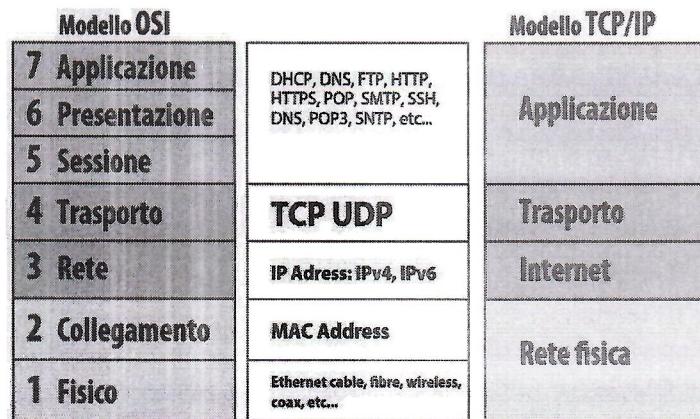


LE APPLICAZIONI DI RETE

1

IL MODELLO ISO / OSI E LE APPLICAZIONI



I modelli di rete sono dei modelli concettuali che definiscono il modo in cui le reti inviano dati dal mittente al destinatario

Il modello ISO/OSI:

- è un primo **modello di rete**, ed è composto **7 livelli o strati** (**modello a pila – stack**);
- è un modello più specializzato, ed implementa più protocolli;

Il modello TCP/IP:

- è un modello rete che semplifica il modello ISO/OSI;
- è un modello che definisce due protocolli di trasporto (**TCP + UDP**);
- rispetto al modello ISO/OSI ha **4 livelli** (5, 6 e 7 del ISO/OSI inglobati nel livello applicazione + 1 e 2 dell'ISO/OSI inglobati nel livello rete fisica detta **Network Interface o link layer**)

Differenza tra ISO / OSI e TCP / IP

ISO/OSI – è un modello teorico / fornisce livelli + servizi / risponde alla domanda: cosa devo fare? Cosa deve accadere ad ogni livello ? Descrive quindi la struttura logica di come le comunicazioni devono essere organizzate

TCP/IP - è un modello pratico / fornisce livelli + protocolli / risponde alla domanda: come deve essere fatto ? descrive quindi come le cose sono effettivamente fatte in Internet

LE APPLICAZIONI DI RETE

2

Nel modello ISO/OSI il **livello delle applicazioni** si occupa di implementare le applicazioni di rete che vengono utilizzate dall'utente finale; del resto lo scopo principale di una rete (sia in locale che in remoto) è quello di condividere dati mediante applicazioni

A tale livello INTERNET è la struttura più complessa esistente.

Il **livello applicazione** implementa i vari protocolli, tra cui:

- **SNMP**: Simple Network Management Protocol;
- **SMTP**: Simple Mail Transfer Protocol;
- **POP3**: Post Office Protocol;
- **FTP**: File Transfer Protocol;
- **HTTP**: HyperText Transfer Protocol;
- **DNS**: Domain Name System.

Questi protocolli vengono utilizzati da tutte le **applicazioni di rete generali** (posta elettronica, web, giochi multiutente via rete, streaming di video clip, telefonia via internet, videoconferenza).

In rete esistono anche le **applicazioni proprietarie**, cioè sviluppate all'interno di una organizzazione privata, (p.e. collegamento tra filiali remote o la connessione tra più sedi della stessa organizzazione)

LE APPLICAZIONI DI RETE - SOCKET

APPLICAZIONI DI RETE

3

Una applicazione di rete è costituita da un insieme di programmi che vengono eseguiti su 2 o più computer contemporaneamente.

Questi computer operano interagendo tra loro utilizzando delle risorse comuni, cioè accedendo **concorrentemente** (cioè contemporaneamente) agli archivi (database) mediante la rete che li connette.

L'applicazione di rete prende anche il nome di **applicazione distribuita**, perché viene eseguita su più elaboratori – è composta da più elementi cooperanti, posti in esecuzione su macchine diverse all'interno di una rete di calcolatori

A contrario l'**applicazione concentrata** viene eseguita su un solo elaboratore.

