe fungovania použitých protokolov.

2. Technické a programové prostriedky

Pre úspešné dosiahnutie cieľov zadania a na efektívny výskum parametrov intenzity signálu, potrebuje študent nasledujúce technické a programové prostriedky:

- Notebook s OS Microsoft Windows / OS Linux
- Interná sieťová karta
- Softvérové vybavenie
 - o Balík UoBWinAODV pre OS Windows ¹
 - o Balík UoBWinAODV pre OS Linux ²
 - o Podporné programové knižnice³
 - o Balík OLSRD pre OS Windows⁴
 - o Balík OLSRD pre OS Linux⁵

3. Význam použitých skratiek

- Ad-hoc sieť – necentralizovaná sieť mobilných uzlov bez stálej topológie
- 802.11 – súbor wifi štandardov
- AP – Access point – zariadenie, ku ktorému sa pripájajú klienti bezdrôtovej siete
- Peer-to-peer komunikácia – komunikácia uzlov, pri ktorej vystupujú aplikácie ako rovnocenné, teda nie je tu prítomný server
- AODV a OLSR – smerovacie protokoly v bezdrôtových ad-hoc sieťach

¹ <http://www.aodv.org/download/UoBWinAODV-0.15.zip>

² <http://www.aodv.org/download/jadhoc-0.11.tar.gz>

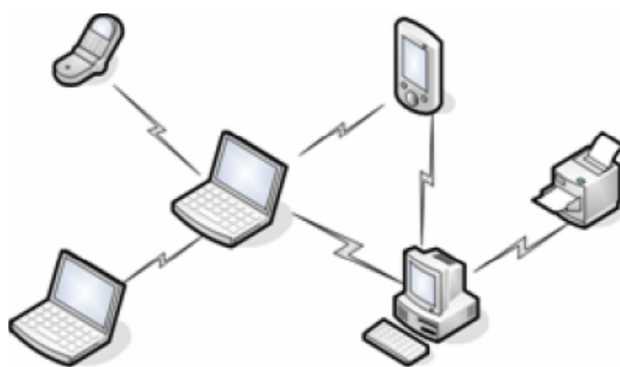
³ mfc71d.dll - <http://www.dll-files.com/dllindex/mfc71d.zip?oVHjXHXlJX>
msvcr71d.dll - <http://www.dll-files.com/dllindex/msvcr71d.zip?oVHjXHXlFW>

⁴ <http://www.olsr.org/releases/0.5/olsrd-0.5.6-r3-pre-cac1df64dcd5-setup.exe>

⁵ <http://www.olsr.org/?q=download>

4. Smerovacie protokoly v ad-hoc bezdrôtových sieťach

Ad hoc siete pozostávajú zo staníc (uzlov), ktoré navzájom komunikujú. Tieto siete je možné v každom momente reprezentovať neorientovaným grafom, ktorého vrcholy reprezentujú stanice (uzly) a hrany reprezentujú spojenia. Na graf ad hoc siete nie sú kladené žiadne ďalšie požiadavky. Uzly a spojenia ad hoc siete môžu v čase pribúdať a miznúť. Sieť je formovaná dynamicky pohyblivými uzlami, ktoré sú prepojené cez bezdrôtové linky bez použitia existujúcej infraštruktúry alebo centralizovanej administratívy. Uzly sa pohybujú náhodne, preto topológia siete sa môže meniť rýchlo a nepredvídateľne.



Obrázok 1 - Príklad ad hoc siete

Hlavné vlastnosti ad-hoc sietí:

- Dynamická zmena topológie (mobilita)- každý uzol sa voľne premiestňuje a zároveň komunikuje s ostatnými uzlami
- Nezávislosť od vybudovanej infraštruktúry - každý uzol pracuje v distribuovanom peer-to-peer móde, správa sa ako smerovač a generuje nezávisle dáta.
- Rovnaké médium - Uzly komunikujú bezdrôtovo prostredníctvom toho istého média napr. rádiové vlny, infračervené žiarenie ...
- Viacnásobné skákanie (multihopping) - cesta zo zdroja k cieľu prejde cez niekoľko ďalších uzlov (každý sa pritom správa ako smerovač)
- Obmedzenie vzdialenosti uzlov - v niektorých aplikáciách môžu ad hoc siete narásť do veľkého počtu uzlov a je dôležité, aby sa kvalita spojenia medzi rôznymi uzlami nezhoršila vplyvom zväčšenia alebo zmenšenia počtu uzlov.
- Energetické obmedzenie - mnohé ad hoc uzly majú limitovanú zásobu elektrickej energie a nemajú schopnosť si generovať svoju vlastnú energiu. Táto vlastnosť má veľký vplyv na životnosť tejto siete.

