Introducción a las Redes de Computador{ae}s y Comunicación de Datos

Examen – 21 de febrero de 2011

(ref: eirc1103.odt)

Instrucciones

- **Indique su nombre completo y número de cédula en cada hoja.**
- **Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.**
- **Escriba las hojas de un solo lado y utilice una caligrafía claramente legible.**
- **Comience cada pregunta teórica y cada ejercicio en una hoja nueva.**
- **Sólo se contestarán dudas de letra. No se aceptarán dudas de ningún tipo los últimos 30 minutos del examen.**
- **El examen es individual y sin material. Apague su celular mientras este en el salón del examen.**
- **Es obligatorio responder correctamente al menos 15 puntos en las preguntas teóricas.**
- **El puntaje mínimo de aprobación es de 60 puntos.**
- **Para todos los ejercicios, si es necesario, puede suponer que dispone de los tipos de datos básicos (p.ej. lista, cola, archivo, string, etc.) y sus funciones asociadas (ej: tail(lista), crear(archivo), concatenar(string, string).**
- **Duración: 3 horas. Culminadas las 3 horas el alumno no podrá modificar las hojas a entregar de ninguna forma.**

Preguntas Teóricas

Pregunta 1 (8 puntos)

Describa claramente las dos funciones clave de la capa de red y explique cómo interactúan entre ellas.

Reenvío (forwarding): mover paquetes entre puertos de entrada y salida del router.
Enrutamiento (routing): determinar la ruta de los paquetes desde origen a destino. Para ello existen los algoritmos de enrutamiento.
Interacción: Cada router intercambia con sus pares, mediante al menos un protocolo de enrutamiento, información que se utiliza, en al menos un algoritmo de enrutamiento, para poblar sus tablas de forwarding. Dichas tablas, en su formato más básico, contiene, para un valor de dirección contenido en la cabecera de los paquetes que llegan al router, el enlace de salida. De esta forma, para cada paquete que arriba a un router por cualquiera de sus interfaces, el router examina la tabla de forwarding (en el caso de IP a partir de la dirección IP destino contenida en el encabezado IP) y a partir del "longest match" determina por qué interfaz debe ser enviado (además de determinar cuál es el "next hop", lo que determina en el caso más genérico, la dirección destino de capa de enlace que deberá contener la trama que lo lleve a su próximo salto).
Referencia:
http://www.fing.edu.uy/inco/cursos/redescomp/teorico/cap04_1_1xhoja_2009.pdf diapositivas 5 y 6.

Pregunta 2 (6 puntos)

Dada una red de conmutación de paquetes, explique los conceptos de congestión, pérdidas y retardo.

Congestión: La congestión está asociada a la escasez, durante un período de tiempo, de los recursos (compartidos) necesarios para procesar adecuadamente todos los paquetes que pretenden hacer uso de los mismos. Esos recursos son básicamente: capacidad de procesamiento y buffers en los routers y, enlaces. De ocurrir congestión en algún segmento de red (conjunto de router/s y/o enlace/s) ella se manifestará en *pérdidas* y/o *retardos* de paquetes, para lo cual algunos protocolos (por ejemplo TCP) implementan ciertas medidas proactivas y/o reactivas para evitar la congestión o minimizar su impacto buscando una mayor resiliencia de la red a la misma.
Referencias:
http://www.fing.edu.uy/inco/cursos/redescomp/teorico/cap01_2010_2xhoja.pdf diapositivas 36 y 40.
http://www.fing.edu.uy/inco/cursos/redescomp/teorico/cap03_4_2009_3xcarilla.pdf diapositivas 2 y 3.

Los paquetes que circulan por una red de conmutación de paquetes están expuestos a *pérdidas* y *retardos*, lo que puede terminar afectando a la aplicación cuyos datos transportan.
Pérdida: Los paquetes se encolan en buffers en los routers antes de ser enviados por el enlace de salida; si un paquete no encuentra espacio disponible en un buffer, será descartado (*pérdida*).
Retardo: Si un paquete encuentra lugar en el buffer pero su envío por el enlace de salida se ve retrasado (por existir otros paquetes esperando poder hacerlo antes que él) sufrirá de *retardo* (retardo de cola). También puede sufrir *retardo* por el procesamiento que se le debe realizar en el nodo previo a ser colocado en el buffer de salida (chequeos de errores, modificaciones de campos de cabecera, ...) Adicionalmente habrá *retardo* debido a la propagación del paquete en el medio físico y también motivado en el ancho de banda disponible en el enlace de salida y su relación con el tamaño del paquete que se desea enviar (retardo de transmisión).
Referencia:

http://www.fing.edu.uy/inco/cursos/redescomp/teorico/cap01_2010_2xhoja.pdf diapositivas 47-50.