RICORDA

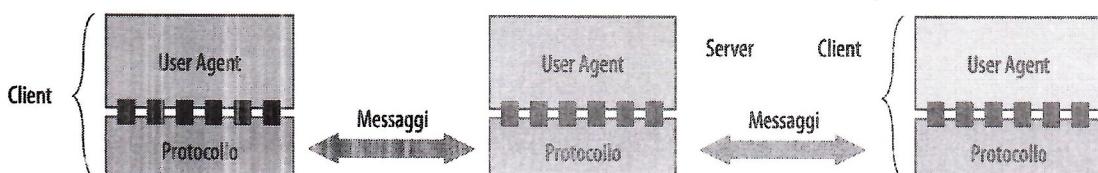
- il livello di applicazione è un livello di un modello di rete, che mette a disposizione dei protocolli mediante i quali le applicazioni possono comunicare tra host remoti
- le applicazioni sono invece degli insiemi di programmi

I **processi** hanno la necessità di scambiare messaggi con gli altri processi della stessa applicazione, sia se sono processi che appartengono alla stessa **rete locale**, sia se sono processi **remoti**.

Per comunicare tra loro i processi della stessa applicazione devono mettersi in “contatto” tramite i loro indirizzi, e utilizzare i servizi offerti dal livello di applicazione.

L'**applicazione di rete** può essere vista come composta da due parti:

1. una **user agent**, che funge da interfaccia tra l'utilizzatore dell'applicazione e gli aspetti comunicativi;
2. l'**implementazione dei protocolli** che permettono all'applicazione di integrarsi con la rete.



ESEMPIO

Possiamo individuare queste due componenti, per esempio, in un browser web:

- l'**interfaccia utente** che serve a visualizzare i documenti ricevuti, a permettere la loro navigazione e a richiedere nuovi documenti specificando la loro URL;
- il **motore del browser** che è la parte che si preoccupa di inviare le richieste ai vari server e di ricevere le risposte.

LE APPLICAZIONI DI RETE - SOCKET

4

PROTOCOLLO DI COMUNICAZIONE



Il protocollo di comunicazione è un insieme di regole di comunicazione che devono essere seguite da due interlocutori per comprendersi.

In una rete il protocollo di comunicazione stabilisce tutti gli aspetti della comunicazione:

- **fisici** (p.e. supporto fisico, meccanismo di segnalazione)
- **logici** (p.e. meccanismo di commutazione, regole di codifica dell'informazione)

Il protocollo di comunicazione è organizzato in maniera gerarchica: ogni protocollo si appoggia ai protocolli di livello più basso per fornire un servizio di migliore qualità

Ricordiamo che la pila protocollare di Internet (o pila TCP/IP) è costituita da cinque livelli, a differenza della pila ISO/OSI che ha sette livelli; l'unità di informazione che viene gestita a ogni livello assume forma e nome diverso, dato che a ogni passaggio vengono aggiunte le informazioni necessarie a ciascun protocollo per poter portare termine i propri compiti:

- al livello applicazione viene elaborato il tipo di dato messaggio;
- al livello trasporto viene elaborato il tipo di dato segmento;
- al livello rete viene elaborato il tipo di dato datagram;
- al livello collegamento (link) viene elaborato il tipo di dato frame;
- al livello fisico non corrisponde alcun tipo di dato (ma solo un segnale fisico).

5 Applicazione	Messaggio
4 Trasporto	Segmento
3 Rete	Datagram
2 Collegamento	Frame
1 Fisico	

PORTE DI COMUNICAZIONE

Normalmente un PC ha solo 1 porta Network: se più applicazioni devono utilizzare la rete, devono essere riconosciute tramite delle **PORTE LOGICHE** identificate con un numero detto **ADDRESS PORT o PORT**

Il numero di PORT è specificato su **2 byte** (vanno da 0 a 65.535):

- da 0 a 1023: numeri di porta dedicati ad applicazioni particolari (HTTP – FTP – DNS) – costituiscono le **well-known port** – devono essere conosciute dai client affinchè possano effettuare richieste tramite socket
- da 1024 a 49151: numeri di porta riservati a porte registrate – utilizzate da tutti i client per collegarsi ad un sistema remoto)
- da 49.152 a 65.535: numeri di porta liberi – possono essere assegnati dinamicamente dai processi applicativi

Il numero di porta è comunque relativo al protocollo considerato (TCP oppure UDP): un numero di porta protocollo TCP è diverso dallo stesso numero di porta protocollo UDP

LE APPLICAZIONI DI RETE - SOCKET

5

IDENTIFICAZIONE DI UN SERVIZIO MEDIANTE SOCKET

Nello scambio di messaggi tra processi in esecuzione su host diversi (**host mittente e host destinatario**) è necessario che il **processo mittente identifichi in maniera univoca il processo destinatario**.

