



Università degli Studi di Verona
Corso di Laurea Magistrale in Informatica

Laboratorio di Sicurezza delle Reti

Andrea Zwirner

 andrea@linkspirit.it

 @AndreaZwirner

Chi sono

- Mi interesso di sicurezza informatica dallo scorso millennio
 - “Connettere” significava “intrecciare”
 - Hacker non aveva ancora alcun significato
- Ho fondato Linkspirit, azienda che si occupa di
 - Consulenza nella progettazione sicura di software e sistemi
 - Verifiche di sicurezza su software e sistemi
 - Formazione in materia di sicurezza informatica

La sicurezza informatica

- Insieme di misure di carattere organizzativo, tecnologico e procedurale atte a garantire

- CONFIDENZIALITÀ
- INTEGRITÀ
- DISPONIBILITÀ

dell'informazione.

Come funziona

- Definizione di politiche di accesso a servizi e informazioni
 - autenticazione → chi è chi
 - autorizzazione → chi può fare cosa
- Difesa perimetrale
- Difesa interna

Sicurezza informatica offensiva

- Utilizzo di strumenti e metodologie usate dagli attaccanti reali
- Accesso a servizi ed informazioni senza possedere i permessi previsti dalle politiche di sicurezza
 - Lettura → Confidenzialità
 - Scrittura → Integrità / Disponibilità
- Ha il fine di evidenziare le vulnerabilità di sicurezza presenti nel sistema

Di cosa parliamo oggi

- Sicurezza del livello di rete
- Utilizzo del livello di rete per la rilevazione e mitigazione ai livelli più alti
- Facciamo un po' di sana programmazione di rete... in Python!

Laboratori di Sicurezza

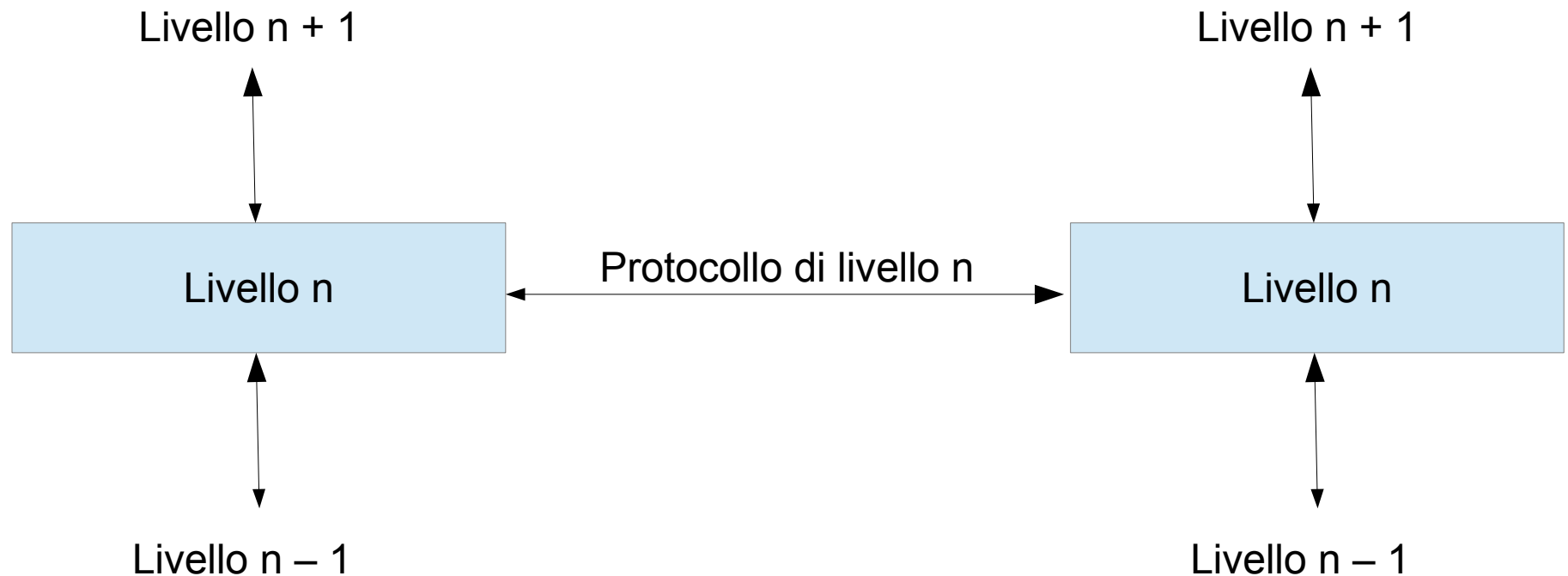
- Introduzione ai Laboratori di Sicurezza

Lo standard Open Systems Interconnection



Architettura a livelli

- Il servizio implementato dal protocollo a livello n viene fornito al livello $n + 1$



Lo standard ISO / OSI

Applicazione
Presentazione
Sessione
Trasporto
Rete
Collegamento
Fisico

Il livello Fisico

- Controlla i dispositivi hardware che compongono la rete
- Tensioni, segnali, modulazioni, codifiche, trasmissioni simultanee, etc

Applicazione
Presentazione
Sessione
Trasporto
Rete
Collegamento
Fisico

Il livello di Collegamento

- Garantisce l'affidabilità del livello fisico
- Incapsulamento e gestione degli header, **controllo degli errori** (CSMA/CD), **controllo di flusso**

