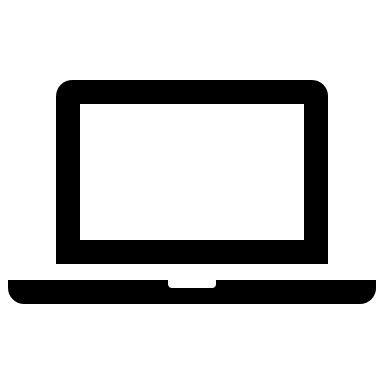
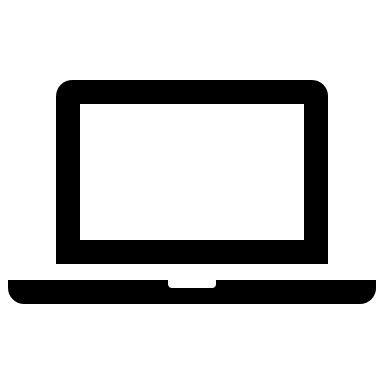
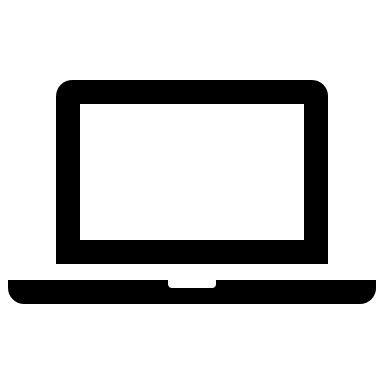
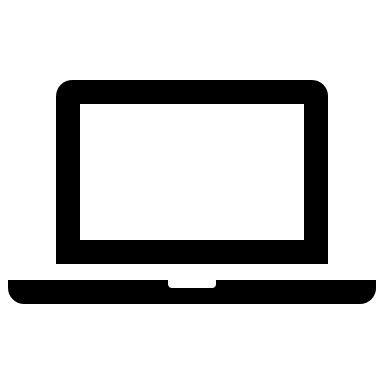
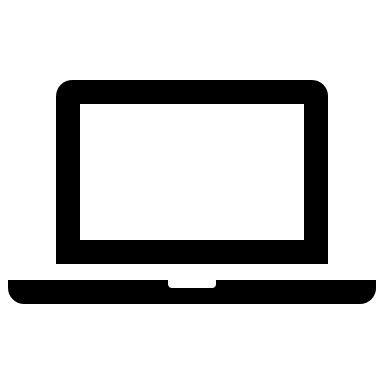
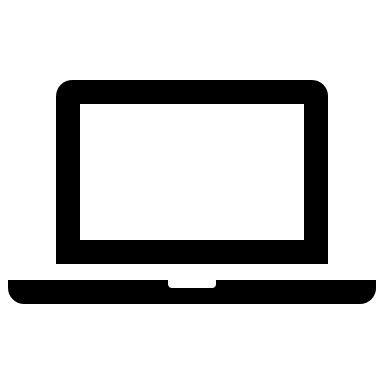
Consideriamo una rete ad anello con 6 switch collegati ciascuno a un host, come in figura



1

1

1

1

1

1

2

2

2

2

2

2

3

3

3

3

3

3

**Funzionamento normale**

In condizioni di funzionamento normali vogliamo che i pacchetti vengano instradati in senso orario, dunque quando un host manda il pacchetto al proprio switch tramite la porta 3, lo switch deve inviare il pacchetto sulla porta 1 a prescindere dalla destinazione (assumiamo che le porte siano uguali per disposizione in tutti gli switch). Ogni switch deve avere come regola base la seguente:

**se un pacchetto arriva dalla porta 2 (o 1), controlla la destinazione, se è destinato all’host collegato inoltra il pacchetto sulla porta 3, altrimenti inoltra sulla porta 1 (o 2).**

I possibili approcci sono due: **reattivo** e **proattivo**. Nel primo caso ogni switch all’arrivo di un pacchetto da una porta per la prima volta chiede al controller cosa fare e il controller installa una regola, che seguirà il criterio della regola base sopra, per ottenere il comportamento voluto.

Nel caso proattivo all’avvio del sistema il controller installa negli switch la regola sopra più eventualmente una regola per la gestione dei guasti.