Pregunta 3 (6 puntos)

Explique el principio de funcionamiento del comando *traceroute* indicando además los diferentes tipos de mensajes involucrados y los nodos que colaboran con el comando.

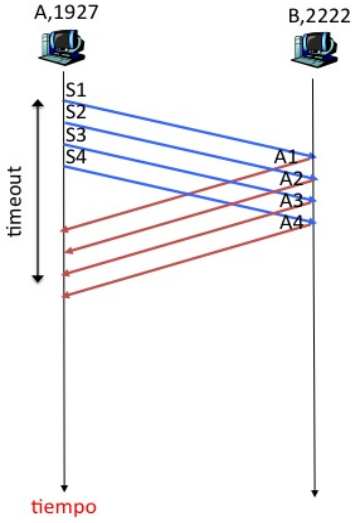
El comando traceroute es una herramienta de resolución de problemas en redes de computadoras que permite identificar el camino *de ida* entre un origen (donde se ejecuta el comando) y un destino. En general se lo encuentra en "dos sabores": basado en UDP (en general configuración por defecto en sistemas Unix, Linux, *BSD, Mac OS) o basado en ICMP (en general en sistemas Windows), aunque también existen otras implementaciones, pero menos utilizadas. En ambos la filosofía es la misma: se envían mensajes especialmente formados de forma de buscar forzar respuestas de cada nodo en el camino de ida hacia el destino. Ello se realiza estableciendo adecuadamente el valor del campo TTL (Time To Live) de la cabecera IP. Por defecto, en cada instancia de envío de dichos datagramas de prueba se envían 3 paquetes. En la primer instancia, se fija el valor del campo TTL en 1, en la segunda en 2 y así hasta llegar a destino o a un número máximo de instancias, que por defecto es en general 30 (se entiende que 30 es un número más que suficiente para una topología como la red Internet) De esta forma, quien recibe dichos paquetes, decrementará el TTL y si no es el destino, informará de ello a la fuente mediante un mensaje de error del protocolo ICMP. Ahora describiremos en qué se diferencian los dos sabores antes mencionados. En el basado en UDP se envían mensajes UDP (*probes*) con puerto destino muy alto (alrededor del 33400; de probabilidad de uso muy baja; por defecto es el 33434); en el basado en ICMP, se envían mensajes ICMP "Echo request" (Tipo 8, Código 0). Mientras los paquetes de las diferentes instancias no llegan a destino, cada nodo intermedio devolverá mensajes ICMP "Time Exceeded"-"TTL expired in transit" (Tipo 11, Código 0), error forzado a partir de establecer el TTL de manera adecuada (comenzando en 1 e incrementando en 1 en cada instancia mientras no se llega a destino). Para cada respuesta a cada *probe* se despliega el RTT (Round Trip Time) correspondiente, aun cuando provengan de hosts distintos en cada instancia. Al llegar al destino, en el caso del traceroute basado en UDP, se devolverá un mensaje ICMP "Destination Unreachable"-"Destination Port Unreachable" (Tipo 3, Código 3) por cada mensaje UDP recibido; en el caso del traceroute basado en ICMP se devolverá un mensaje ICMP "Echo Reply" (Tipo 0, Código 0) por cada mensaje ICMP recibido (El traceroute basado en ICMP, al llegar a destino, es el comando "ping").
Dado que el camino que siguen los paquetes de ida y vuelta de una comunicación sobre Internet no necesariamente son el mismo, las respuestas, tanto de los nodos intermedios como del destino pueden recorrer un camino distinto al que siguieron los paquetes que las motivaron.
Además, puede ocurrir que tanto en los nodos intermedios así como en el destino los mensajes "de prueba" enviados por el origen (UDP o ICMP) así como sus respuestas, estén filtradas (por razones de seguridad y/o performance) por lo que el no obtener respuesta desde los nodos intermedios o desde el nodo final no nos permite concluir que haya algún problema en la red o en el destino.
Referencia:
http://www.fing.edu.uy/inco/cursos/redescomp/teorico/cap01_2010_2xhoja.pdf diapositivas 54 y 55.
http://www.fing.edu.uy/inco/cursos/redescomp/teorico/cap04_2_3xhoja_2009.pdf diapositivas 61 y 62.