Applicazione
Presentazione
Sessione
Trasporto
Rete
Collegamento
Fisico

Il livello di Rete

- Generalizzazione dei livelli inferiori rispetto ai superiori
- **Routing**, risoluzione indirizzi, gestione di frammentazioni, gestione di protocolli differenti nell'utilizzo di gateway



Il livello di Trasporto

- Gestisce la trasparenza e l'affidabilità del trasporto end-to-end
- Gestione delle **connessioni**, gestione **congestion**



Il livello di Sessione

- Gestisce la comunicazione e sincronia tra applicazioni cooperanti
- Aggiunge al trasporto le **logiche di cooperazione** del livello applicativo



Il livello di Presentazione

- Standardizza ed offre servizi di comunicazione comune al livello applicativo
- Gestisce la **sintassi dell'informazione** da trasferire (e.g. crittografia, compressione, etc)

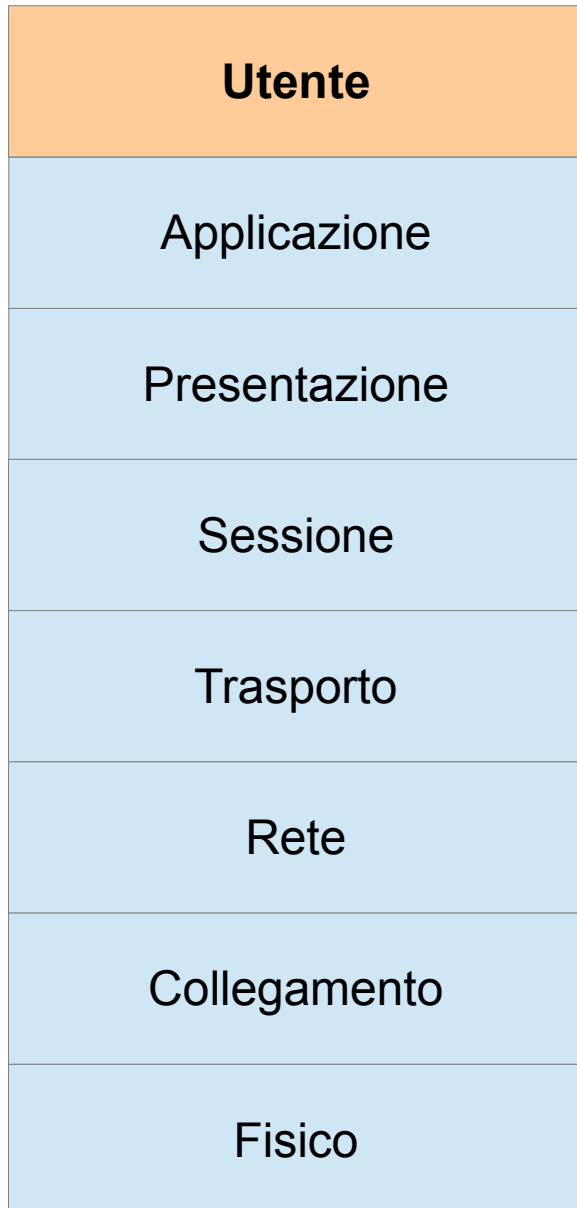
Lo standard ISO / OSI

Applicazione
Presentazione
Sessione
Trasporto
Rete
Collegamento
Fisico

Il livello Applicativo

- Gestisce l'interfaccia fra utente e la macchina
- Fornisce i **protocolli** con cui operano le **applicazioni**