**Gestione dei guasti**

Possiamo contemplare tre possibili scenari di guasto:

1. Rottura di un collegamento
2. Uno switch fuori servizio
3. Due collegamenti non adiacenti fuori servizio (ignoriamo?)

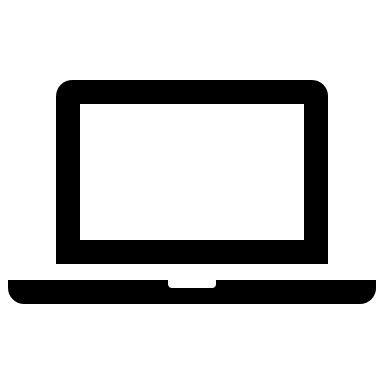
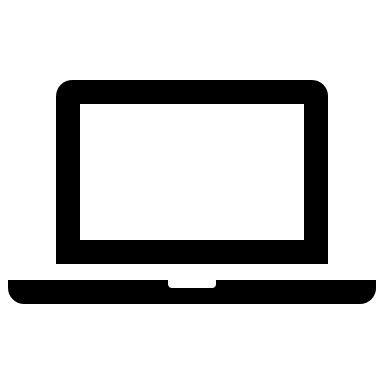
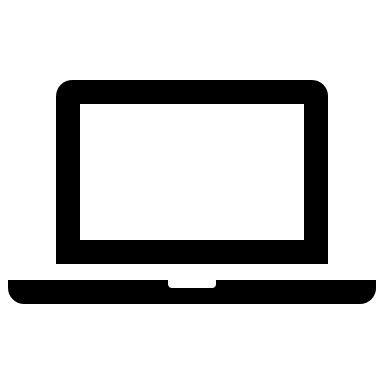
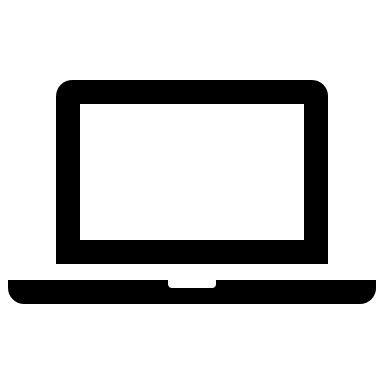
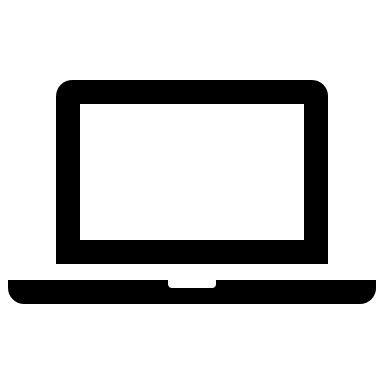
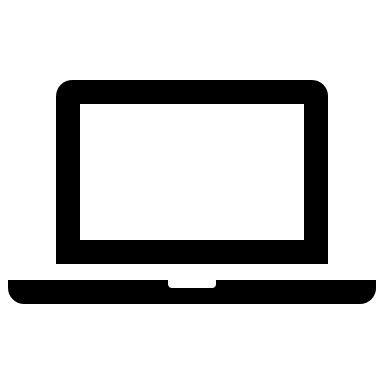
A prescindere dallo scenario di guasto possiamo usare sempre due approcci per gestirlo:

1. **Reattivo** 🡪 gli switch non sono istruiti su come gestire possibili guasti, se una porta risulta fuori servizio lo switch chiede al controller cosa fare e questo installa una nuova regola solo sullo switch che vede il guasto
2. **Proattivo** 🡪 all’avvio del sistema il controller installa oltre alle regole di instradamento anche quelle di gestione dei guasti questa volta su tutti gli switch

In ogni caso ci interessa trovare il modello di gestione che richieda il minor dispendio di risorse.

