Quello che c’è nel buffer di trasmissione, anche se è già stato inviato dall’altra parte, deve rimanervi finché non si riceve l’acknowledgement. Se un host deve mandare più datagrammi in una volta (perché la rete non supporta datagrammi di dimensioni grandi) quando il primo di questi arriva può succedere che il ricevente prima di inviare l’acknowledgement decida di aspettare (per verificare di non dover mandare informazioni a sua volta). Se però nel frattempo arriva anche il secondo messaggio, allora il ricevente può inviare un solo acknowledgement relativo alla ricezione del secondo messaggio (perché aumentando l’ack# si comprendono sia i byte del primo datagramma che del secondo).  
Se capita che il terzo messaggio non arrivi ma il quarto sì, si può comunque mandare l’acknowledgement soltanto del secondo (perché c’è un buco). Quando arriva anche il terzo quindi si può inviare un solo acknowledgement per tutti e 4.  
Viene dato l’acknowledgement dell’ultimo messaggio consecutivo arrivato, fermandosi al primo buco.   
Un ulteriore ottimizzazione è che se il ricevente deve frazionare la quantità di informazioni che deve inviare indietro, questo può inviare i datagrammi di acknowledgement con le proprie informazioni inserite nel payload (riducendo così il numero di datagrammi scambiati).

In questo modo il protocollo TCP è realizzato per favorire lo scambio di una grande quantità di informazioni in maniera bidirezionale (Piggy Backing???)

Riprendiamo il discorso della lezione precedente. Abbiamo visto l’indirizzamento IPv4.

Gli indirizzi sono specificati su 32 bit (con gli sviluppi della rete che si stanno avendo si è destinati al passaggio alla versione 6 con 128 bit).  
Abbiamo visto la ripartizione degli indirizzi in due parti usando la netmask (i bit al valore 1 nella netmask corrispondo all’indirizzo di rete nell’indirizzo).

Introduciamo il discorso degli indirizzi privati (RFC 1918).

Gli indirizzi a 32 bit vengono rappresentati come un insieme di 4 numeri separati da punti (in cui ognuno dei 4 rappresenta un byte convertito in base 10).  
Tra tutti gli indirizzi possibili sono stati individuati 3 intervalli di valori che vengono chiamati “indirizzi privati”. Essi sono fatti in questo modo:

10.x.y.z //tutti gli indirizzi che partono col valore 10 nei primi 8 bit

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

Queste sono tre reti, che vengono chiamate reti private. La prima corrisponde a una rete di classe A (perché ha 8 bit di indirizzo di rete e 24 bit di host number). La terza corrisponde a una rete di classe B (16 bit di rete e 16 di host number) e la seconda è una via di mezzo (12 bit di rete e 20 bit di host number).

La particolarità degli indirizzi privati è che se si sceglie come destinatario uno di quelli indirizzi i router non li recapitano (si rifiutano di mandare messaggi a quelle tre reti). Queste tre reti servono per avere degli indirizzi con cui fare quello che vogliamo senza fare pasticci (sono indirizzi a tutti gli effetti esclusi da internet).   
Noi abbiamo il vincolo dell’unicità degli indirizzi per far funzionare la rete (o meglio gli algoritmi di instradamento). Se noi assegnamo a un host un indirizzo privato non c’è bisogno che quell’indirizzo sia unico (perché non verrà utilizzato a livello globale). Si possono utilizzare questi indirizzi per creare una rete

locale senza creare interferenze (io posso prendermi l’indirizzo 10 ma anche il mio amico può usare l’indirizzo 10 per casa sua). Tipicamente per le reti domestiche si utilizza la rete 192.168.x.y.