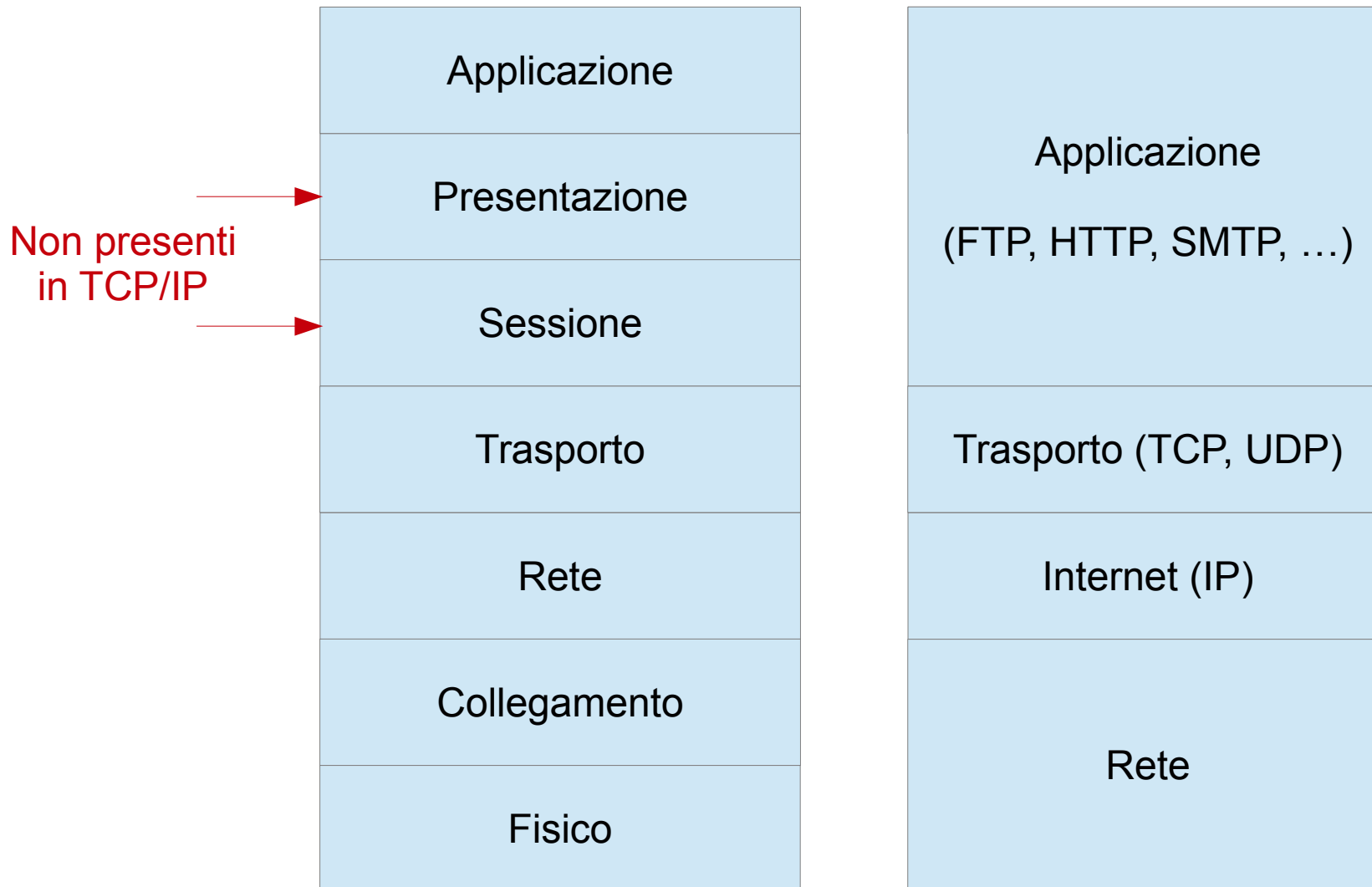
Over layer 7



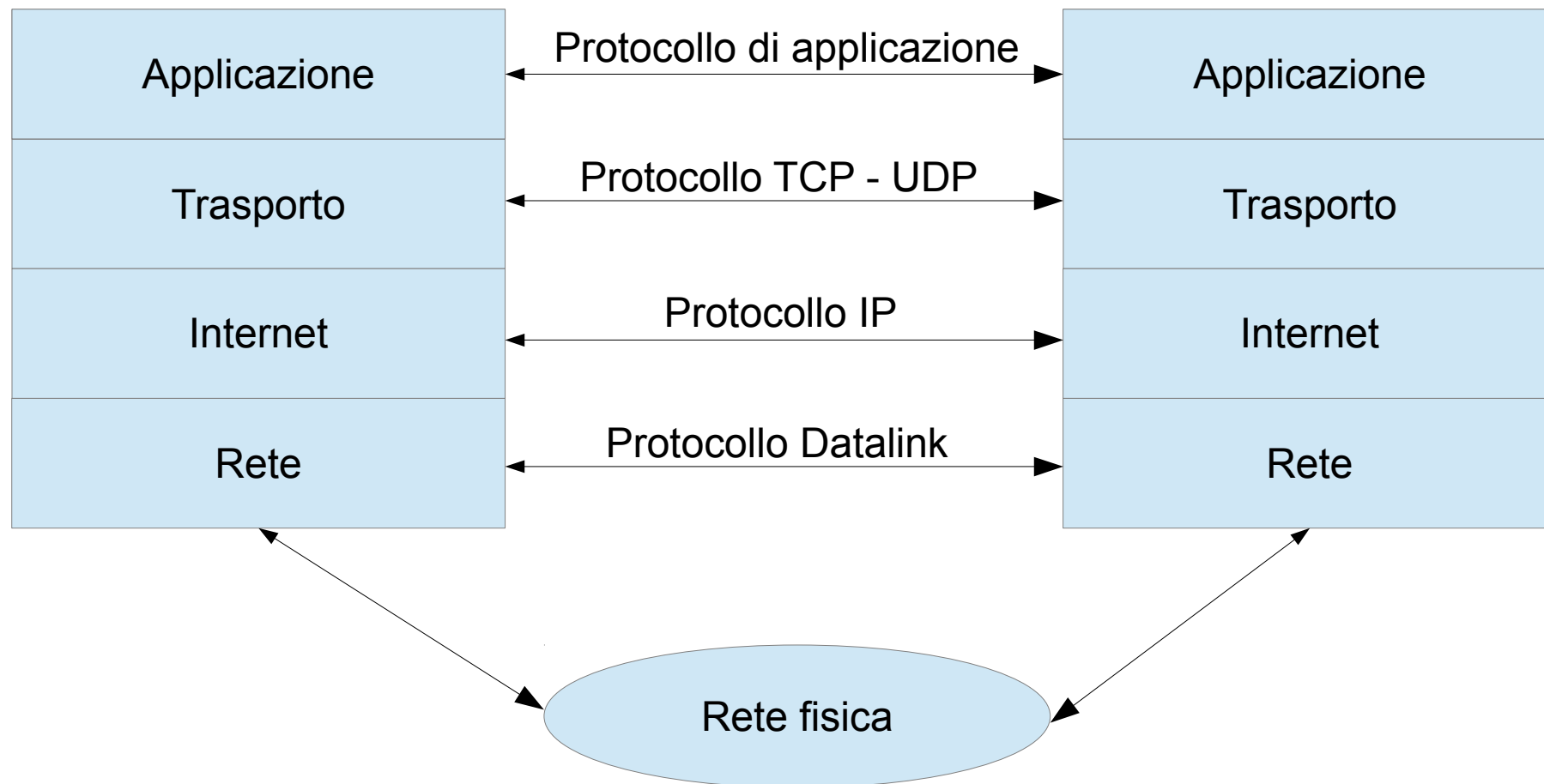
Il livello Utente

- Identifica l'utente (autenticazione) che esegue l'applicazione
- Permette di operare / discernere / filtrare in base al suo livello di autorizzazione