4.1. Smerovací protokol AODV

Smerovací protokol AODV (The Ad-hoc On-Demand Distance Vector) je navrhnutý pre použitie v ad-hoc mobilných a wifi sieťach. Umožňuje komunikovať dvom uzlom hoci nie sú vzájomne viditeľný, a to prostredníctvom ich susedov. AODV je protokol reaktívny (reactive), čiže cesty v sieti sú vytvárané iba ak ich je treba. Cesty (routes) sú ukladané v klasických smerovacích tabuľkách (routing tables), čiže jeden záznam na jednu destináciu.

AODV s použitím tzv. poradových čísel (sequence numbers) hľadá a vytvára najkratšie cesty medzi uzlami a zabráňuje vznikanie smerovacích slučiek.

Keďže topológia a počet uzlov v ad-hoc sieti sa dynamicky menia, je významnou vlastnosťou AODV časové značkovanie stavov jednotlivých uzlov a ciest medzi nimi. Ak nie je cesta dlhšie používaná, vyprší jej platnosť a je odstránená z tabuľky. O tejto skutočnosti môže uzol notifikovať susedné uzly.

Hľadanie efektívnych ciest je docielené pomocou kontrolných informácií posielaných medzi uzlami, ktoré si nimi aktualizujú smerovaciu tabuľku. Medzi tieto kontrolné informácie patria:

- RREQ (routing request) – uzol posiela všetkým susedom keď sa pýta na cestu k druhému uzlu
- RREP (routing reply) – je jedna odpoveď suseda, ktorý pozná cestu k požadovanému druhému uzlu
- RERR (route error) – informácia o strate linky na nejaký uzol pre ostatné uzly
- HELLO – kontrolné správy o stave liniek k susedom

4.2. Smerovací protokol OLSR

Smerovací protokol OLSR (Optimized Link State Routing Protocol) je navrhnutý pre použitie v ad-hoc mobilných a wifi sieťach. Umožňuje komunikovať dvom uzlom hoci nie sú vzájomne viditeľný, a to prostredníctvom ich susedov. OLSR je protokol proaktívny (proactive), čiže pozná cesty k všetkým uzlom už predtým ako ich potrebuje. Požíva správy typu Hello a Topology Control (TC) na vyhľadávanie a rozhlasovanie ciest v bezdrôtovej sieti. Jednotlivé uzly siete sa z týchto správ učia najkratšie cesty ku všetkým uzlom v sieti (presnejšie iba susedov, cez ktorých majú k uzlom komunikovať).

Takýto prístup ku smerovacej konvergencii môže spôsobiť veľké zahltenie pásma smerovacími aktualizáciami. Linkové protokoly v pevných sieťach (napr. OSPF a IS-IS) riešili tento problém zvolením si vyhradeného smerovača na každej linke, s ktorým si tieto informácie vymieňali. Pri bezdrôtových sieťach by bolo toto riešenie neúčinné, keďže komunikácia na každom uzle prechádza jedným bezdrôtovým rozhraním.

OLSR protokol na každom uzle určí dvojuzlovú susednú informáciu (2-hop neighbor information) a určí množinu tzv. MPR (multipoint relays). Uzly si vyberú MPR tak aby existovala cesta ku každému dvojuzlovému susedovi cez uzol vybraný ako MPR. Tieto MPR potom preposielajú informácie o zmene topológie (TC messages).

