



Gestión de Prestaciones y Optimización

Gestión de Redes

Prestaciones: Objetivo y Beneficios

2

Asegura que los recursos en la red permanecen accesibles para que los usuarios los utilicen eficientemente.

□ Beneficios:

- ▣ Mejor nivel de servicio (**Calidad**):
 - Ayuda a prevenir congestión
 - Aumentando el rendimiento
 - Mejora la disponibilidad
- ▣ Ayuda a planificar la futura capacidad necesaria de la red (**Optimización**).

- **Monitorización de datos:** utilización de dispositivos y enlaces para determinar el comportamiento y nivel de servicio de la red.
- Poner **umbrales** de utilización que avisan antes de llegar a valores máximos.
- **Generación de tráfico** que permite observar el comportamiento de la red bajo cierto nivel de carga controlada.
- **Análisis de datos:** permite verificar si se sobrepasa la capacidad de la red y planificar la futura capacidad.
- **Simulación** de red permite determinar cómo la red puede ser modificada para optimizarla.

Aspectos a considerar

4

- ¿Qué medir?
 - ▣ Elementos, indicadores
- ¿Cómo obtener medidas?
- ¿Qué hacer con las medidas?

¿Qué elementos pueden ser medidos?

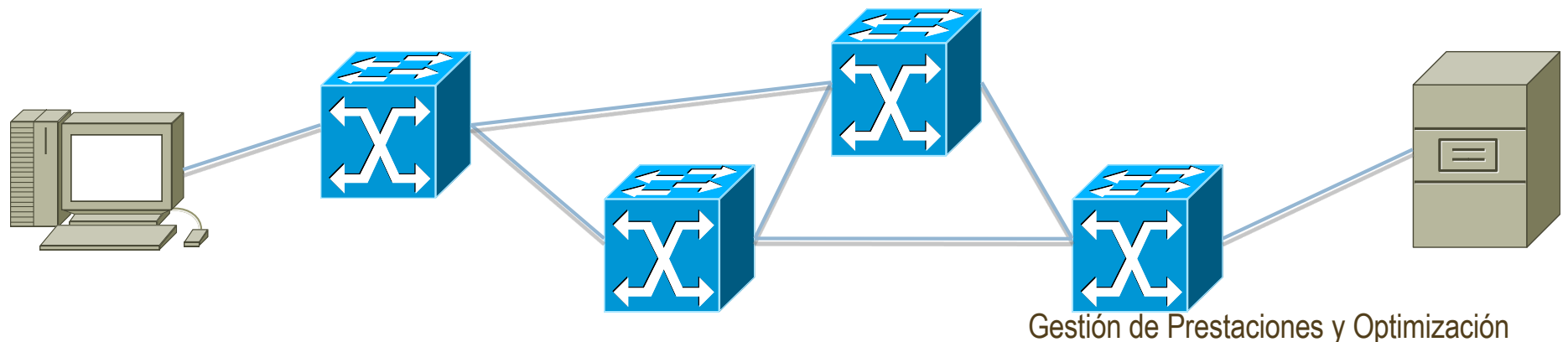
5

- Enlaces de red
 - ▣ **Utilización** del enlace en el tiempo
 - Detectar anomalías (gestión de fallos)
 - Planificación de capacidad
- Elementos de red (router, conmutador)
 - ▣ **Paquetes procesados**
 - ▣ Algunos dispositivos incorporan herramientas específicas de análisis de rendimiento.
- Servicios de red y Aplicaciones (dns, VPN, ldap, smtpd, http, base de datos....)
 - ▣ Medir las prestaciones desde el punto de vista del cliente
 - ▣ Medir el **tiempo de respuesta** tanto para peticiones internas como externas

¿Cómo medir?