I livelli ISO/OSI e TCP/IP



Protocolli oggetto di indagine



Sicurezza del livello di rete

- Concezione antiquata della sicurezza di rete
- Basata su socket e sull'associazione statica protocollo – porta
 - IP + porta del mittente
 - IP + porta del destinatario
- Ha limitazioni fortissime
- E' ancora uno strumento di *network security* ampiamente diffuso (e spesso usato in esclusiva)

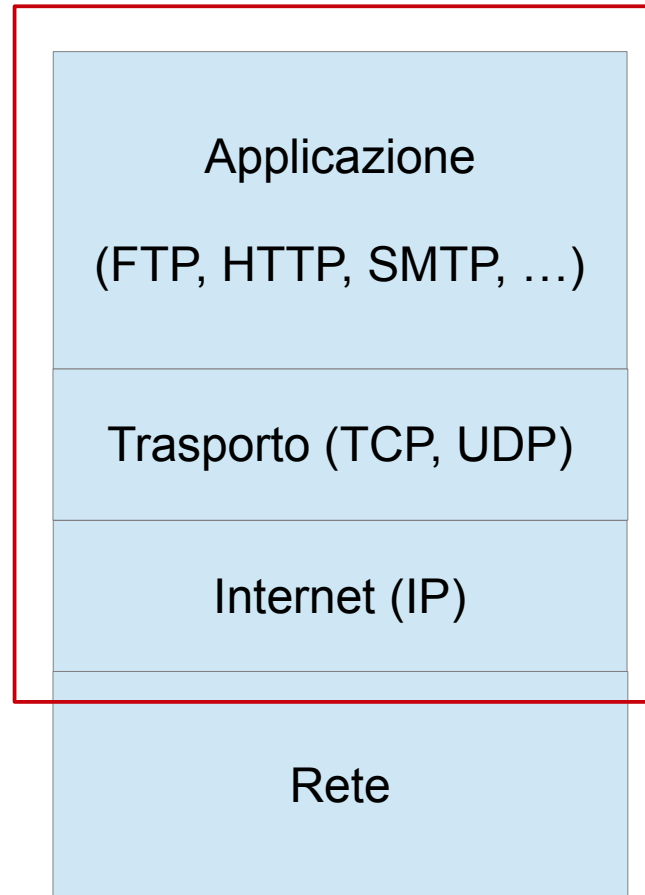
Limitazioni evidenti

- Filtrare protocolli di rete su porte non standard
- Filtrare in base all'applicazione utilizzata
- Filtrare in base all'utente che esegue l'applicazione

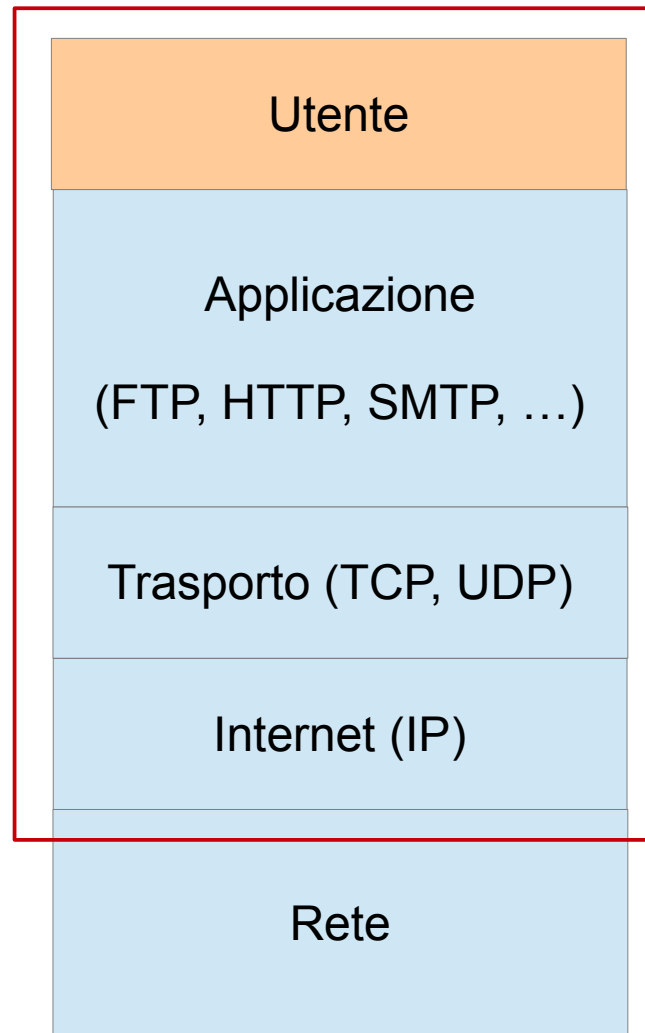
Sicurezza dal livello di rete

- L'ambito della sicurezza di rete non si limita pertanto al livello di rete
- A livello di rete transitano (incapsulate) tutte le informazioni destinate ai livelli superiori
- E' possibile mettere in campo a livello di rete controlli avanzati basati su tali informazioni
- Ciò permette di rilevare e filtrare minacce prima che si presentino al livello di destinazione

