Abbiamo visto il protocollo ARP. Bisogna correggere alcune cose sbagliate sulla lezione precedente. L’RFC per il protocollo ARP è il numero 826. Si può vedere che la definizione di questo protocollo prevede una serie di funzionalità che nel 1982 potevano sembrare significative ma che oggi sono obsolete: esse permettono il funzionamento senza sapere a priori quali protocolli sono utilizzati al livello 2 e 3. È stato studiato come protocollo generale in grado di funzionare con qualsiasi protocollo di rete e qualsiasi protocollo di data link. Per questo motivo la sua implementazione non sta al livello applicativo, ma sta in una via di mezzo tra il livello data-link e quello di networking. Se lo vogliamo usare come supporto a una connessione ethernet il protocollo ARP deve essere incluso in un frame ethernet: inoltre il messaggio non deve incapsulare livelli superiori (esso dovrà contenere solo l’header ethernet e un payload ARP che conterrà le informazioni necessarie per la traduzione degli indirizzi).

Non essendoci UDP di mezzo non ci sono porte di comunicazione: questa avviene tra il sistema operativo di un host a quello di un altro (senza avere applicazioni /protocolli applicativi di mezzo).

I datagrammi di tipo ARP hanno dunque una struttura che prevede una prima riga suddivisa in due campi da 2 byte ciascuno. Nei primi 16 bit è definito il tipo di hardware e negli altri 16 il tipo di protocollo (questo vuol dire che sono stati definiti dei numeri su 16 bit sulle possibili implementazioni del data link e sul protocollo di networking: al giorno d’oggi questi campi sono “fissi” a Ethernet e IPv4, cioè 1 per ethernet e 0x800 per IPv4). Dopodiché abbiamo, alla seconda riga, due byte separati in cui sono indicate le dimensioni degli indirizzi: cioè dell’indirizzo MAC e dell’Indirizzo IP (6 e 4). Seguono 16 bit che sono il numero di operazione (richiesta = 1/risposta = 2).  
Ci sono poi 6 byte (su una riga e mezzo) per l’indirizzo MAC del mittente e 4 byte per l’indirizzo IP del mittente. Dopo ci sono altri 6 byte che sono l’indirizzo MAC Target e infine 4 byte che contengono L’indirizzo IP che vogliamo tradurre (con la richiesta), che stiamo cercando (e che verranno inizializzati con la risposta).  
Per fare la richiesta ARP un host deve compilare **l’header Ethernet** (nota bene: quello di livello 2, non l’ARP) mettendo come indirizzo di destinazione l’indirizzo di broadcast (tutti 1 nel MAC) e poi inizializzare tutti i campi **dell’header ARP** tranne l’indirizzo MAC Target.   
Tutte le macchine della rete ricevono il messaggio e se un sistema operativo nota il proprio indirizzo IP nell’header ARP questo invia la risposta, inserendo nell’header ARP come sender IP e sender MAC i propri e mettendo sempre i propri IP e MAC nei target IP e MAC (sì, è ridondante), modificando invece l’header ethernet in modo da mettere come MAC di destinazione quello dell’host che aveva fatto la richiesta: in questo modo solo chi ha fatto la richiesta riceverà la risposta.

Tutte le problematiche che abbiamo visto la volta scorsa rimangono.

ARP non è definito per il funzionamento con IPv6.

Vediamo un altro protocollo, questa volta a livello applicativo, il DHCP. L’acronimo sta per Dynamic Host Configuration Protocol. Ha come scopo l’assegnazione dinamica di indirizzi IP alle macchine di una LAN (se si ha un portatile questo deve prevedere di avere indirizzi diversi in LAN diverse).  
Quando ci si connette ad una LAN si sta mandando un messaggio al server DHCP della rete, chiedendo un indirizzo IP valido. Il server DHCP risponde fornendo all’host un IP valido per la rete locale. La problematica che sorge da ciò è che se un host non ha a priori un indirizzo IP come fa a inviare un messaggio al server e ottenere una risposta? Non può inserire nulla di sensato nell’indirizzo sorgente. Inoltre, se non ci si è mai connessi a una rete, come si fa a conoscere l’indirizzo del server DHCP? Per risolvere questi problemi si utilizza la seguente tecnica.  
Innanzitutto DHCP è basato su UDP, utilizza la porta 68 per il server e la porta 67 per il client. Il motivo di tale scelta è che UDP permette di utilizzare l’indirizzo di broadcast per i messaggi (in questo modo si può inviare la richiesta su qualsiasi rete anche se non si conosce l’indirizzo preciso del server). La richiesta arriverà su tutte le macchine sulla porta 68, ma soltanto il server DHCP Non ignorerà la richiesta (visto che sarà l’unico in ascolto lì). Per l’indirizzo mittente, invece, si mette la costante 0.   
A livello di trasporto abbiamo le porte 67 e 68, a livello di networking abbiamo l’indirizzo 0 e l’indirizzo broadcast, poi abbiamo il payload DHCP.