6

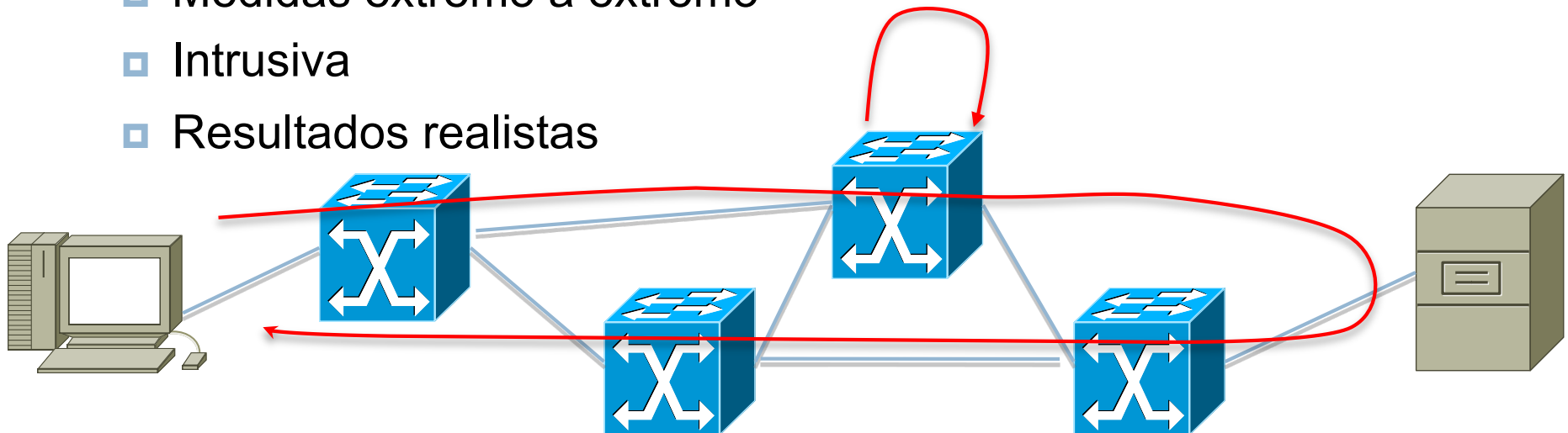
- Es sencillo tomar **medidas** de indicadores **locales** en cada elemento de un enlace.
 - ▣ Ejemplo: retardo, paquetes perdidos, throughput... en un router concreto.
- El usuario final está interesado en medidas de indicadores **extremo a extremo**
 - ▣ Ejemplo: retardo en obtener respuesta de un servidor web.
- No es inmediato deducir medidas extremo a extremo a partir de medidas locales.



Aproximaciones de medidas en la red

7

- **Observación Pasiva:** obtiene medidas sobre el rendimiento de una forma no intrusiva.
 - ▣ Medidas locales.
- **Pruebas activas:** tests que generan tráfico para tomar medidas, afectando al tráfico existente.
 - ▣ Medidas extremo a extremo
 - ▣ Intrusiva
 - ▣ Resultados realistas



¿Qué parámetros medir? -- Indicadores

8

- **Disponibilidad:** tiempo que la red está disponible y tiempo de recuperación ante problemas.
- **Tiempo de respuesta:** tiempo que tarda el sistema en reaccionar a una entrada determinada.
- **Fiabilidad:** porcentaje de tiempo sin errores en la transmisión y entrega de información.
- **Velocidad efectiva o Throughput:** eventos/seg.
- **Utilización:** porcentaje de la capacidad que está siendo utilizada
- **Latencia:** retardo de un paquete al atravesar la red (ms).
- **Pérdida de paquetes:** porcentaje de paquetes que la red pierden (%).
- **Jitter:** varianza en el retardo de paquetes (importante para tráfico en tiempo real como es el de voz).