Ambito della sicurezza di rete



Ambito della sicurezza di rete



Transmission Control Protocol

- Orientato alla connessione
 - Necessario stabilire una connessione prima di avviare la trasmissione di dati
- Affidabile
 - Garantisce il recapito dei segmenti di comunicazione attraverso i meccanismi di *acknowledgements* e ritrasmissione (timeout)

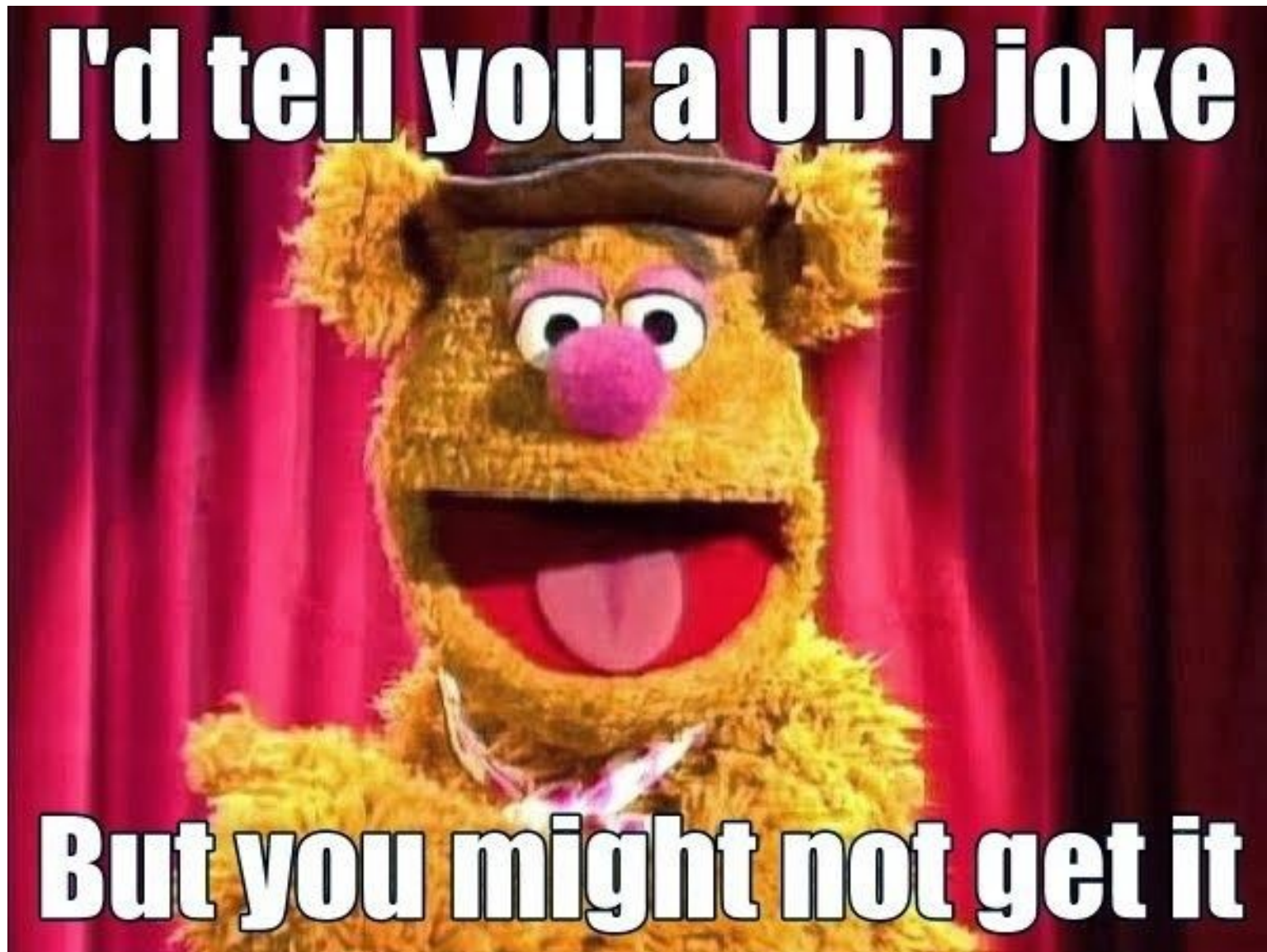
Transmission Control Protocol

- Garantisce l'elaborazione dei flussi nell'ordine di invio indipendentemente dall'ordine di arrivo a destinazione
- Implementa controllo degli errori di trasmissione mediante checksum
- Implementa controllo di flusso e di congestione mediante meccanismo della *finestra scorrevole*