Pregunta 4 (10 puntos)

Suponga una comunicación TCP entre dos procesos A (dir IP = A; puerto TCP = 1927) y B (dir IP = B; puerto TCP = 2222). En un momento determinado A envía a B cuatro segmentos seguidos (no hay ningún dato previo sin asentir), el primero con número de secuencia igual a 1100 y con 20 bytes de datos, y los tres restantes con otros 20 bytes de datos cada uno.

a) Si B no tiene datos que enviar hacia A y no hay pérdidas de segmentos: represente en un diagrama el intercambio de segmentos resultante hasta que ninguno de los dos lados tenga nada que enviar.

Si suponemos que el timeout es mayor que el tiempo de llegada del 1er ACK, y las latencias son aprox. uniformes, las entidades TCP se comportan "normalmente", como se ve en la figura.

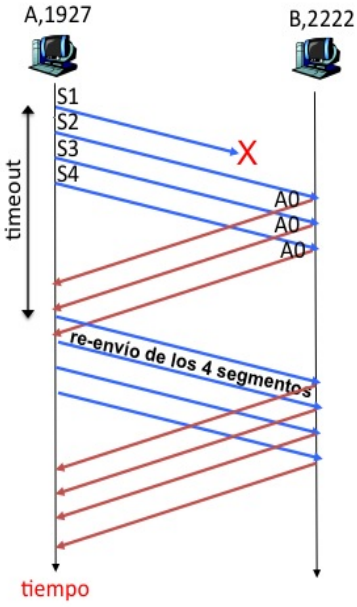


Campos relevantes:

S1: Sec=1100, 20 bytes de datos
S1: Sec=1120, 20 bytes de datos
S1: Sec=1140, 20 bytes de datos
S1: Sec=1160, 20 bytes de datos

A1: ACK=1120
A2: ACK=1140
A3: ACK=1160
A4: ACK=1180

b) Represente de nuevo el diagrama si B no tiene datos que enviar hacia A y se pierde el primer segmento enviado de A a B.



Si se pierde el primer segmento S1, cuando llegue S2, el receptor reenviará el ACK correspondiente al segmento esperado, que llamamos A0. Sucede lo mismo cuando llegan S3 y S4.

El emisor comenzará a retransmitir los segmentos cuando pase alguno de estos dos eventos:

– vencimiento del timeout de S1 (caso reprsentado en la figura), o
– llegada del 3er ACK duplicado.

Campos relevantes:

S1: Sec=1100, 20 bytes de datos
S1: Sec=1120, 20 bytes de datos
S1: Sec=1140, 20 bytes de datos
S1: Sec=1160, 20 bytes de datos

A0: ACK=1100
Cuando se reenvían los segmentos, estamos en el caso a)

Nota: Detalle para todos los apartados anteriores el contenido de los campos relevantes de la cabecera TCP. Haga las suposiciones que considere necesarias, justificándolas.

Pregunta 5 (10 puntos)

Considere un paquete de longitud L que tiene su origen en el sistema terminal A y que viaja a través de tres enlaces hasta un sistema terminal de destino B. Estos tres enlaces están conectados mediante dos dispositivos de conmutación de paquetes. Sean d_i, s_i y R_i la longitud, la velocidad de propagación y la velocidad de transmisión del enlace i, para i = 1, 2, 3. Cada dispositivo de conmutación de paquetes retarda cada paquete d_{proc} .

a) Suponiendo que no se produce retardo de cola, ¿Cuál es el retardo total terminal a terminal del paquete en función de d_{proc} ,d_i, s_i, R_i , (i = 1, 2, 3) y L?