**Scenario 1: rottura di un collegamento**





1

1

1

1

1

1

2

2

2

2

2

2

3

3

3

3

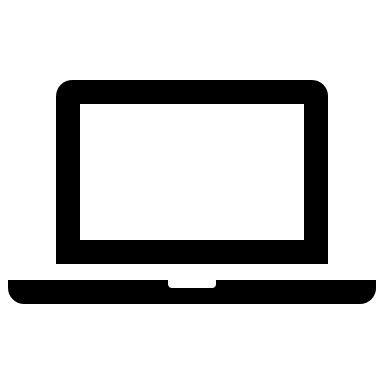
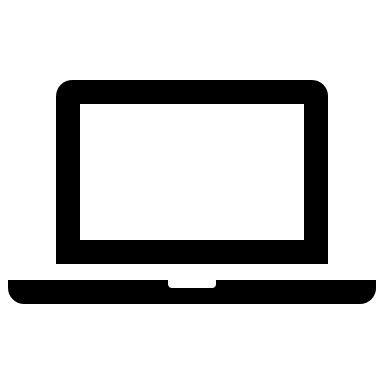
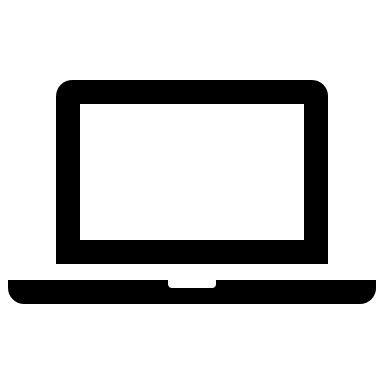
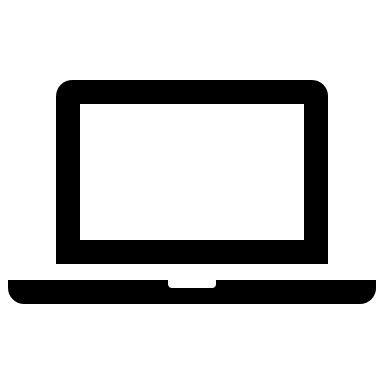
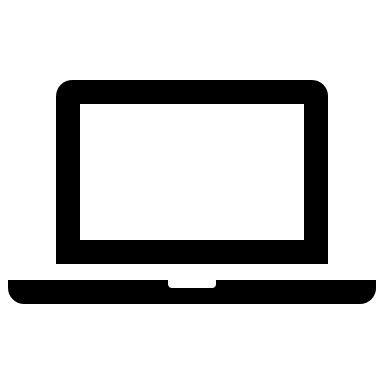
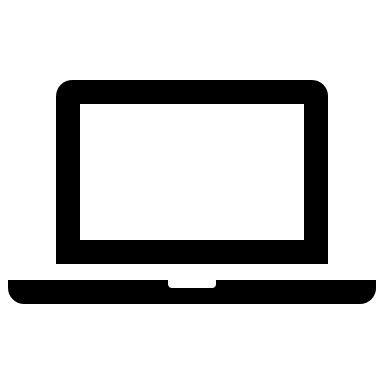
3

3

In questo scenario gli host sono ancora tutti raggiungibili, dunque non sarà mai necessario fare il drop di un pacchetto. Lo switch cerchiato, ovvero quello adiacente al collegamento guasto in senso orario, sarà il primo ad accorgersi del problema. Lo switch all’altro capo del collegamento non deve accorgersene poiché il pacchetto nel caso peggiore verrà consegnato proprio al suo host tramite la porta 3, quindi non cercherà mai di accedere alla porta 2. Gli altri switch mantengono un comportamento normale seguendo le regole istallate (reattivamente o proattivamente).

**Scenario 2: switch fuori servizio**





1

1

1

1

1

1

2

2

2

2

2

2

3

3

3

3

3

3

Uno switch fuori servizio causa due problematiche:

1. Ci sono due collegamenti inutilizzabili
2. Ci sono uno o più host che diventano irraggiungibili 🡺 dovrò droppare alcuni pacchetti

Anche in questo caso il primo host ad accorgersi del guasto è quello adiacente in senso orario, la procedura da seguire dipende dalla conoscenza o meno dell’host irraggiungibile:

* Se accortomi del guasto sono in grado di determinare gli host non raggiungibili, installo sugli switch la regola di droppare tutti i pacchetti con quelle destinazioni
* Se non sono in grado di farlo, uso lo stesso approccio dello scenario 1, ma in più istruisco l’altro switch di bordo con la regola di buttare tutti i pacchetti che non siano destinati alla porta 3, perché sicuramente indirizzati a host non raggiungibili

**Implementazione STP**

Si può implementare il protocollo STP (spanning tree protocol) (vedi Ryubook), che si riadatta autonomamente in caso di guasto, per la gestione dei pacchetti arp.

**Regole da installare sugli switch**

if (packet in from port 3) {send to port 1};

if (packet in from port 2) {

check dest address

if (dest address == host) {send to port 3}

else if (dest address != host && port 1 is up) {send to port 1}

else {send back to port 2}

}

if (packet in from port 1) {

check dest address

if (dest address == host) {send to port 3}

else if (dest address != host && port 2 is up) {send to port 2}

else {drop packet}

}