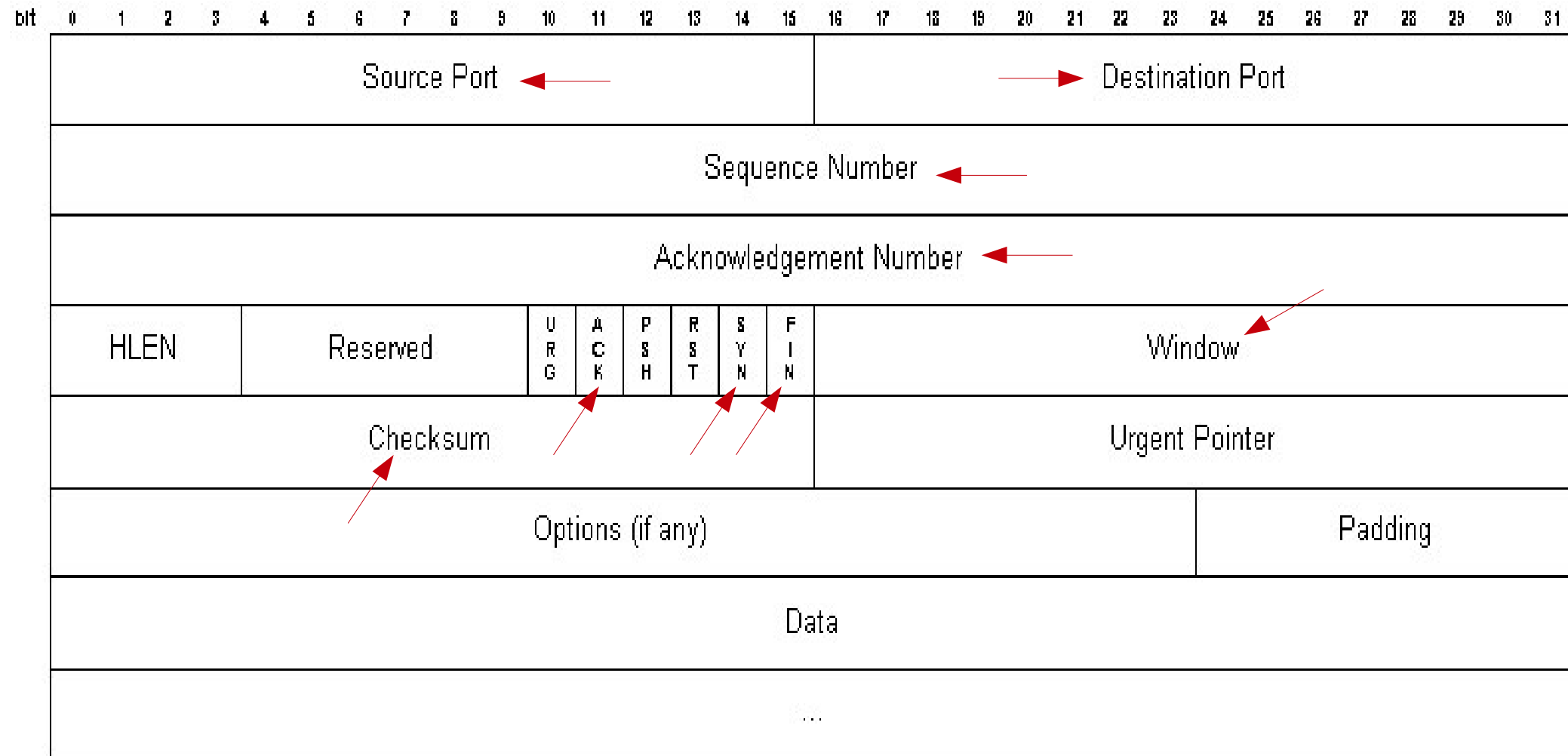
User Datagram Protocol

- Privo di connessione
- Non garantisce il recapito dei messaggi
- Invio dei soli datagrammi richiesti a livello applicativo, nessun pacchetto *di servizio* per la gestione del flusso e degli stati
 - Possono giungere a destinazione anche frammenti di messaggio
 - Non è garantita la consistenza degli stati