En general el retardo extremo a extremo es la suma de los componentes del retardo: procesamiento, cola, transmisión y propagación. Si despreciamos el retardo de cola, el retardo de extremo a extremo se expresa como:

$$d_{end2end} = \sum_{i=1}^{Q} d^{i}_{proc} + \frac{L}{R_i} + \frac{d_i}{s_i}$$

Para el caso concreto, se reduce a:

$$d_{end2end} = d_{proc} + d_{proc} + \frac{L}{R_1} + \frac{L}{R_2} + \frac{L}{R_3} + \frac{d_1}{s_1} + \frac{d_2}{s_2} + \frac{d_3}{s_3}$$

b) Suponga ahora que *R1 = R2 = R3 = R* y d_{proc}= 0. Suponga también que el conmutador de paquetes no almacena los paquetes y los reenvía, sino que transmite inmediatamente cada bit que recibe sin esperar a que llegue el paquete completo. ¿Cuál será el retardo terminal a terminal?

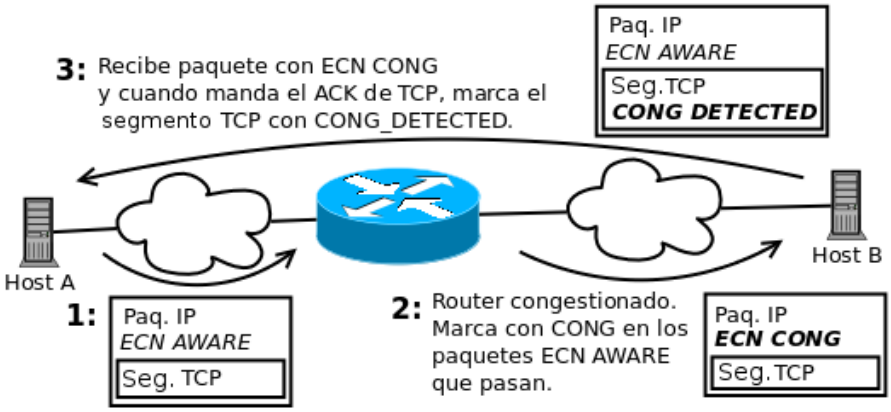
Dado que los bits se transmiten inmediatamente, los conmutadores de paquetes no introducen ningún retardo (en particular no introducen retardo de transmisión), y por lo tanto ahora el retardo de extremo a extremo se expresa como:

$$d_{end2end} = \frac{L}{R} + \frac{d_1}{s_1} + \frac{d_2}{s_2} + \frac{d_3}{s_3}$$

Problemas Prácticos

Problema 1 (30 puntos)

ECN (*Explicit Congestion Notification*) es un protocolo que actúa en capa de red y capa de transporte que permite que los *hosts* de una sesión TCP reciban notificaciones explícitas de congestión generadas por los routers de la red que reenvian los paquetes de dicha sesión. Un paquete IP puede estar marcado como: NO_ECN, ECN_AWARE o ECN_CONG.



TCP puede activar ECN solicitando a su entidad de red marcar la bandera ECN_AWARE del encabezado IP de todos los paquetes que trasportan segmentos de dicha sesión. En caso de congestión, los routers deben marcar la bandera ECN_CONG del encabezado IP en todos aquellos paquetes marcados con ECN_AWARE antes de reenviarlos. Cuando el receptor recibe paquetes marcados con la bandera ECN_CONG le notifica al emisor que existe congestión marcando la bandera CONG_DETECTED del encabezado de los segmentos TCP.
En el diagrama se muestra un ejemplo de funcionamiento.

Considere dadas las siguientes primitivas y tipos de datos:

Primitivas:

- `void setCongWinSize(int valor)`/ `int getCongWinSize()`: Establece/lee el tamaño actual de la ventana de congestión.
- `void setCONGReceived()`: Marca los próximos segmentos TCP a enviar con la bandera `CONG_DETECTED`.