5. Postup cvičenia

Pokiaľ ešte nie sú požadované programy nainštalované, je potrebné ich nainštalovať. Balík *UoBWinAODV* pod OS Windows vyžaduje sprevádzkovanosť podporu .NET Runtime⁶. Ďalej je potrebné doinštalovať špeciálny ovládač sieťovej karty, označovaný ako *Passthru driver*, ktorý je súčasťou balíka. Presné inštrukcie inštalácie sa nachádzajú v súbore *INSTALL* v inštalačnom adresári.

Pre správne nakonfigurovanie programu je potrebné poznať presné označenie sieťovej karty. Toto je možné získať z výstupu programu *factotum.exe* spusteného s parametrom */enum*. Program zobrazí všetky dostupné sieťové karty, ich systémové označenie a popis. Zaznamenajte si označenie sieťovej karty, ktorú budete používať. Toto označenie sa nachádza na druhom riadku výstupu a má nasledovný formát:
`\DEVICE\{FF3FF804-1038-421E-B8C1-2CE21DDC85A1}`

Ďalším krokom je úprava konfiguračného súboru *UoBWinAODV.cfg*, kde je potrebné nastaviť IP adresu sieťového rozhrania, prípadne iné parametre protokolu AODV. Dôležitým parametrom je *ExecutionMode*, ktorý treba zmeniť na hodnotu *GUI*.

Po úspešnej inštalácii a nakonfigurovaní je potrebné sieťovú kartu prepnúť v OS do režimu ad-hoc (*Control Panel - Network Connections – Properties - Wireless Networks - Advanced*). Následne vytvorte ad-hoc sieť s názvom *BDKS_ad_hoc* ako preferovanú sieť a spustite protokol AODV na danom sieťovom rozhraní z adresára *Installable* príkazom:
`UoBWinAODV.exe /config UoBWinAODV.cfg`
Grafické rozhranie spustíte z adresára *Installable* príkazom:
`GUI.exe`

Program sa ukončuje sekvenciou Shift-q (Q).

Inštalácia OLSRD je pomerne jednoduchšia. Stačí spustiť inštalačný program a po úspešnej inštalácii reštartovať počítač. Ovládanie a konfigurovanie protokolu sa realizuje v grafickom prostredí po spustení programu.

Po inštalácii a sprevádzkovaní programov začnite s experimentovaním s uvedenými protokolmi. Vytvorte bezdrôtovú ad-hoc sieť medzi dvoma počítačmi a následne na nej spustite jeden z protokolov. Pozorujte správanie protokolu a jednotlivé správy, ktoré sú medzi počítačmi prenášané. Zamerajte sa na typ a formát týchto správ a porovnajte túto implementáciu so štandardom protokolov. Tak isto sa zamerajte aj na pozorovanie susedných uzlov.

Rozšírte svoju sieť o ďalší počítač tak, aby počítač A bol priamo v dosahu počítača B. Zároveň, aby počítač C bol v dosahu B, avšak počítače A a C nesmú byť vo vzájomnom dosahu. Na takejto topológii overte činnosť smerovacieho protokolu a experimentujte s jeho parametrami. Vyskúšajte obidva protokoly a zistené výsledky zdokumentujte v záverečnej správe z cvičenia.