Disponibilidad

9

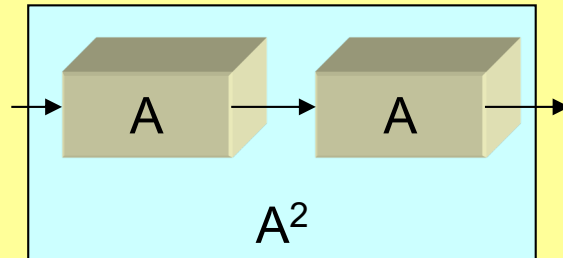
- Porcentaje de tiempo que un sistema de red, componente o aplicación está disponible para el usuario.
- Medida de disponibilidad A:

$$A = \frac{MTBF}{MTBF + MTTR}$$

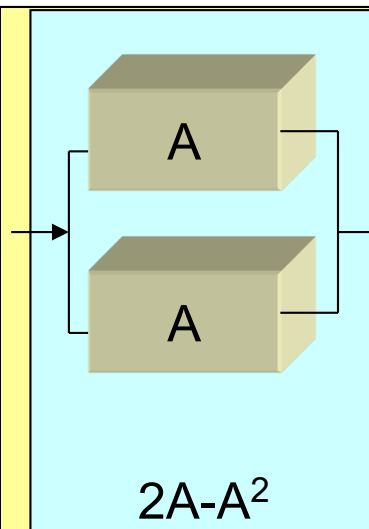
 - ▣ MTBF: tiempo medio entre fallos.
 - ▣ MTTR: tiempo medio de reparación.
- La disponibilidad de un sistema depende de la disponibilidad de sus componentes.
- El fallo de un componente puede ocasionar:
 - ▣ Que el sistema no funcione: serie.
 - ▣ Que el sistema siga funcionando igual: paralelo (redundancia).
 - ▣ Que el sistema funcione con una reducción de la capacidad: el cálculo se hace más complejo al tener en cuenta la carga.

Disponibilidad, ejemplos

10



Sistema disponible si ambos
elementos lo están: $A * A = A^2$



Sistema indisponible
si ambos elementos
lo están: $(1-A)^2$

Entonces: $1 - (1-A)^2 = 2A - A^2$

Ejercicio

11

- Dos enlaces en paralelo con disponibilidad A.
 - ▣ En periodos de carga normal (40% de peticiones de servicio) cualquiera de los enlaces puede absorber todo el tráfico (100%).
 - ▣ En periodos de picos de carga un solo enlace absorbe el 80% del tráfico.
- Calcule la disponibilidad total

Solución

12

- Disponibilidad Funcional para una situación de carga:

$$Af = C_1 * Pr_1 + C_2 * Pr_2$$

- C_i es la capacidad con i enlaces.
- Probabilidad de que funcionen los dos enlaces:

$$Pr_2 = A^2$$

- Probabilidad de que funcione un enlace:

$$Pr_1 = A(1-A) + (1-A)A = 2A - 2A^2$$

- En situaciones de carga normal: $C_1 = C_2 = 1$;

$$Af_n = 1 * Pr_1 + 1 * Pr_2$$

- En situaciones de pico de carga: $C_1 = 0.8$, $C_2 = 1$;

$$Af_p = 0.8 * Pr_1 + 1 * Pr_2$$

- Disponibilidad total:

$$Af = 0.4 * Af_n + 0.6 * Af_p$$

Tiempo de respuesta

13

- Tiempo que tarda el sistema en reaccionar a una entrada determinada.
- Un mejor tiempo de respuesta aumenta la productividad pero tiene un coste que hay que evaluar:
 - ▣ Mayores requisitos hardware: cpu, memoria...
 - ▣ Competencia entre procesos: penaliza a otros procesos.
- Medidas de tiempo de respuesta se pueden utilizar para:
 - ▣ Corregir problemas y dimensionar adecuadamente la red.
 - ▣ Identificar cuellos de botella (estudio más detallado).

- Porcentaje de tiempo sin errores en la transmisión y entrega de información.
- Es un parámetro dependiente de la tecnología usada y por lo tanto no controlable por el usuario.
- La tasa de errores que deben ser corregidos es un indicador de:
 - ▣ Fallos intermitentes en la línea.
 - ▣ Existencia de fuentes de ruido o interferencias.

Ejemplos problemas de fiabilidad en TCP

15

- ❑ Indicador de fiabilidad de la red (rechazos de conexión):
 - ❑ tcpAttemptFails: nº de intentos fallidos para abrir una conexión.
 - ❑ tcpEstabResets: nº de resets para una conexión establecida.
- ❑ tcpRetransSegs: número de segmentos retransmitidos.
- ❑ tcpInErrs: número de paquetes recibidos con error.
- ❑ tcpOutRsts: nº de intentos de hacer reset en una conexión indica un problema que afecta a la fiabilidad.