Il server DHCP avrà una tabella di indirizzi disponibili, quindi quando riceve una richiesta preparerà una risposta, inserendo nel campo DHCP del datagramma quello assegnato (il resto del datagramma sarà uguale con la differenza che indirizzi e porte sorgenti e destinazione sono invertiti: solo che al posto di esserci l’indirizzo di broadcast ci saranno gli indirizzi IP e MAC del server DHCP). Il messaggio arriverà a destinazione (anche se l’IP destinazione è 0) sfruttando il fatto che nel messaggio di richiesta ci sarà stato l’indirizzo MAC del nuovo host, quindi mettendo quello come destinazione sarà possibile recapitare la risposta.

MAC (Destinazione, Sorgente) IP (Destinazione, Sorgente) UDP (Dest., Sorg.) DHCP



BroadCast, Client | BroadCast, 0 | 68, 67 | -



Client, Server | 0, Server | 67, 68 | IP Disponibile.

Se sulla rete locale sono stati configurati più server DHCP, tutti quanti riceveranno la richiesta: dunque un client potrebbe ricevere più di una risposta, ognuna con indirizzi IP potenzialmente diversi. Il protocollo prevede quindi che ci sia un ulteriore scambio di informazioni. C’è quindi un terzo messaggio “di accettazione”, che viene mandato in broadcast a tutti. Il messaggio conterrà l’indirizzo del server scelto dal client (cioè l’indirizzo del server che ha fornito l’IP scelto dal client) tra quelli che gli hanno risposto; in questo modo ciascuno dei server saprà se il nuovo host userà l’indirizzo che gli ha inviato lui oppure no. Pertanto, ognuno di essi saprà se tenere bloccato quell’indirizzo IP per la nuova macchina (e quindi toglierlo dalla tabella di quelli disponibili per etichettarlo come assegnato) oppure no.

C’è una fase finale: dopo l’accettazione, l’unico server che ha ricevuto responso positivo (notando che il suo indirizzo è quello usato dal client) invia al client le informazioni necessarie per l’utilizzo della rete. Tipicamente, oltre all’indirizzo IP questo messaggio conterrà la Netmask, la Default Gateway (la rete che connette il router con l’esterno) e il DNS locale (per poter fare la risoluzione di indirizzi simbolici) + una “data di scadenza” dell’indirizzo IP.

Uno potrebbe predefinire gli indirizzi MAC corrispondenti agli indirizzi IP (e quindi riservare alcuni indirizzi IP a particolari indirizzi MAC) attraverso la configurazione del server DHCP. Alcuni ISP offrono servizi a pagamento per dare a una stessa macchina sempre lo stesso indirizzo IP.

L’RFC di riferimento per il protocollo DHCP è il 2131.

Il payload DHCP è strutturato così: Il primo byte indica l’operazione del messaggio (1 = richiesta, 2 = risposta), poi abbiamo il tipo di hardware (che indica il protocollo di livello Data Link che stiamo utilizzando), poi la lunghezza dell’indirizzo MAC (cioè 6) e un quarto numero che nel caso normale deve essere 0. Alla seconda riga abbiamo un unico numero da 4 byte che è un numero casuale che identifica la transazione. Dopodiché, abbiamo 2 byte che codificano il numero di secondi che sono trascorsi dall’inizio della richiesta da parte del client. Abbiamo altri 2 byte contenenti flags.   
Poi abbiamo l’indirizzo IP del client su 4 byte, che all’inizio varrà 0. Poi abbiamo altri 4 byte per l’indirizzo IP proposto dal server e altri 4 byte per l’indirizzo IP del server (così che il client lo possa ricontattare, anche se è ridondante perché si trova già a livello di networking).   
Poi abbiamo l’indirizzo IP di un eventuale Proxy (che potrebbe intervenire tra client e server).  
Poi abbiamo 16 byte per il client hardware address e poi ci sono altri 64 byte che corrispondono al server hostname (è una stringa di massimo 63 caratteri terminati da 0) e sono un campo opzionale.  
Ci sono poi 128 byte che rappresentano il boot file\_name e infine ci sono delle Opzioni che possono esserci o non esserci.