L'identificazione deve tenere conto di 2 informazioni:

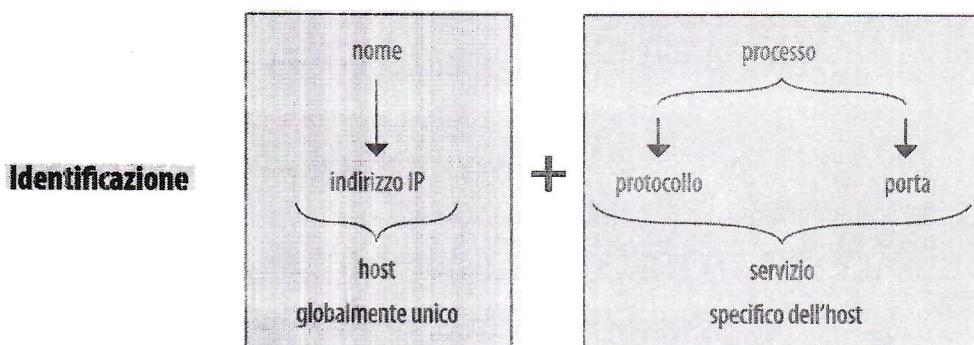
- **dell'indirizzo IP del destinatario** (che da solo non è sufficiente, in quanto identifica l'host destinatario, ma non un **processo specifico** in esecuzione, e un host può avere anche più di un processo in esecuzione contemporaneamente);
- **del processo appartenente a quell'host destinatario** - per individuare il processo, ad ogni servizio offerto da un server è assegnato un identificatore unico per l'host, chiamato **numero di porta**

L'identificazione di un servizio avviene quindi combinando:

- **l'indirizzo IP dell'host destinatario**: identifica il nodo su cui opera il processo con cui si desidera comunicare;
- **la porta logica**: identifica la porta utilizzata dal particolare processo all'interno di quel nodo (host)

Quindi, nel modello client-server:

- **il server** si registra con il software del protocollo di trasporto usando il **numero di porta (bind)**
- **il client** lo contatta usando **l'identificatore dell'host (indirizzo IP) + l'identificatore del servizio (numero di porta)**



Un numero di porta non può essere associato a più processi (server o client), ma un server può offrire più servizi, ciascuno su un'apposita porta

LE APPLICAZIONI DI RETE - SOCKET

Meccanismo dei SOCKET

6

Per richiedere un servizio al server è necessario:

- specificare l'indirizzo dell'host
- il tipo di servizio desiderato

e quindi gli identificatori del server devono essere conosciuti dal client prima di connettersi

Il meccanismo dei socket è l'identificazione univoca di un processo, che avviene conoscendo sia l'indirizzo IP dell'host presso cui è in esecuzione il processo, sia il numero di porta associato al processo in esecuzione su quell'host

Un socket è formato dalla coppia: <indirizzo IP:numero di porta>

Un processo che vuole inviare un messaggio:

- lo fa uscire dal **socket del mittente** (<indirizzo IP:numero di porta> del mittente)
- l'infrastruttura esterna lo trasporta attraverso la rete
- arriva al processo dell'host destinatario attraverso la **socket del destinatario** (<indirizzo IP:numero di porta> del destinatario)

La socket è pertanto una quadrupla dato che contiene 2 indirizzi IP (host mittente + host destinatario) e 2 numeri di porta (mittente/destinatario)

La socket ci permette di comunicare attraverso un canale virtuale all'interno di una rete, utilizzando la pila TCP/IP, ed è quindi parte integrante del protocollo.

Le **API (Application Programming Interface)** mettono a disposizione del programmatore gli strumenti necessari a codificare la connessione e l'utilizzo del protocollo di comunicazione.