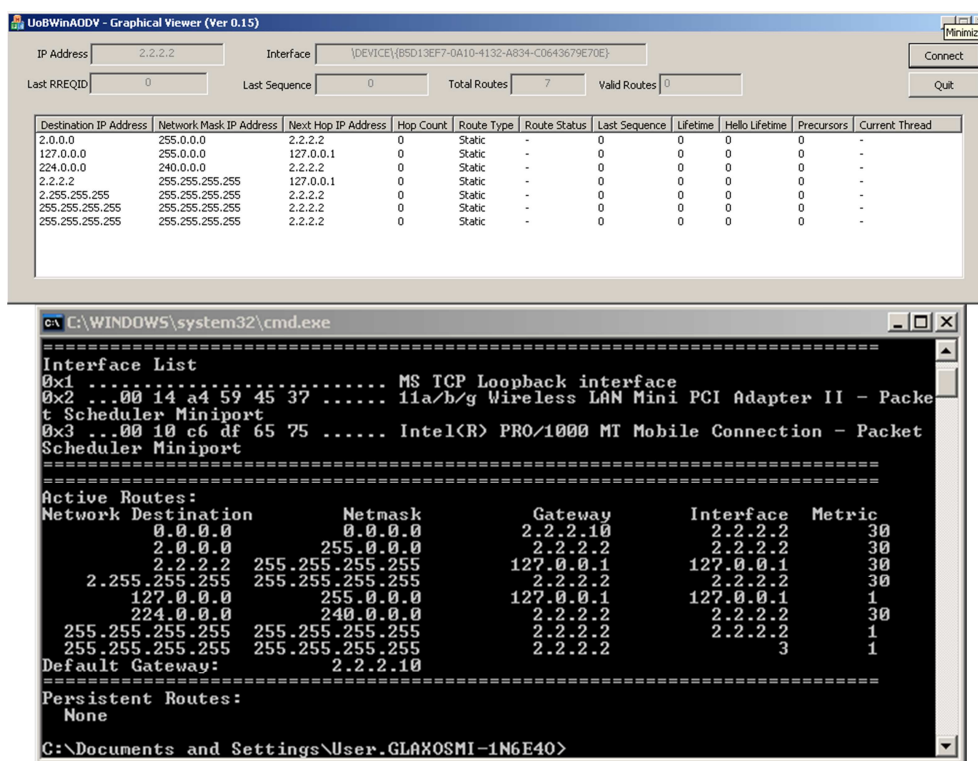
⁶ <http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&displaylang=en>

6. Výsledky skúmania

V tejto kapitole je opísané praktické experimentovanie so zvolenými smerovacími protokolmi

6.1. Experimentovanie s AODV

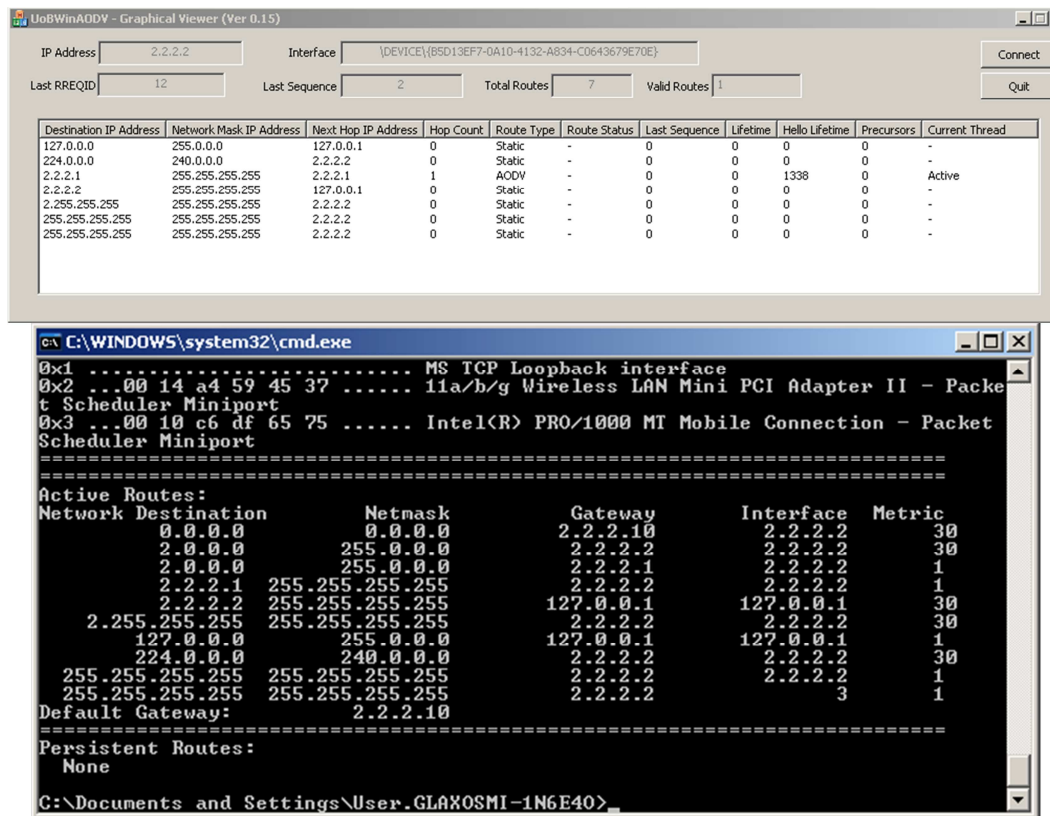
Pre praktické vyskúšanie smerovania v AODV sme spojzdnili dva stroje ktoré komunikovali týmto smerovacím algoritmom. Takáto topológia postačuje na demonštráciu základných vlastností. Po zložitej inštalácii sme spustili grafické rozhranie programu. Smerovacia tabuľka v grafickom rozhraní je podobná smerovacej tabuľke operačného systému (vid' obrázok 2).



