UNIVERSITÀ DEGLI STUDI
DI TRENTO
Dipartimento di Ingegneria
e Scienza dell'Informazione

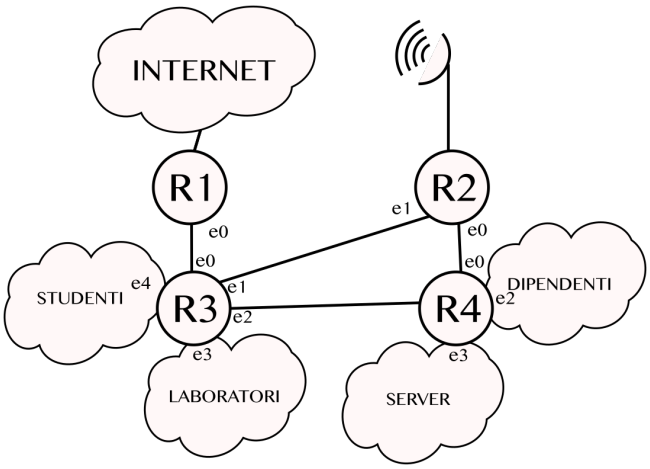
Corso di Laurea in informatica

Reti AA 2018/19

Esempio di una prova d'esame con soluzioni

Esercizio 1 (configurazione di una rete locale)

Una università deve configurare la rete di un nuovo edificio contenente aule didattiche, laboratori, sala server e uffici. La rete permette ai computer presenti nell'edificio di comunicare tra di loro e con Internet. L'Università ha acquistato un link cablato a 10 Gbps più un link di backup wireless a 1 Gbps per garantire la connettività ad Internet anche in caso di guasto del link cablato. Il link wireless, tuttavia, ha un costo di utilizzo molto alto, quindi non deve essere utilizzato quando il link cablato è attivo.



1. Supponendo che entrambi i router R1 ed R2 notifichino agli altri router la loro capacità di raggiungere Internet, come posso assicurare che il link wireless venga usato solo in caso di necessità?
2. Configurare le sottoreti STUDENTI, LABORATORI e DIPENDENTI utilizzando pool di IP privati a scelta, tenendo conto che l'edificio può ospitare circa 1000 studenti, 120 computer per i laboratori e 40 dipendenti. Si assegnino gli indirizzi IP alle interfacce dei router.
3. Configurare due sottoreti per la sala SERVER, una per le macchine che ospitano i servizi accessibili pubblicamente (20 server con i servizi web, mail, storage, ecc.) e 40 macchine con IP privato usate come cluster di calcolo. Il pool di indirizzi pubblico di proprietà dell'Università è 193.205.112.128/25. È possibile collegare la singola interfaccia fisica "e3" del router R4 alle due sottoreti dei server? Perché?
4. Si configurino le reti punto a punto fra i router assegnando gli indirizzi IP alle interfacce. Le interfacce di R1 ed R2 verso Internet sono già state configurate dal provider.
5. Per far comunicare i dispositivi presenti nelle sottoreti private (STUDENTI, LABORATORI, DIPENDENTI, SERVER DI CALCOLO) con i server che hanno indirizzo IP pubblico, è necessario usare un meccanismo particolare oppure no? Perché?

Esercizio 2 (Progetto di un Protocollo)

Si deve progettare un protocollo di livello applicativo per l'interrogazione e la configurazione di sensori remoti. Per questioni architetturali questo protocollo si deve appoggiare sul protocollo di trasporto UDP ed il server userà la porta 5555.
Per motivi legati ai dispositivi sensori è richiesto che la dimensione dei pacchetti IP non sia superiore ai 128 byte, compresi tutti gli header, inoltre comandi e risposte devono sempre essere contenuti in un singolo datagramma UDP.
Con questi vincoli si faccia un progetto di massima di un protocollo che possa supportare applicazioni che interagiscono con i sensori per leggere i valori dei sensori e per effettuare semplici operazioni come il reset di una delle periferiche di sensing, il riavvio etc.

Tracce di possibili soluzioni

Nota: La soluzione proposta per l'Es. 1 è un tra le tantissime possibili, non solo scegliendo altri gruppi di indirizzi privati, ma anche altre dimensioni delle subnet e così via.
La traccia dell'Es. 2 non è il vero e proprio progetto di un protocollo, cosa per la quale non abbiamo tutte le competenze necessarie, ma, come richiesto, un "progetto di massima" per evidenziare le competenze acquisite durante il corso, la propria conoscenza di cosa sia un protocollo e delle possibili scelte in funzione delle richieste e dei vincoli posti.