Esiste una struttura chiamata **association** che consente d'identificare ogni singola connessione in modo univoco e che contiene le seguenti informazioni:

Protocollo (TCP, UDP...)	Indirizzo IP (locale)	Indirizzo IP (remoto)
	Porta (locale)	Porta (remota)

ESEMPIO

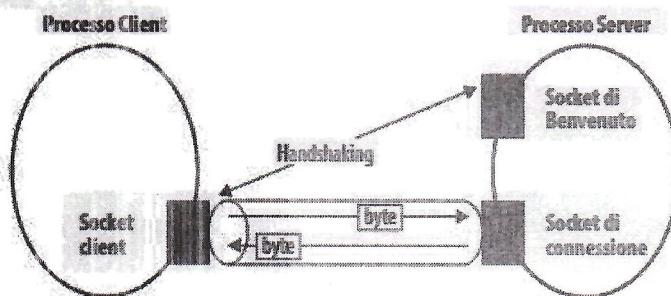
Un esempio di association è: **TCP, 192.168.1.2, 1500, 192.168.1.14, 21**. Nel dettaglio:

- **TCP** è il protocollo da utilizzare;
- **192.168.1.2** è l'indirizzo IP del PC mittente;
- **1500** è il port address del processo locale;
- **192.168.1.14** è l'indirizzo IP del PC destinatario;
- **21** è il port address del processo remoto.

LE APPLICAZIONI DI RETE - SOCKET

7

Il server offre il servizio a più client e, quindi, deve gestire la concorrenza: il socket di connessione è anche chiamato **socket di benvenuto** e alla richiesta accettata di un client il server crea dinamicamente un nuovo thread e gli assegna un socket di connessione.



Il **thread concorrente del server**, con il suo flusso di controllo autonomo operante su memoria condivisa, soddisfa la richiesta del client attraverso il socket di connessione e, contemporaneamente, si rende disponibile ad attendere sul socket di benvenuto le successive richieste di altri client.

Quando un client apre un socket, non dovrebbe utilizzare una porta nota, in quanto riservata per l'offerta di servizi.

LE APPLICAZIONI DI RETE - SOCKET

SERVIZI OFFERTI DALLO STATO DI TRASPORTO ALLE APPLICAZIONI

8

Le applicazioni richiedono al **livello trasporto** del modello di rete (presente sia in ISO/OSI che in TCP/IP) un insieme di servizi specifici e di protocolli necessari per realizzarli, protocolli che possono essere **standard** oppure **realizzati ad hoc**.

Tutti i due tipi di protocolli hanno una funzione: **TRASFERIRE I MESSAGGI DA UN PUNTO ALL'ALTRO DELLA RETE**

Ogni applicazione al livello applicazione deve scegliere quale protocollo di trasporto è migliore per le sue esigenze di trasferimento dati, tra queste:

- **trasferimento dati affidabile**
- **ampiezza di banda**
- **temporizzazione**
- **sicurezza**

TRASFERIMENTO DATI AFFIDABILE

Per trasferimento dati affidabile si intende un servizio che assicuri la trasmissione dei dati completo e corretto al 100 %, senza perdita di dati.

Il livello di trasporto mette a disposizione 2 protocolli di trasporto:

- **UDP – User Datagram Protocol**
- **TCP – Transmission Control Protocol**

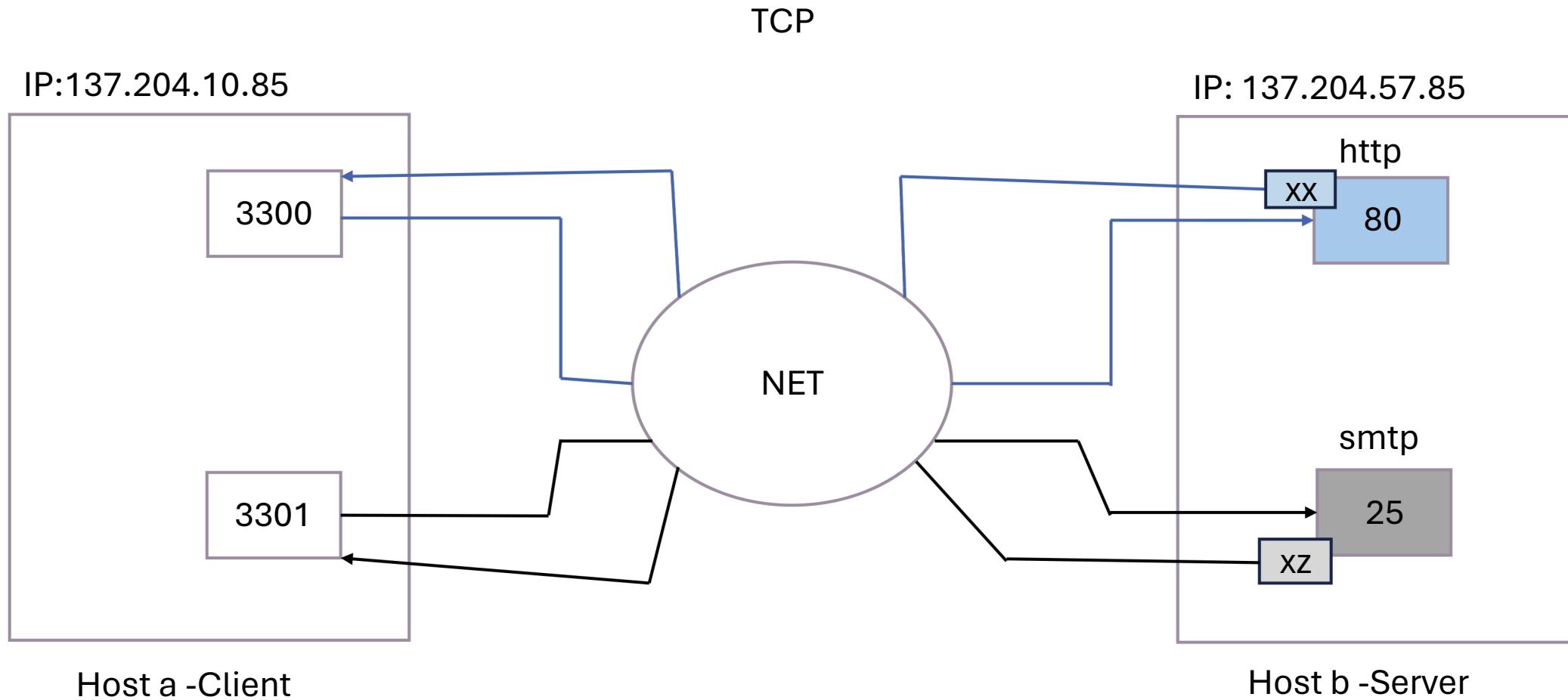
Entrambi usano IP a livello di network – infatti il nome del protocollo è TCP/IP

UDP – User Datagram Protocol:

- è un **protocollo di trasporto senza connessione (connectionless)**
- è un **protocollo NON affidabile** perché non offre il controllo di flusso, della congestione, del ritardo ed ha una banda minima
- lo si utilizza quando si accetta la possibilità della perdita di dati

TCP – Transmission Control Protocol:

- è un **protocollo di trasporto orientato alla connessione (connection oriented)**
- è un **protocollo affidabile** – offre il controllo del flusso (se il ricevente è più lento del mittente, il mittente rallenta la trasmissione per non ingolfare il ricevente), il controllo della congestione (limita il mittente se la rete è sovraccarica), non da garanzie di banda minima, dispone di un sistema per segnalare un errore al mittente (scambio messaggi di acknowledge)
- garantisce un trasporto senza errori o perdita di dati
- lo si utilizza quando **NON si accetta la possibilità di errori o di perdita di dati**