Obrázok 2 - Smerovacie tabuľky pred začatím experimentu

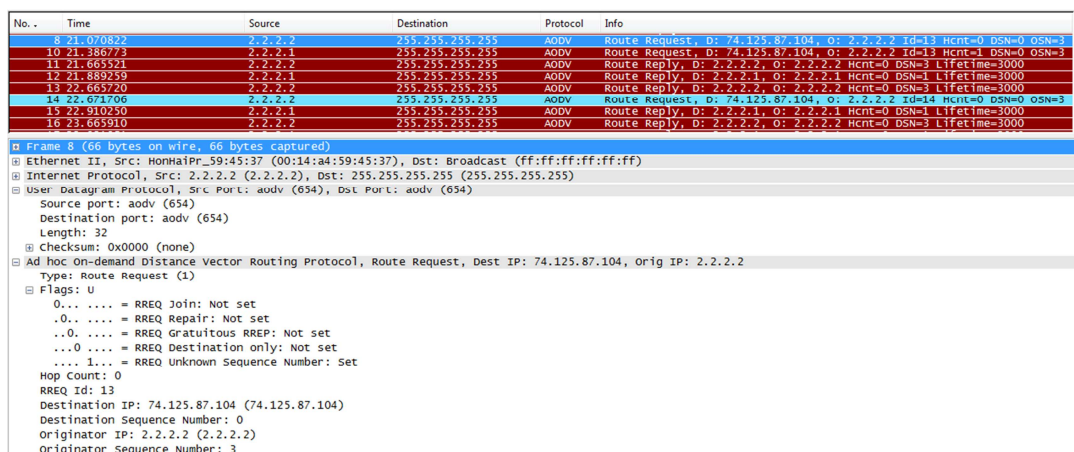
Na kontrolných výpisoch z programu Wireshark sme si overili základnú funkčnosť protokolu, a to nevedomosť o ceste k uzlu pokiaľ ju uzol nepotrebuje. Čiže pokiaľ počítač nepotrebuje komunikovať, negeneruje protokol žiadnu sieťovú prevádzku.

Keď sme sa pokúšali simulovať komunikáciu pomocou príkazu *ping* na druhý uzol, protokol začal generovať všesmerové volania. Jediný uzol v sieti mu odpovedal, a to sa prejavilo aj v smerovacej tabuľke grafického rozhrania a operačného systému.



Obrázok 3 - Smerovacie tabuľky po vyslaní požiadavky o komunikáciu

Všesmerová komunikácia zdrojového uzla bolo vidieť aj na výpisoch programu Wireshark.



Obrázok 4 - Komunikácia odchytená programom Wireshark

Do smerovacích tabuliek bola pridaná nová cesta k cieľovému uzlu. Keďže sú ad-hoc siete dynamické a rýchlo sa menia, bola k ceste pridaná krátka životnosť a po chvíli bola zmazaná.