Esercizio 1

Punto 1.
È sufficiente che R2 annunci un costo del link wireless tale per cui, indipendentemente dalla sottorete dove si trova un qualsiasi host della rete, il costo di raggiungibilità di Internet attraverso R2 sia superiore a quello attraverso R1. Nel caso in oggetto, supponendo una metrica "hop count" come usato normalmente da OSPF o RIP, assegnare al link wireless costo 4 e a tutti gli altri link costo 1 è sufficiente a garantire che anche eventuali pacchetti generati da R2 transitino da R1 per raggiungere Internet se il collegamento è disponibile.

Punto 2.
Utilizziamo indirizzi privati presi dal pool 10.10.0.0/16.
Avrò bisogno di una sottorete con 1024, o meglio 2048 indirizzi per gli studenti, mentre per le altre due sono sufficienti sottoreti con 256 indirizzi. Ad esempio:
STUDENTI: 10.10.128.0/21
LABORATORI: 10.10.192.0/24
DIPENDENTI: 10.10.160.0/24

È chiaro che le due sottoreti /24 sono ridondanti in termini di numero di indirizzi e che è corretto assegnare anche sottoreti più piccole, ma è prassi, con gli indirizzi privati che sono di fatto una risorsa "rinnovabile," non assegnare sottoreti più piccole se non per ragioni specifiche.

Indirizzi delle interfacce dei Router:
R3e3: 10.10.192.1
R3e4: 10.10.128.1
R2e2: 10.10.160.1

Punto 3.
Sono sufficienti 32 indirizzi pubblici e 64 ulteriori indirizzi privati, ad esempio:
SERVER PUBBLICI: 193.205.112.160/27
SERVER PRIVATI: 10.10.161.0/26

Anche in questo caso si poteva assegnare un /24 ai server sulla sottorete privata, ma trattandosi di un cluster di calcolo, quindi probabilmente di macchine con indirizzi statici e di un sistema "chiuso" è valida anche una scelta di assegnare il numero minimo possibile di indirizzi.

Si è possibile assegnare due indirizzi IP alla stessa interfaccia, perché è necessario che un indirizzo unicast appartenga a una e una sola interfaccia, ma non il viceversa.
Non è necessario alcun accorgimento particolare, però a volte si definiscono diverse interfacce virtuali sull'interfaccia fisica a livello di SO.

Punto 4
Si assegnano sottoreti di tipo /30. Ad esempio:

R1e0 - R3e0: 10.10.0.4/30;	R1e0: 10.10.0.5,	R3e0: 10.10.0.6
R3e1 - R2e1: 10.10.0.8/30;	R3e1: 10.10.0.9,	R2e1: 10.10.0.10
R3e2 - R4e2: 10.10.0.12/30;	R3e2: 10.10.0.13,	R4e2: 10.10.0.14
R4e0 - R2e0: 10.10.0.16/30;	R4e0: 10.10.0.17,	R2e0: 10.10.0.18

Ipotizzando che le interconnessioni tra i router siano realizzati su una rete di tipo broadcast come una ethernet commutata, visto che siamo in una rete di campus è anche possibile assegnare una singola sottorete a tutte le interfacce di interconnessione dei router. In questo caso si assegnerà una /28, per avere gli 8 indirizzi unicast necessari (una / 29 ne avrebbe solo 6). Ad esempio 10.10.0.0/28, con la conseguente assegnazione dei singoli indirizzi unicast.

Punto 5
No, non serve nulla di particolare, basta che i router annuncino le sottoreti. Un IP privato non può essere instradato su Internet, ma in una Intranet si.

Esercizio 2

Data la limitazione della dimensione dei pacchetti è opportuno usare un protocollo a campi di dimensione fissa. Infatti un protocollo testuale basato su tag e campi alfanumerici, come http, renderebbe il protocollo decisamente inefficiente con il rischio aggiuntivo che i messaggi del protocollo non possano essere contenuti in un solo datagramma UDP.

Il normale header IP ha 20 byte di header, mentre UDP ne ha 8, quindi il protocollo di applicazione deve lavorare su messaggi di dimensione M = 128 - 28 = 100 byte.

Il protocollo può essere di tipo client/server, con il server del protocollo installato sui sensori, in modo che possano essere interrogati da un terminale remoto (es. PC) in qualsiasi momento.
Dovendo definire un protocollo applicativo non dobbiamo curarci degli indirizzi IP e neppure delle porte dei client, che sono effimere e vengono assegnate dal SO al momento dell'invio delle informazioni.

Il protocollo è di tipo transazionale, cioè deve supportare "transazioni" tra il client e il server. Le transazioni sono ad esempio l'interrogazione di un sensore, la trasmissione di dati "storicizzati" (es. il valore degli ultimi 10 campioni misurati) o anche il "recupero" da parte del client dei dispositivi di misura (es. temperatura, umidità, posizione GPS, accelerazione, ...) che sono installati a bordo del sensore e così via.