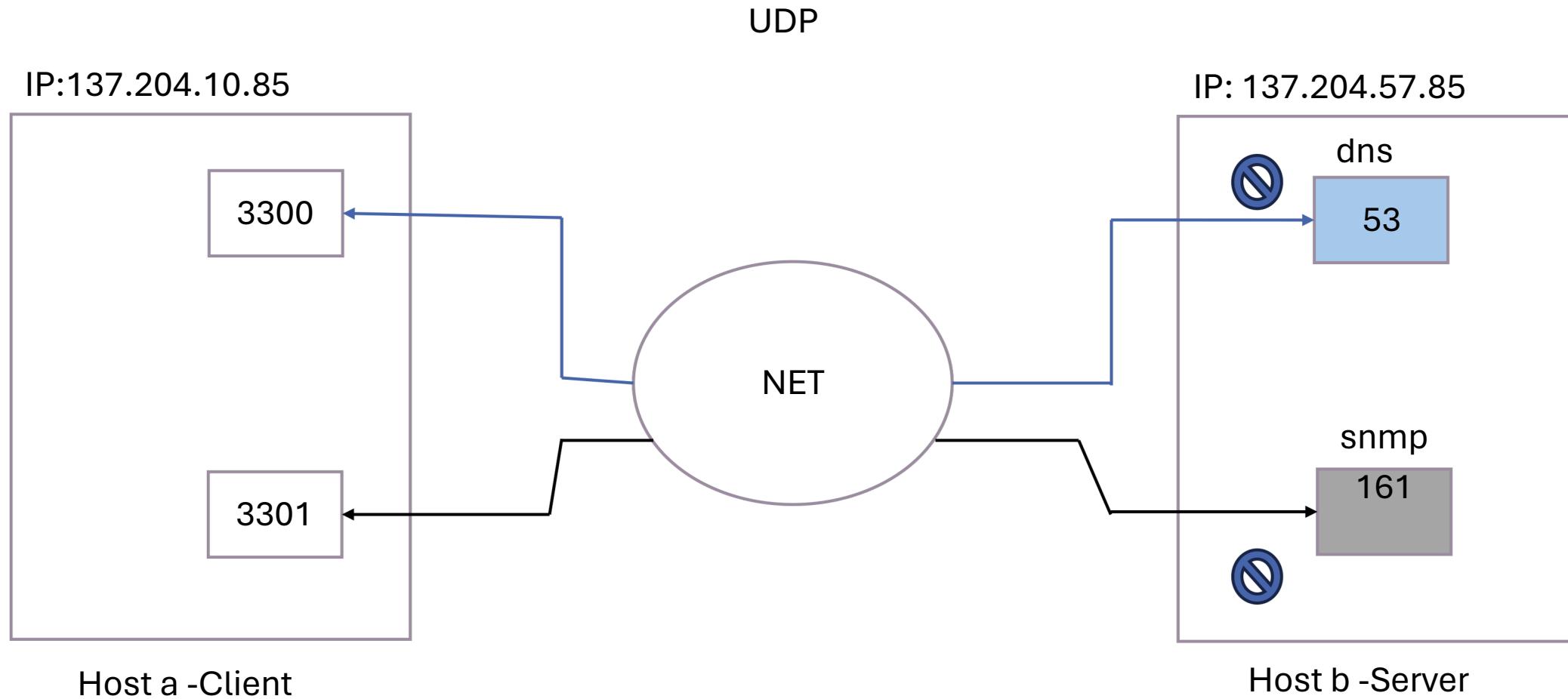


1 Socket: <137.204.10.85:3300> -<137.204.57.85:80>

2 Socket: <137.204.10.85:3301> -<137.204.57.85:25>

COMMENTI TEORICI -TCP

- Sul server è presente la socket di benvenuto che è formata dalla coppia indirizzo IP del server e numero di porta del servizio, in questo caso http.
- La coppia **<137.204.57.85:80>** corrisponde alla socket di benvenuto.
- Il server per ogni client che richiede il servizio crea una socket di connessione identificata dalla quadrupla indirizzo IP del client, porta mittente client e indirizzo IP del server, porta destinazione.
- La quadrupla **<137.204.10.85:3300> -<137.204.57.85:80>** identifica una connessione associata ad un socket dedicato.



1 Socket UDP condivisa:<137.204.57.85:53>
2 Socket UDP condivisa: <137.204.57.85:161>

COMMENTI TEORICI - UDP

- Sul server non sono presenti socket di benvenuto e socket di connessione ma una socket condivisa identificata dalla coppia - 137.204.57.85:53 o da 137.204.57.85:161
- UDP non crea infatti una socket per ogni client ma una sola gestisce tutti i client.
- Con UDP è il livello applicazione che riceve IP e porta del client, li memorizza se necessario e invia i datagrammi al client corretto.
- Nel protocollo UDP la quadrupla è associata al singolo datagramma e non identifica una connessione.