Ejercicio.

16

- Los paquetes descartados y los errores en un interfaz indican **problemas en el interfaz** o el medio.
- Ocasionan pérdida de rendimiento.
- Medir el porcentaje de descartes y de errores en la transmisión y recepción de un interfaz de comunicación.

Solución: descartes

17

- Porcentaje de paquetes descartados en un interfaz:

$\% \text{ in_Discards} = \text{ifInDiscards} / (\text{ifInUcastPkt} + \text{ifInNUcastPkts})$

$\% \text{ out_Discards} = \text{ifOutDiscards} / (\text{ifOutUcastPkt} + \text{ifOutNUcastPkts})$

- ifInDiscards: paquetes descartados de entrada.
- ifOutDiscards: paquetes descartados de salida.
- ifIn(Out)UcastPkts: paquetes unicast
- ifIn(Out)NUcastPkts: paquetes no unicast
- Si **ifInDiscard** crece al mismo ritmo que **ifInUnknownProtos**. Los paquetes se descartan por protocolo desconocido. El problema no es del interfaz.

- Porcentaje de errores en un interfaz:

$\% \text{ input errors} = \text{ifInError} / (\text{ifInUcastPkts} + \text{ifInNUcastPkts})$

$\% \text{ output errors} = \text{ifOutError} / (\text{ifOutUcastPkt} + \text{ifOutNUcastPkts})$

- ▣ ifIn(Out)Errors: paquetes con error
- ▣ ifIn(Out)UcastPkts: paquetes unicast
- ▣ ifIn(Out)NUcastPkts: paquetes no unicast

Las medidas se pueden tomar en un periodo de tiempo (j-i):

$$\text{valor} = \text{valor}_j - \text{valor}_i$$

¿Cómo procesar varios OIDs?

19

- Podemos usar librerías: por ejemplo en python

<http://pysnmp.sourceforge.net/>

```
from pysnmp.entity.rfc3413.oneliner import cmdgen

cmdGen = cmdgen.CommandGenerator()

errorIndication, errorStatus, errorIndex, varBinds = cmdGen.getCmd(
    cmdgen.CommunityData('public'),
    cmdgen.UdpTransportTarget(('demo.snmplabs.com', 161)),
    cmdgen.MibVariable('SNMPv2-MIB', 'sysName', 0)
)

# Check for errors and print out results
if errorIndication:
    print(errorIndication)
else:
    if errorStatus:
        print('%s at %s' % (
            errorStatus.prettyPrint(),
            errorIndex and varBinds[int(errorIndex)-1] or '?'
        ))
    else:
        for name, val in varBinds:
            print('%s = %s' % (name.prettyPrint(), val.prettyPrint()))
```

Optimización

Velocidad efectiva

20

- Velocidad a la que ocurren eventos.
- Ejemplos:
 - ▣ Transmisión o procesamiento de datos: bps, pkts/seg...
 - ▣ N° de llamadas a un servicio en un periodo de tiempo.
 - ▣ N° de conexiones en periodo de tiempo.
- Su monitorización permite:
 - ▣ Conocer la demanda de un determinado servicio (planificación).
 - ▣ Localización de problemas de prestaciones.

Ejemplos de velocidad efectiva

21

- Velocidad de segmentos:
 - ▣ Segmentos recibidos (tcpInSegs) o enviados (tcpOutSegs).
 - ▣ Medir la velocidad: n° de segmentos procesados en un periodo de tiempo entre los instantes i y j.
 - ▣ $V_{ef} = (\text{tcpInSegs}_j + \text{tcpOutSegs}_j) - (\text{tcpInSegs}_i + \text{tcpOutSegs}_i) / j - i$

- Porcentaje de la capacidad teórica del recurso que está siendo utilizada.
- Es una medida más fina que la velocidad efectiva.
- Se aplica a localización de potenciales cuellos de botella y áreas de congestión (t. respuesta crece exponencialmente con la utilización).