Destination IP Address	Network Mask IP Address	Next Hop IP Address	Hop Count	Route Type	Route Status	Last Sequence	Lifetime	Hello Lifetime	Precursors	Current Thread
127.0.0.0	255.0.0.0	127.0.0.1	0	Static	-	0	0	0	0	-
224.0.0.0	240.0.0.0	2.2.2.2	0	Static	-	0	0	0	0	-
2.2.2.1	255.255.255.255	2.2.2.1	1	AODV	Invalid	1	0	0	0	Delete
2.2.2.2	255.255.255.255	127.0.0.1	0	Static	-	0	0	0	0	-
2.255.255.255	255.255.255.255	2.2.2.2	0	Static	-	0	0	0	0	-
255.255.255.255	255.255.255.255	2.2.2.2	0	Static	-	0	0	0	0	-
255.255.255.255	255.255.255.255	2.2.2.2	0	Static	-	0	0	0	0	-

Obrázok 5 - Smerovacia tabuľka v programe s odstránenou cestou

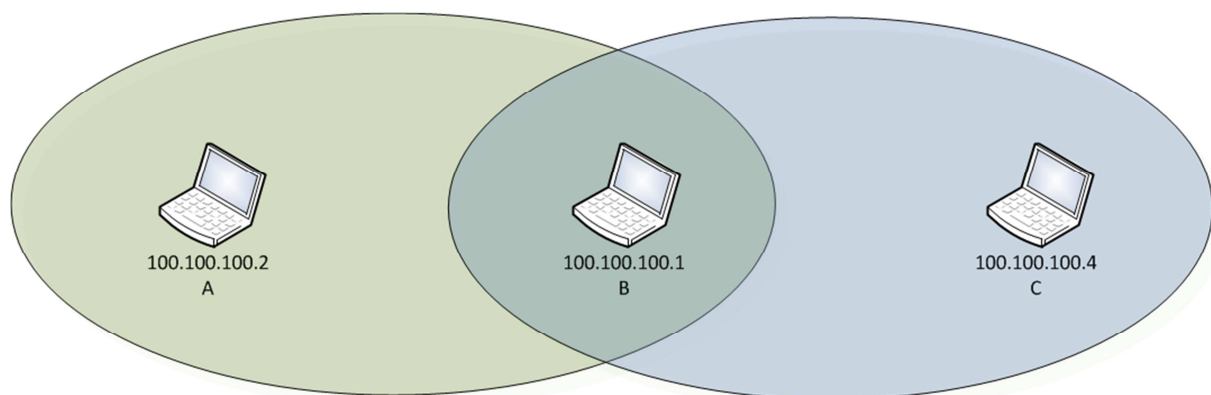
6.2.Experimentovanie s OLSR

Pre praktické vyskúšania smerovania v bezdrôtovej sieti pomocou OLSR sme si zvolili nasledujúce tri uzly. Strategické umiestnenie uzlov možno pozorovať na obrázku Obrázok 6.

Uzol A - 100.100.100.2

Uzol B - 100.100.100.1

Uzol C - 100.100.100.4



Obrázok 6 - topológia pre OLSR

Protokol OLSR je linkový stavový protokol (link-state), čiže smerovacia tabuľka k ostatným uzlom sa vytvára predtým ako je samotná cesta potrebná. Po pripojení sa do testovacej siete sme pozorovali kontrolné rámce protokolu OLSR pomocou programu Wireshark. Zachytávali sme rámce typu *Topology Change* pri zmene uzla v sieti a rámce typu HELLO pre kontrolu stavu liniek medzi uzlami (viď Obrázok 7 a Obrázok 8)

No.	Time	Source	Destination	Protocol	Info
673	180.029624	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
676	182.388772	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
680	183.821570	100.100.100.1	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
681	183.839004	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 76 Bytes
688	185.280964	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
691	186.021561	100.100.100.1	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 52 Bytes
702	186.778648	100.100.100.1	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
Frame 669: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) Ethernet II, Src: Intel_72:7d:5b (00:19:d2:72:7d:5b), Dst: Broadcast (ff:ff:ff:ff:ff:ff) Internet Protocol, Src: 100.100.100.2 (100.100.100.2), Dst: 100.255.255.255 (100.255.255.255) User Datagram Protocol, Src Port: olsr (698), Dst Port: olsr (698) Optimized Link State Routing Protocol Packet Length: 76 Packet Sequence Number: 8135 Message: TC (LQ, olsr.org) (202) Message Type: TC (LQ, olsr.org) (202) Validity Time: 368,000 (in seconds) Message: 32 Originator Address: 100.100.100.1 (100.100.100.1) TTL: 1 Hop Count: 1 Message Sequence Number: 28523 Advertised Neighbor Sequence Number (ANSN): 3 Neighbor Address: 192.168.208.1 (255/255) Neighbor Address: 100.100.100.102 (233/224) Message: HELLO (LQ, olsr.org) (201)					