Questo è un modo per avere reti locali anche di grandi dimensioni che usano lo stesso protocollo della rete globale IPv4 evitando di passare per le autorità che assegnano gli indirizzi e senza fare alcun danno alla rete geografica.  
Cosa succede però se si vuole che una rete privata possa comunicare anche con un’altra rete privata non locale? Per fare ciò si è inventato un meccanismo di traduzione degli indirizzi, il NAT, che serve per mettere in collegamento tra loro reti private diverse. Si mette nella prima rete un router diverso da solito, che effettuerà la traduzione, il NAT. Questo router avrà almeno due porte di comunicazione, quella verso l’interno avrà un indirizzo privato (del tipo 192.168.x.y) e quella verso l’esterno avrà un indirizzo pubblico (per esempio 11.11.11.11) e si potrà collegare a internet.  
Il NAT può ricevere da un host dei datagrammi destinati ad un indirizzo pubblico. Il ricevente dovrà a sua volta avere un NAT con una NIC pubblica (con, ad esempio, indirizzo 12.12.12.12) e una NIC privata (es. 172.16.x.y).

Il dispositivo NAT, dunque, cambia l’indirizzo del mittente all’interno del datagramma quando lo deve inviare in internet (sovrascrivendolo col proprio indirizzo pubblico), lasciando inalterato l’indirizzo di destinazione, così che possa essere inviato tramite l’algoritmo di instradamento.  
Il NAT del ricevente, invece, deve fare l’operazione opposta (traducendo da indirizzo pubblico a uno di quelli privati della sua rete locale) -> per far avvenire questa traduzione si effettua il PORT FORWARDING (che è l’attuale nome con cui viene chiamato il NAT ricevente) che usa le porte di destinazione per determinare l’host nella rete privata (l’unica nota da fare è che le porte sono codificate su 16 bit, dunque non ci potranno essere più di 2^16 host, anche se la rete ne permetterebbe 20 io più: se si volesse avere più host servirebbe avere più dispositivi che fanno il PORT FORWARDING). Se l’host ricevente vuole rispondere, si può fare l’operazione inversa (infatti la porta sorgente del messaggio permetterà di ritrovare l’host trasmittente, poiché è stata determinata durante la prima traduzione della NAT).

La tabella di traduzione da porta a host privato va ovviamente decisa a priori (dall’amministratore della rete), mentre quella da host privato a porta può essere decisa dinamicamente dalla NAT.

Le operazioni di NAT e PORT FORWARDING permettono quindi di usare un solo indirizzo pubblico per collegare più host alla rete internet (è come espandere la “capienza” degli indirizzi IP, detta semplicemente è quasi come avere indirizzi IP da 48 bit anziché 32). Così facendo è comunque possibile avere più server per lo stesso servizio (si possono avere due porte 80 interne su cui girano server http, ma dall’esterno una delle due deve essere visibile con un numero diverso dall’80, la porta esterna infatti può essere diversa da quella interna: In tal caso la NAT farà prima la traduzione usando la porta esterna e poi attaccherà all’indirizzo IP trovato la porta interna per determinare il tipo di servizio).

--- Una piccola porzione degli indirizzi IPv6 corrisponde a un remapping degli indirizzi IPv4 già assegnati, quindi non tutti sono utilizzabili.  
Al giorno d’oggi i cellulari funzionano con una rete che è una via di mezzo tra quella telefonica e quella tramite indirizzi IP (4G, 5G, …). ---

Abbiamo visto il livello di Trasporto e il livello di Networking. Per completare la collezione di vaghe idee che abbiamo introduciamo un protocollo a livello applicativo: il protocollo DNS. DNS = Domain Name System.  
Questo protocollo serve per rendere più facile l’utilizzo degli indirizzi: anziché dover memorizzare dei numeri possiamo permetterci di memorizzare delle stringhe di caratteri. Un esempio può essere [www.unige.it](http://www.unige.it). Questo è un indirizzo, ma è scritto in forma simbolica, di stringa di caratteri. Lo scopo del DNS è quello di effettuare una traduzione da stringa di caratteri a indirizzi IP veri (con quattro numeri tra 0 e 255 separati da punti).

La traduzione dell’indirizzo avviene in maniera abbastanza complicato, ciò deriva dal fatto che stiamo cercando di implementare un servizio che funzioni a livello globale con una grandissima quantità di indirizzi. Il DNS deve funzionare come una guida telefonica globale (a ogni numero di telefono è associato il nome di una persona). Per far funzionare l’algoritmo in maniera efficiente è stata stabilita una gerarchia, realizzata con una grande quantità di server sparsi in giro per il mondo.