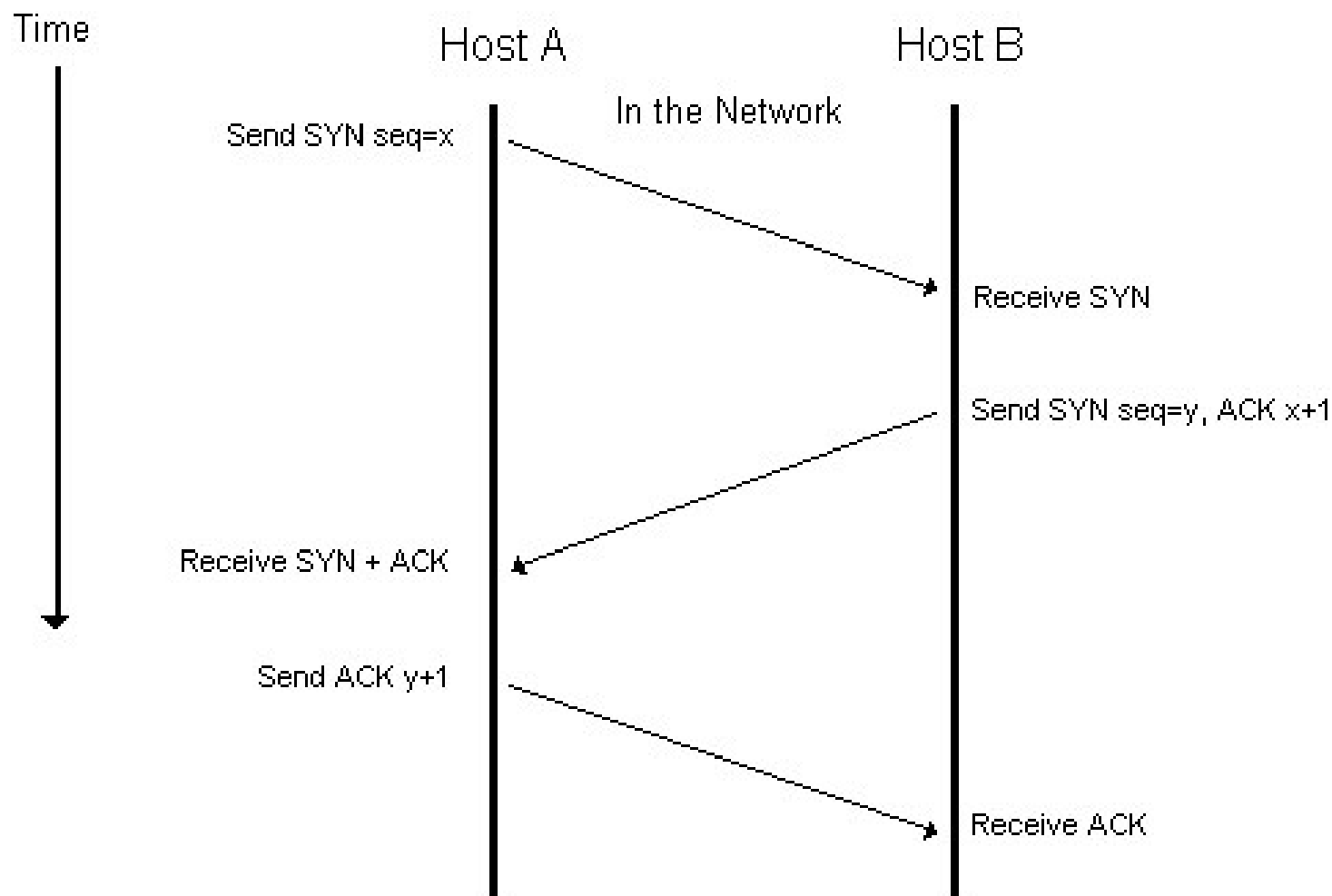
User Datagram Protocol



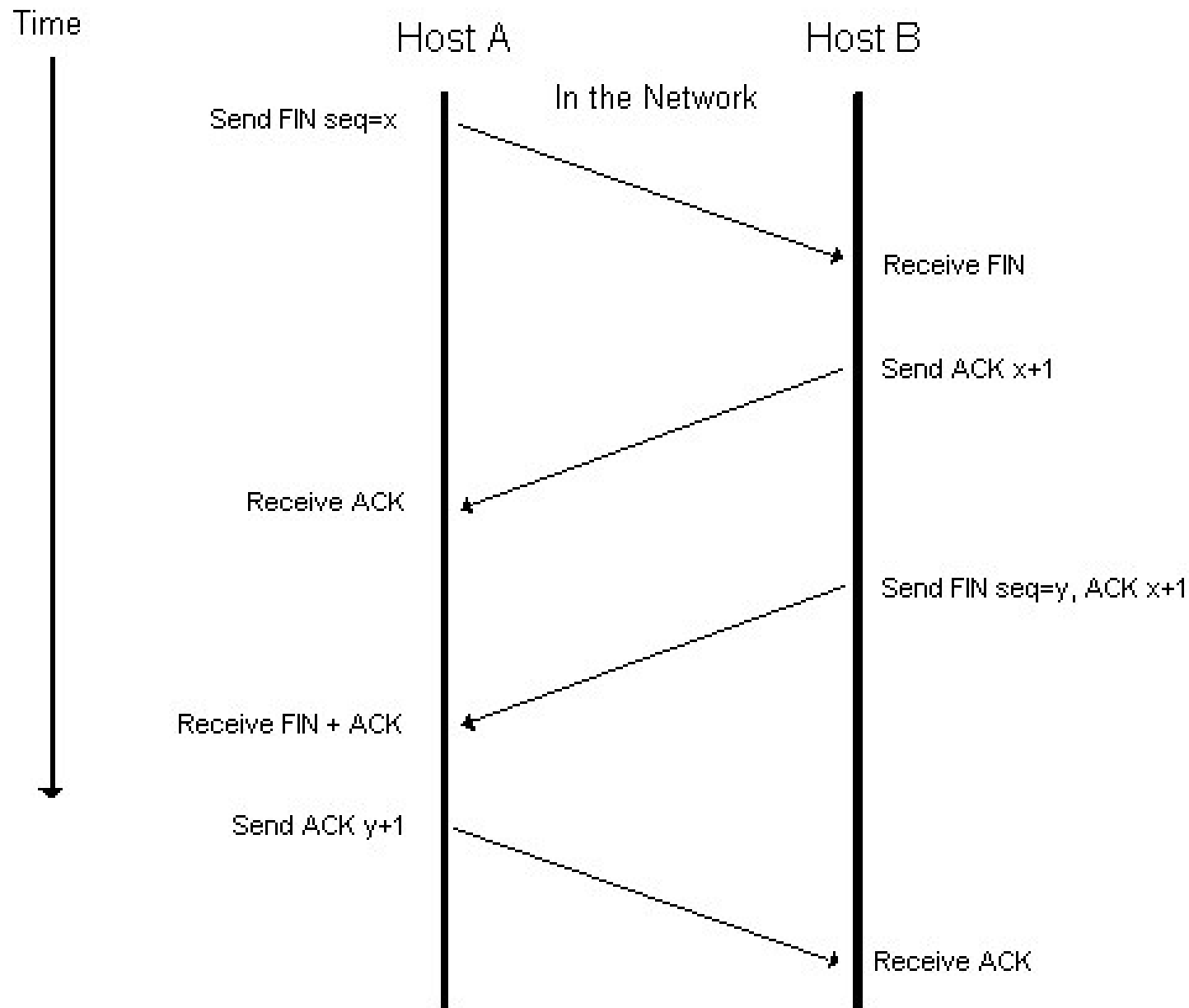
Header TCP



Apertura connessione TCP – three-way handshake



Chiusura connessione TCP – four-way handshake



Chiusura connessione TCP

- Non del tutto standard
 - Dipende dell'implementazione dello stack TCP
- Alle volte è a tre vie (disconnessione immediata, senza attesa di ACK):
 - FIN / ACK
 - FIN / ACK
 - ACK

Esempio 1

Analisi di connessione e disconnessione di connessioni TCP ed UDP

Wireshark

Netcat

Esempio 2

Analisi del traffico di una connessione TCP (contenente dati)

Wireshark

Netcat

Python-echoer

Esercizio 1

Procedura Python per la scansione delle porte di un sistema

Evidenziare le porte TCP standard (1 – 1024) aperte
su un dato sistema (BackBox)

Suggerimento: vedere `socket.connect_ex`

Rilevazione e contromisure

- Monitoraggio del numero di richieste di connessione [SYN] in un intervallo temporale
 - Su porte diverse
 - Conclude con immediato [FIN,ACK] o [RST] di eventuali connessioni stabilite
- Filtraggio degli host che superano un limite su tale valore
 - a livello di host scansionato (efficace)
 - a livello di apparati di rete
 - limita il costo computazionale
 - evita l'uso malevolo del link fra apparato ed host sotto attacco

Address Resolution Protocol

- Serve ad ottenere l'indirizzo fisico (MAC) di un host di cui si conosce l'indirizzo IP
- Il mittente invia una richiesta in broadcast (tutta la rete)
- Il destinatario la riceve e risponde con un messaggio diretto in cui include il proprio MAC address
- Il mittente salva il MAC nella propria tabella ARP e lo utilizza per l'invio della comunicazione in unicast

Address Resolution Protocol

- Dove sta il problema in tutto questo?

Address Resolution Protocol

- Dove sta il problema in tutto questo?
- Qualunque altro host in rete potrebbe rispondere alla richiesta fornendo il proprio MAC address

ARP spoofing

- Ci ha pensato per primo Alberto Ornaghi (ALoR), inventando la pratica dell'ARP spoofing o ARP poisoning
- Italians: spaghetti, pizza, mandolino, ARP spoofing



Funzionamento dell'ARP spoofing