Obrázok 7 - odchytený rámec TC a HELLO

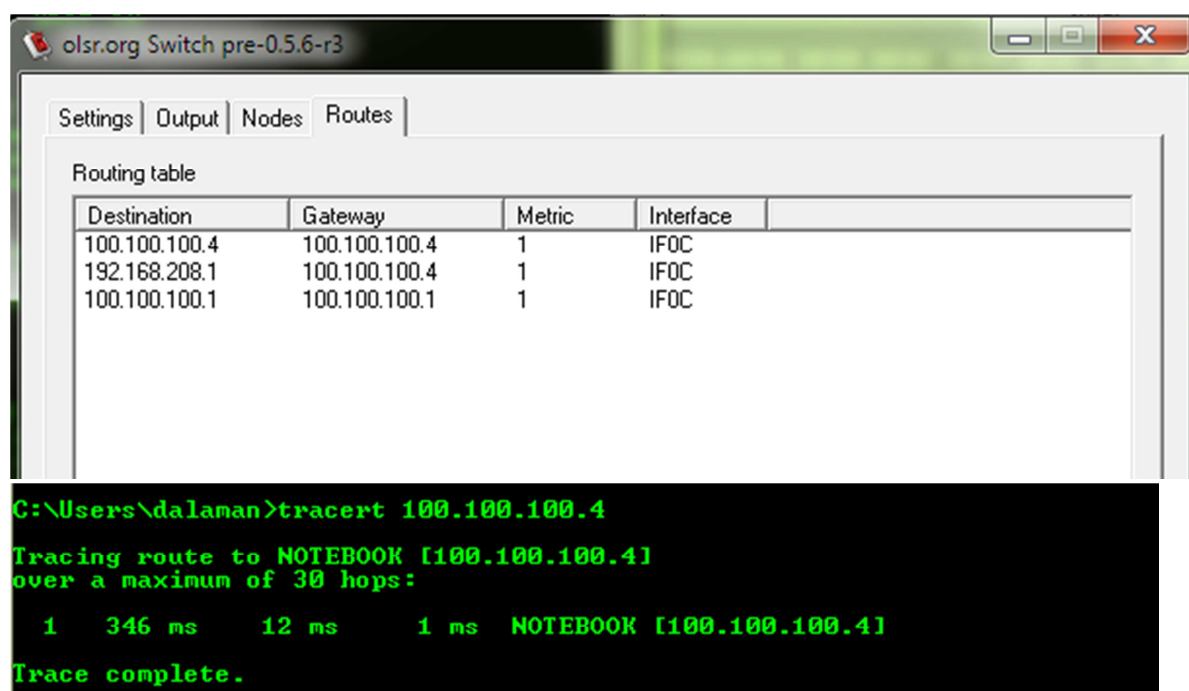
No.	Time	Source	Destination	Protocol	Info
673	180.029624	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
676	182.388772	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
680	183.821570	100.100.100.1	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
681	183.839004	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 76 Bytes
688	185.280964	100.100.100.2	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
691	186.021561	100.100.100.1	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 52 Bytes
702	186.778648	100.100.100.1	100.255.255.255	OLSR v1 OLSR (IPv4)	Packet, Length: 36 Bytes
Frame 691: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) Ethernet II, Src: Azurewav_a9:56:81 (1c:4b:d6:a9:56:81), Dst: Broadcast (ff:ff:ff:ff:ff:ff) Internet Protocol, Src: 100.100.100.1 (100.100.100.1), Dst: 100.255.255.255 (100.255.255.255) User Datagram Protocol, Src Port: olsr (698), Dst Port: olsr (698) Optimized Link State Routing Protocol Packet Length: 52 Packet Sequence Number: 8133 Message: HELLO (LQ, olsr.org) (201) Message Type: HELLO (LQ, olsr.org) (201) Validity Time: 124,000 (in seconds) Message: 48 Originator Address: 100.100.100.1 (100.100.100.1) TTL: 1 Hop Count: 0 Message Sequence Number: 28527 Hello Emission Interval: 5,000 (in seconds) Willingness to forward messages: Unknown (3) Link Type: Asymmetric Link (1) Link Type: Symmetric Link (6)					