Questa gerarchia è legata al modo in cui scriviamo gli indirizzi (che in realtà è una semplificazione rispetto a quella che leggiamo nell’RFC: la versione vera dovrebbe avere un punto finale, che però è opzionale e viene sempre omesso). Il punto finale rappresenterebbe la radice di un albero, dalla quale partono dei server chiamati “root” (root name server): ci sono più server (anziché uno) perché con uno solo si formerebbe un collo di bottiglia vista la grande quantità di utenti della rete internet.

Il livello successivo viene chiamato Top Level, ed è ciò che viene dopo il primo punto scandendo DA DESTRA VERSO SINISTRA (nel nostro esempio sarebbe la parola it). A questo livello si trovano delle sigle che normalmente rispecchiano le varie nazioni del mondo, oppure quello di qualche associazione (come “edu”, che indica un ente scolastico), o qualche ente governativo (.gov), o organizzazioni senza scopo di lucro (.org).

Dopo il Top Level abbiamo il livello Autoritativo (nel nostro esempio la parola “unige”). È detto autoritativo perché la responsabilità della gestione di questo tipo di indirizzi viene data a qualcuno: c’è un’organizzazione a livello mondiale che dispone i Top Level Domain e all’interno di questi Top Level vengono disposte delle regole che permettono a chiunque di attivare un livello autoritativo: ciò di solito avviene dietro il pagamento di una cifra che corrisponde a comprarsi il diritto a utilizzare il dominio internet. Per esempio, il nostro prof potrebbe pensare di comprarsi il dominio “giovannichiola.it” (a Top Level diverso corrispondono regole e costi per l’acquisto diversi).   
Dietro all’assegnazione di questi nomi c’è quindi anche un business.

La cosa interessante è che ci possono essere più livelli di tipo autoritativo, annidati uno sotto l’altro. Se quindi ti compri un dominio come unige.it, poi puoi crearti quanti sottodomini vuoi senza dover rendere conto a nessuno (gestendo tutto internamente), ad esempio potresti crearti dibris.unige.it o studenti.unige.it ecc. Ci sono service provider che comprano un dominio autoritativo e poi rivendono porzioni di tale dominio (creando sottodomini) ai propri clienti.

Quando si compra un dominio autoritativo bisogna poi mettere su un server per accettare tutte le richieste dei clients che provano a collegarsi a quell’indirizzo.

Il protocollo di comunicazione è molto semplice: prevede un solo tipo di messaggio datagramma che viene usato per mandare richieste e ottenere risposte. Il formato di questi datagrammi vede la solita rappresentazione su righe da 32 bit. I primi 16 bit sono un identificatore (un numero compreso tra 0 e 2^16-1), poi abbiamo dei flag. Alle righe successive abbiamo 4 numeri, di cui il primo è il numero di richieste (#domande), infatti si possono chiedere più cose contemporaneamente compattando tutto in un solo payload. Le domande saranno del tipo “dimmi qual è l’indirizzo IP corrispondente a questa stringa”, se stiamo chiedendo più traduzioni avremo tante domande quante ne chiediamo. Il secondo numero è il numero di risposte (#risposte), queste conterranno tipicamente l’indirizzo IP da 32 bit. Il terzo e il quarto numero contengono il numero di risposte autoritative (#risp. Autoritative) e il numero di risposte addizionali (#risp. Addizionali), quest’ultime relative a domande che non sono state fatte direttamente (il server potrebbe decidere di rispondere non solo alle domande che il client ha fatto ma anche a quelle che potrebbe fare).

Dopo questi quattro numeri ci sono altrettanti spazi dedicati alle domande e ai tre tipi di risposte.

I server contattati dai client per avere questo meccanismo di traduzione vengono tipicamente chiamati “Server Locali” (perché si presuppone che siano vicini ai client -> potrebbero essere all’interno della stessa rete locale per un servizio più efficiente). I server locali a loro volta fanno delle richieste sulla gerarchia dei server remoti per poter ottenere le informazioni per effettuare la traduzione (i server locali di per sé non hanno le informazioni per tradurre gli indirizzi, quindi fanno da tramite tra host e la gerarchia di server).

Il www è una convenzione.