- Una aplicación práctica: comparar la utilización de varios enlaces de una red.
 - ▣ La carga de cada enlace nos permite comprobar el grado de utilización de cada enlace comparándolo con su capacidad.
 - ▣ La relación %carga/%capacidad nos permite comprobar si el reparto de carga es justo:
 - Medimos la carga de cada enlace (Kbps) en relación (%) con la carga total (suma de todas las cargas).
 - Medimos la capacidad de cada enlace (Kbps) en relación (%) con la capacidad total (suma de todas las capacidades).
 - En cada enlace se calcula: % de carga/% capacidad.

- La utilización U toma valores entre 0 y 1.
- **Si aumenta la utilización aparece congestión.**
- Una estimación de la relación Retardo-utilización:
 - ▣ D Retardo o Latencia efectiva
 - ▣ D_0 Retardo o Latencia en ausencia de tráfico
 - ▣ U utilización

$$D \approx \frac{D_0}{1 - U}$$

Ejercicio: cálculo de utilización de un interfaz

25

- Utilización de un interfaz:
 - ▣ Total de bytes que pasan por el interfaz por segundo:

$$\text{Bytes/s} = [(ifInOctets_j - ifInOctets_i) + (ifOutOctets_j - ifOutOctets_i)] / (j - i)$$

Donde j e i son dos instantes de tiempo.

$$\text{Utilización} = (\text{Bytes/s} * 8) / \text{ifSpeed}$$

Indicadores IP aplicados a prestaciones

26

□ Medidas de prestaciones:

1. **Porcentaje de datagramas IP recibidos: B/A**

A: Suma: iflnNUcastPkts + iflnUcastPkts por cada interfaz

B: ipSystemStatsInReceives

2. **Porcentaje de tráfico snmp o icmp (control).**

3. **Velocidad de reenvío:** dos medidas en instantes i y j.

$Ip-forwarding-rate = (ipSystemStatsForwDatagram_j - ipSystemStatsForwDatagram_i) / (j - i)$

4. **Velocidad de paquetes recibidos:** dos medidas en instantes i y j.

$Ip-input-rate = (ipSystemStatsInReceives_j - ipSystemStatsInReceives_i) / (j - i)$

Si Ip-forwarding-rate es similar a Ip-input-rate, buenas prestaciones.

Indicadores IP aplicados a prestaciones

27

- Problemas de prestaciones
 - ▣ **Indicador de falta de recursos:**
 - ipSystemStatsInDiscard, ipSystemStatsOutDiscard
 - ▣ **Carga del sistema** si se produce un gran porcentaje de **errores** o **fragmentaciones**:

% errorsIN =
$$(\text{ipSystemStatsInDiscards} + \text{ipSystemStatsInHdrErrors} + \text{ipSystemStatsInAddrErrors}) / \text{ipSystemStatsInReceives}$$

% fragmetsIN = $\text{ipSystemStatsReasmReqds} / \text{ipSystemStatsInReceives}$

Elegir el indicador más apropiado es complicado.

□ Diferentes aproximaciones

▣ Cuantificación de uno o más indicadores:

- Latencia
- Troughput
- Jitter
- Pérdida de paquetes
- Disponibilidad
-



▣ N° de reclamaciones de usuarios



Dependencias de la calidad

29

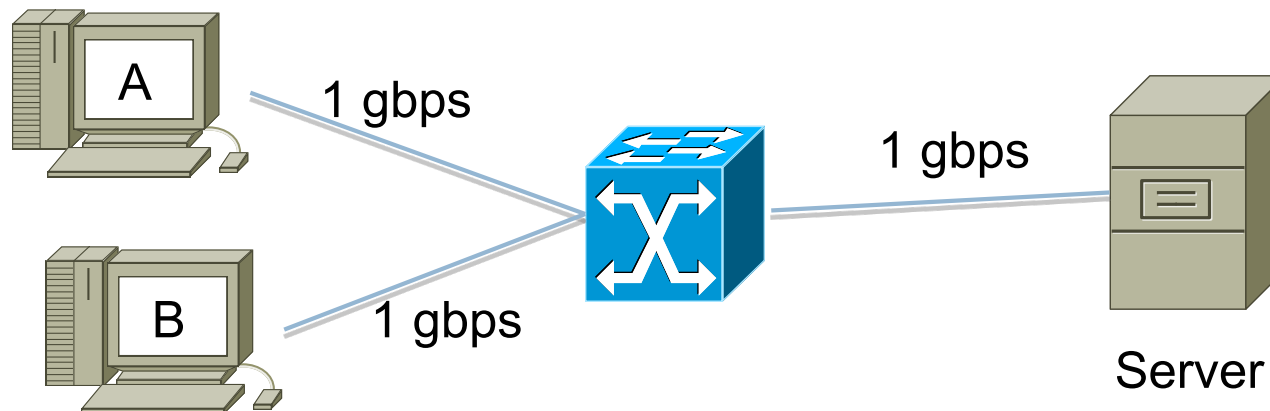
- EXTREMOS: depende de los enlaces extremo a extremo
 - ▣ Cada camino puede pasar por diferentes dispositivos con diferentes características y estado (nivel de congestión)
 - ▣ No tiene la misma calidad las comunicaciones entre los extremos A y B que entre los extremos C y D
- APLICACIONES: cada tipo de aplicación tiene unos requisitos diferentes de calidad.

	Remote Login	File Transfer	VoIP
Troughput	-	High	Low
Latency	Low	High	Low
Jitter	High	-	Low

Degradación del servicio

30

- Un servicio se degrada cuando alguno de sus indicadores tienen valores menores que los esperados.
- Causas:
 - ▣ Fallos: hardware que funciona mal y tira paquetes.
 - ▣ Problemas de rendimiento (principal causa):
 - Rutas no óptimas en routers
 - Cambios continuos de rutas en routers
 - Enlace de un conmutador saturado
 -



Gestión de Prestaciones y Optimización

- **Buffers** de paquetes en elementos de red (router, conmutadores...) permiten que estos elementos soporten ráfagas de tráfico.
- Si el buffer se llena se produce **congestión** que influye en:
 - ▣ Aumento de pérdida de paquetes: descarte de paquetes
 - ▣ Aumento de latencia: mayor retardo
 - ▣ Reducción del Throughput: menor velocidad efectiva

Ejercicio: congestión en interfaz

32

- Varios indicadores:
 - ▣ La longitud de la cola de salida permanece con un valor elevado.
 - ▣ El nº de paquetes descartados a la salida de un interfaz es alto y por el interfaz salen pocos octetos.

- ¿Cómo puedo detectar la congestión?

¿La longitud de la cola de salida permanece con un valor elevado?

- Monitorizar en intervalos de tiempo $j-i$ la evolución de $ifOutQLen$ (**Syntax Gauge32**)
- Valor cola interfaz salida en instante j : $A = ifOutQLen$
- Evolución de la cola: $B = ifOutQLen_j - ifOutQLen_i$
- Congestión: Si A elevado y B bajo

¿El nº de paquetes descartados a la salida de un interfaz es alto y por el interfaz salen pocos octetos?

□ Monitorizar en intervalos de tiempo $j-i$:

▣ $A = \text{ifOutDiscard}_j - \text{ifOutDiscard}_i$

▣ $B = \text{ifOutOctets}_j - \text{ifOutOctets}_i$

□ Congestión si A elevado y B bajo

- Objetivo:
 - ▣ Optimizar el rendimiento
 - ▣ Para ello hay que identificar cuellos de botella en la red
- Cuellos de botella: elementos o subsistemas con respuesta lenta
 - ▣ Enlaces saturados
 - ▣ Routers/servidores al límite de su capacidad
- Optimización actual:
 - ▣ Identificar cuellos de botella que se resuelvan con configuración o gestión de fallos
- Optimización futura:
 - ▣ Identificar cuellos de botella por incremento de solicitud de servicio para planificar futuro crecimiento de la red
 - ▣ Adquisición de nuevo hardware adaptado a las necesidades