Obrázok 8 - odchytený rámec HELLO

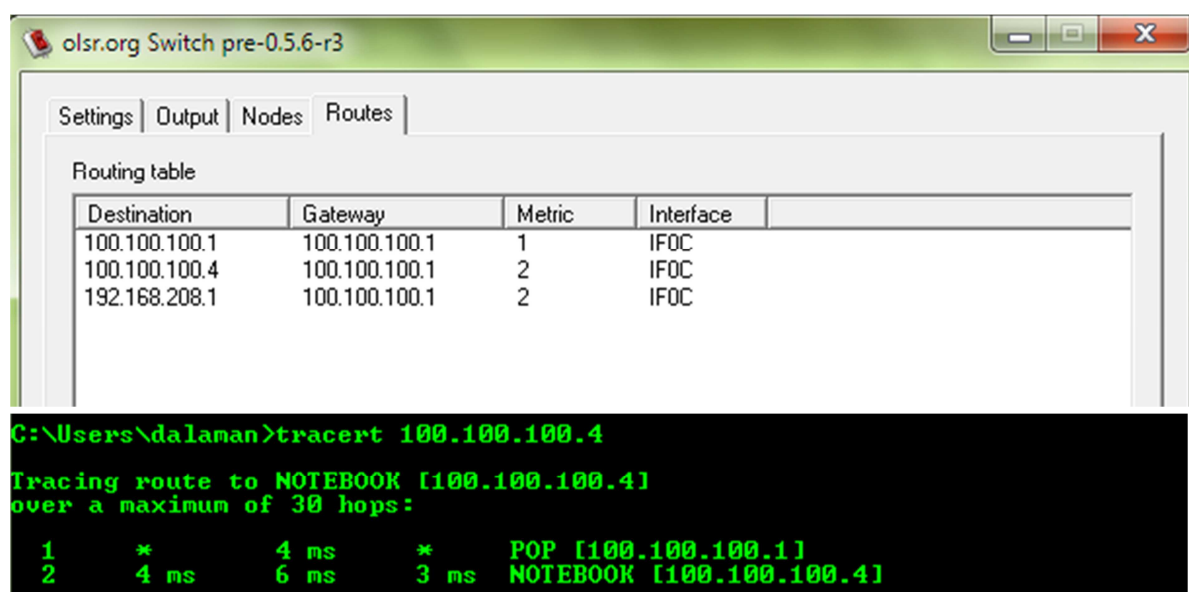
Aby sme odskúšali funkčnosť smerovania, prakticky sme odskúšali dva možné scenáre komunikácie uzlov v bezdrôtovej Ad-hoc sieti:

- Komunikácia uzlov, ktoré sú všetky navzájom viditeľné
- Komunikácia uzlov, ktoré sa vzájomne nevidia. Čiže komunikácia medzi uzlami A a C sa sprostredkováva cez uzol B.

Pre každý scenár sme sledovali smerovaciu tabuľku uzla A a výstup príkazu *tracert* k uzlu C (viď Obrázok 9 a 10).



Obrázok 9 - smerovacia tabuľka a cesta medzi vzájomne viditeľnými uzlami



Obrázok 10 - smerovacia tabuľka a cesta medzi vzájomne neviditeľnými uzlami

7. Záver

Na praktickom cvičení č. 3 sme sa oboznámili so smerovacími protokolmi bezdrôtových ad-hoc sietí. Svoje experimenty a zistenia sme vykonávali na testovacích sieťach s protokolmi AODV a OLSR, s ktorými sme sa vopred patrične oboznámili. Prakticky sme odskúšali základný rozdiel princípu fungovania oboch smerovacích protokolov.

Smerovanie medzi uzlami A a C sme otestovali vzájomným vzdialením sa uzlov do vzdialenosti, pri ktorej sa obidva uzly nevidia, no sú schopné komunikovať prostredníctvom uzla B.