Come progetto di massima definiamo i campi del protocollo, almeno quelli essenziali per farlo funzionare, e alcuni esempi dei messaggi necessari a supportare le transazioni applicative. Si noti che la codifica specifica dei dispositivi di misura e altri "particolari" di questo tipo sono parte del dominio applicativo, quindi non è compito del protocollo di applicazione definire, ad esempio, il numero di bit che servono a codificare tutti i possibili sensori a bordo o che un sensore di temperatura ha il codice 0:0:0:1 piuttosto che F:F:F:0 in esadecimale se la codifica è su due byte.

Campi "obbligatori" per la gestione del protocollo, loro possibile dimensione e scopo del campo.

Protocol Version:	4 bit	versione del protocollo
Transaction Id:	12 bit	numero casuale che identifica la richiesta, per evitare "confusione" in caso di pacchetti persi o risposte ritardate; il valore viene copiato nel messaggio di risposta; i valori 0000 0000 0000 e 1111 1111 1111 sono riservati per usi speciali
Message Type:	16 bit	tipo di operazione richiesta, include anche se è una richiesta o una risposta, ad esempio in base al valore del bit più significativo
CRC	16 bit	Codice a ridondanza ciclica (es. lo stesso usato da TCP) per verificare la correttezza del messaggio. UDP può implementare il checksum in modo che controlli anche il payload, ma non è obbligatorio, quindi prevedere un ulteriore controllo a livello applicativo è sicuramente utile
Payload Length	8 bit	ne servirebbero solo 7, ma non vale la pena disallineare i byte, anzi sarebbe deleterio
Payload	0-93 byte	
Padding	0-93 byte	complemento a 73 del payload lenght, valore fisso dei byte a CC

Si è supposto, anche se l'esercizio non lo specificava in modo chiaro ed esplicito, che la dimensione dei pacchetti sia fissa a 128 byte e non solo "al massimo" di 128 byte. La scelta è arbitraria (e anche discutibile visto il testo dell'esercizio), ma consente di "discutere" l'uso di un campo di padding. Ovviamente l'esercizio è altrettanto corretto assumendo un pacchetto di dimensione variabile senza padding, nel qual caso anche il campo Payload Length è superfluo, perché il payload termina alla fine del segmento UDP, che è ben definito.

Esempio di possibili codici di comando

Il codidi di comando sono su 16 bit, per semplicità assumiamo che siano "obbligatori" solamente i codici che hanno i due byte più significativi a zero, mentre tutti quelli con i due byte più significativi diversi da zero solo riservati per possibili espansioni proprietarie del protocollo.
La tabella che segue riporta quindi solo i valori dei due byte meno significativi del campo Message Type, essendo gli altri due sempre a zero.
Quelli che seguono sono solamente alcuni possibili comandi e risposte, ma il protocollo può prevederne molti altri. In genere non è possibile essere esaustivi, ma è utile mettere un sottoinsieme significativo che consenta un funzionamento "ragionevole" di una applicazione.

0000 0000	riservato
1111 1111	riservato
0xxx xxxx	richieste
1xxx xxxx	risposte, la risposta a una richiesta può avere valore NR+128, ma non necessariamente come vedremo; le risposte contengono sempre lo stesso Tr Id delle richieste/comandi a cui si riferiscono
0000 0001	reboot del dispositivo
1000 0001	comando di reboot ricevuto, inizio il reboot
1000 0010	reboot effettuato, NB perché questo comando funzioni deve esistere una EPROM su cui viene scritto il Tr Id e altri parametri
0000 0011	reset di un particolare sensore a bordo; il payload contiene il sensore di cui si chiede il reset
1000 0011	reset effettuato; il payload contiene l'esito del reset (es. sensore funzionante, oppure non risponde, oppure ...)
0001 0000	richiesta elenco sensori a bordo, nel payload ci saranno parametri di applicazione, ad esempio per il formato dei dati
1001 0000	riposta con l'elenco dei sensori, il payload contiene l'elenco stesso; caso particolare che si può analizzare se c'è tempo è quello in cui si prevede che l'elenco dei sensori possa superare i 73 byte
0001 0001	richiesta ultima misura di un singolo sensore; il payload contiene il sensore di cui si chiede la misura
1001 0001	risposta ultima misura di un singolo sensore; il payload contiene il valore, oppure la coppia sensore/valore, oppure un codice di errore se la misura o il sensore non sono disponibili
0001 0010	richiesta ultime n misure di un singolo sensore; il payload contiene il sensore di cui si chiede la misura e il valore di n
1001 0010	risposta con le ultime n misure di un singolo sensore; il payload contiene gli n valori, oppure n coppie sensore/valore, oppure l'id sensore seguito da n valori, oppure un codice di errore se la misura o il sensore non sono disponibili; questa risposta facilmente non "sta" su un singolo pacchetto se n è grande, si può lasciare il compito all'applicazione di definire come estendere le risposte su più pacchetti, oppure definire una ulteriore risposta di continuazione
1111 1110	comando non supportato; il payload è vuoto
1111 1101	errore di sistema; il payload può contenere dei codici di errore