- Medir el uso actual: tráfico
- Predecir el uso futuro: estimar tráfico futuro
- Elegir los componentes de red para que soporten la demanda estimada
- Posibles escenarios para mejorar los recursos de red según las estimaciones.
 - ▣ Incrementar el número de puertos de un elemento de red.
 - ▣ Añadir nuevos elementos de red.
 - ▣ Añadir nuevos enlaces.
 - ▣ Incrementar la capacidad de los enlaces.
 - ▣
- Elegir una solución que tenga una relación de compromiso entre rendimiento, fiabilidad y coste.

□ Conmutador

- ▣ N° de conexiones necesarias (puertos)
- ▣ Velocidad de cada conexión (10/100/1000 Mbps)
 - Configuración automática en el conmutador.

□ Router

- ▣ Varios servicios: encaminamiento, DHCP, IPv6, VPN...
- ▣ Capacidad de cada conexión
- ▣ Capacidad depende del entorno en el que el router se instale

□ Conexión a Internet

- ▣ Medir **utilización** del enlace con ISP por sencillez.
- ▣ Otros indicadores: latencia, pérdidas de paquetes, throughput o jitter.
- ▣ Tráfico variable: elegir compromiso entre coste del enlace y calidad

- **Utilización** máxima (pico) y media durante la semana
- Dividir semana en intervalos
 - ▣ Intervalos grandes (15 min):
 - Menos recursos
 - Válido para la mayoría de los casos. **Recomendado normalmente.**
 - 672 valores a la semana
 - ▣ Intervalos cortos (5 min, 1 min): mayor precisión en detección de ráfagas
- Crear una base repitiendo la medida durante sucesivas semana (hasta un mes, un año...).
- El tráfico normalmente es asimétrico, si es necesario medir en **ambas direcciones**:
 - ▣ Utilización subida: empresa-ISP
 - ▣ Utilización bajada: ISP-empresa (normalmente mayor)

- ¿Qué tenemos?
 - ▣ Estadísticas de utilización media y pico durante varias semanas en ambas direcciones.
- ¿Cómo lo uso para determinar si necesito aumentar la capacidad?
- Para prevenir pérdida de paquetes.
 - ▣ Actualizar capacidad antes de que picos de utilización lleguen a 100%.
 - ▣ Es necesario medir utilización absoluta máxima.
 - ▣ Seleccionamos el máximo valor de los obtenidos en la serie de medidas de 15 minutos.

- Eventos puntuales en routers pueden provocar **cortos picos en utilización** no indicativos de tráfico normal.
 - ▣ Condiciones de error
 - ▣ Cambio de rutas
 - ▣
- Queremos filtrar estos picos suavizándolos
- Calcular picos de utilización usando 95° percentil.
 - ▣ Medimos tráfico en intervalos de 15 minutos
 - ▣ Ordenamos la serie de valores de utilización en cada intervalo
 - ▣ Eliminamos el 5% de valores más altos
 - ▣ Seleccionamos el valor más alto del resto (95%)

Estimación realista de picos usando 95° percentil

- Experiencia en observación del tráfico de grandes organizaciones, en su enlace a Internet
 - ▣ Observación en **enlaces con fuerte carga**
 - ▣ No hay cambios bruscos en el tiempo
 - ▣ No es necesaria una observación precisa
 - ▣ Existe una relación aproximada entre la utilización media y pico

$$\frac{Pico_Utilización_estimado}{Utilización_media} \approx 1.3$$

El valor estimado de la relación depende de la organización

Se puede evitar el cálculo de picos

Regla 50/80

42

- Aplicable a enlaces con fuerte carga
- El gestor mide la utilización media y hace estimaciones:

Utilización media	Pico	Estado del enlace
40%	52%	Infrautilizado
50%	65%	Normal
60%	78%	Muy utilizado
70%	91%	Saturado
80%	100%	Completamente saturado

- Utilización media durante los periodos de actividad.

La utilización media se debe mantener entre el 50% y 80% de la capacidad del enlace.