- Alterazione malevola delle tabelle ARP delle postazioni sotto attacco
- Invio di risposte ARP (non richieste) che includono indirizzi IP non di proprietà dell'attaccante
 - Perché gli host accettano e mettono in cache risposte a richieste ARP mai effettuate?

Obiettivi dell'ARP spoofing

- Porsi nel mezzo di una comunicazione (Attacchi Man-in-the-middle o MITM)
 - Ricezione del traffico per l'host destinatario
 - Lettura ed eventuale alterazione dello stesso
 - Ritrasmissione

Esempio 3

ARP poisoning ed intercettazione del traffico di rete

Ettercap

Wireshark

Server FTP su target

Contromisure all'ARP poisoning

- Utilizzo di tabelle ARP statiche
- Filtrare le risposte ARP non richieste
- Attivare le opzioni di port security sugli switch
- Implementare il protocollo S-ARP
- Utilizzare protocolli che garantiscano l'identità dell'host remoto

Rilevazione di pratiche di ARP poisoning

- Rilevare gli host che inviano risposte ARP non richieste
- Rilevare le modifiche al MAC address associato agli IP di rete
- Per le macchine critiche (gateway, server, etc), rilevare la presenza di MAC non coincidenti nelle risposte ARP in transito
- Rilevare la presenza di indirizzi MAC assegnati a più di un IP (*) nelle tabelle ARP degli host di rete

Esercizio 2

Procedura Python per attacco a forza bruta a login FTP

Effettuare un attacco a forza bruta al login di un dato server FTP, utilizzando un dizionario con la struttura

`username:password`

Suggerimento: vedere `ftplib`

Suggerimento: `ftplib.FTPError`



Università degli Studi di Verona
Corso di Laurea Magistrale in Informatica

Laboratorio di Sicurezza delle Reti

Andrea Zwirner

 andrea@linkspirit.it

 @AndreaZwirner

Riferimenti

- Internetworking con TCP/IP, Douglas Comer – Addison Wesley ISBN 88-7192-139-9
- The Transmission Control Protocol, John Kristoff – <http://condor.depaul.edu/jkristof/technotes/tcp.html>
- Internet Security, Maurizio Cinotti – Hoepli Informatica ISBN 88-203-3045-8
- S-ARP: a Secure Address Resolution Protocol, Bruschi, Ornaghi, Rosti – Università degli Studi di Milano

- **Software**
 - Ettercap Project – <https://ettercap.github.io/ettercap/>
 - Wireshark – <https://www.wireshark.org/>