In questo caso il meccanismo di sicurezza è costituito dal Random Trans # presente nella seconda riga (analogamente a come funziona il seq# nel protocollo TCP). A differenza di ARP, questo Non è un protocollo senza memoria, perché si tiene salvato il numero casuale che identifica la transazione.

C’è un piccolo dettaglio da aggiungere sul protocollo ARP: abbiamo visto che è su base volontaria accettare le proposte di indirizzo IP proposte da un server DHCP. Cosa succede se per caso un client (per errore o mal intenzione) prova a usare un indirizzo IP che non è stato assegnato a lui? Potrebbe succedere che il protocollo ARP quando farà richiesta per un indirizzo otterrà due o più risposte. Quindi Il protocollo ARP può essere usato per scoprire anomalie nella rete.  
Oltre alla modalità normale di funzionamento che prevede di usare ARP per tradurre un indirizzo IP in indirizzo MAC, lo si può usare per verificare che non ci siano più macchine che stanno usando lo stesso indirizzo IP: questo avviene attraverso richieste di “Probe”.  
Il protocollo ARP può essere usato come ausilio ogni volta che una nuova macchina si collega alla rete via protocollo DHCP attraverso i messaggi di Probe. La particolarità dei messaggi di Probe è che nell’indirizzo IP mittente va inserito 0 (e mettendo il nuovo IP come Target), aspettandosi che nessuno risponda: se qualcuno risponde qualcosa è andato storto nell’assegnazione dell’IP.

Anche DHCP funziona solo con IPv4 (cioè non con IPv6).

Una alternativa al protocollo ARP che funziona con IPv6 è il protocollo NDP, specificato dall’RFC 4861. NDP non è un protocollo a se stante, è bensì definito come un sottoinsieme del protocollo ICMP (versione 6), che è a sua volta un sottoinsieme del protocollo IPv6. Se ICMPv6 è sicuramente al livello 3 di rete, anche NDP dovrebbe essere al livello 3, ma è in uno strato intermedio che si affaccia anche al livello di Data Link.

NDP non fa solo il lavoro svolto da ARP, ma racchiude in sé anche una parte delle funzionalità di ICMPv4: sono state messe insieme la comunicazione tra host di una LAN (che devono tradurre IP con MAC) con quella parte di ICMP che permette ai router sulla rete internet di calcolare le tabelle di forwarding (sulla base di quali router hanno vicini).

Ci sono quindi due tipi di domande e due tipi di risposte, che permettono di dare informazioni agli altri su se stessi e sulle proprie capacità e di ricevere informazioni dagli altri. Esistono dunque le richieste di tipo router e le richieste di tipo host, con corrispondenti risposte.

Le richieste e risposte di tipo host sono praticamente equivalenti a quelle ARP, le richieste e risposte di tipo router sono analoghe a quelle ICMPv4: queste richieste sono inviate quando viene modificata la rete con l’aggiunta o rimozione di un router.

Le differenze tra versione 4 e 6 è che nella versione 6 gli indirizzi IP sono da 16 byte invece che 4 (quindi i messaggi occupano un po’ più spazio); inoltre mentre in ARP e ICMPv4 si assume che in una rete locale si usi solo un indirizzo di rete (la parte di network) e che utilizzando la netmask sia possibile generare tutti gli indirizzi presenti nella rete (cioè tutti gli host name). Nella versione 6 questo non avviene, per conoscere gli indirizzi effettivamente utilizzati sono necessari i messaggi di scambi NDP.   
Questo aggiornamento che sembra “da niente” in realtà permette di ottimizzare la compilazione delle tabelle di forwarding. A questo scopo viene definito un quinto tipo di messaggio NDP, cioè la risposta di instradamento: questo messaggio permette di specificare quali macchine fanno già parte della LAN della macchina che si vuole raggiungere, permettendo eventualmente di accorciare di un hop la strada percorsa dai datagrammi.  
Questo può essere molto utile per trovare un punto di incontro tra provider diversi.