Tipos:

```
TCP_Segment{
    int destPort, srcPort, seqNum, ackSeqNum
    void[] data
    bool cong_detected
    IP_Packet prev_header // Header IP del
paquete TCP
    ...
}
```

```
IP_Packet{
    ip_addr src, dst
    enum_ecn ecn_field // Valores: {NO_ECN,
ECN_AWARE, ECN_CONG}
    ...
}
```

Se pide:

a) Cuando el emisor recibe un segmento TCP con la bandera CONG_DETECTED marcada, interpreta dicha situación como congestión. ¿Qué eventos interpreta TCP sin ECN como una señal de congestión en la red?

b) Considere una implementación de TCP que carga el segmento TCP recibido en una estructura de tipo `TCP_Segment` que se usa como parámetro para invocar la función `int TCP_congDetection(TCP_Segment segmento)` cuya salida es el estado de congestión. Implemente dicha función considerando los mecanismos tradicionales de detección de congestión de TCP y las mejoras provistas por ECN (si se invoca `TCP_ congDetection` con NULL como parámetro, es porque existió un *timeout* para la recepción de un ACK).

c) TCP utiliza la función `TCP_congDetection` y le pasa el resultado a la función `void TCP_congControl(Int congStatus)` que implementa el control de congestión tradicional de TCP y los mecanismos necesarios para utilizar ECN.
Implemente dicha función considerando que una vez detectada la congestión se pasa al estado “slow start” (arranque lento).

Solución:

a) TCP interpreta que hay congestión en la red cuando se da un evento de pérdida de paquete. Un evento de pérdida puede ser por timeout en la recepción de un ACK o por la recepción de 3 ACKs duplicados.

b)

```
int ultimo_ack, contador = 0;

int TCP_extractCongData(TCP_Segment paquete){
    int estadoCongestion = 0;

    if ( paquete == NULL ){
        contador = 3;
    }
    else if ( paquete.ackSeqNum == ultimo_ack ){
        contador++;
    }
    else if ( paquete.ackSeqNum > ultimo_ack ){
        ultimo_ack = paquete.ackSeqNum;
        contador = 0;
    }

    if ( contador == 3 ) {
        estadoCongestion = 1;
        contador = 0;
    }

    if ( paquete != NULL and paquete.CONG_DETECTED ) {
        estadoCongestion = 1;
    }

    if ( paquete != NULL and paquete.prev_header.ecn_field == ECN_CONG ){
        setCONGReceived();
    }

    return estadoCongestion;
}
```

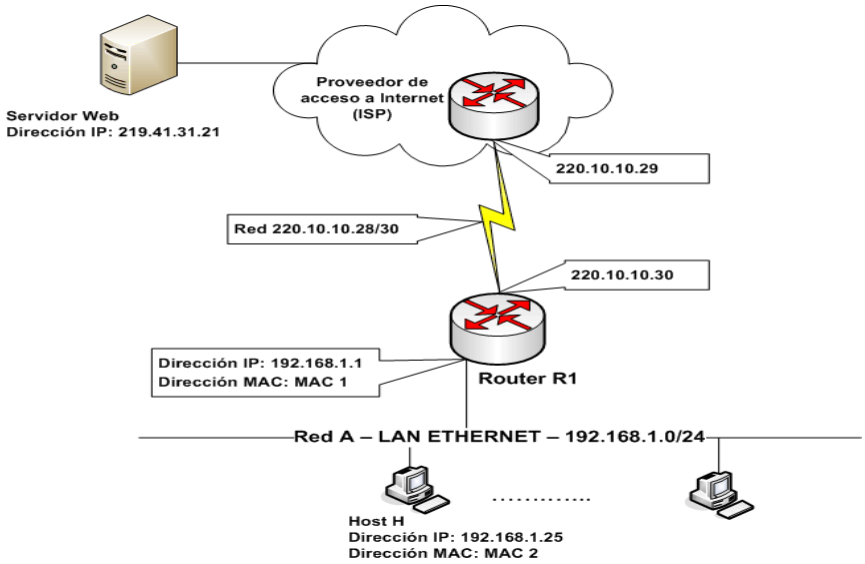
c)

```
#define MSS_BYTES ...
int estado = SLOW_START;
int umbral = valor_inicial;
int ultimoTamañoCongWin;

void TCP_congControl(int congStatus){
       if(congStatus = 1){
              estado = SLOW_START;
             umbral = getCongWinSize() / 2;
              setCongWinSize(MSS_BYTES);
       }
       else{
              if (estado == SLOW_START) {
                     setCongWinSize(MSS_BYTES + getCongWinSize());
                     if(getCongWinSize() > umbral){
                       estado = CONG_AVOIDANCE;
                  }
             }
              else{
                     setCongWinSize(getCongWinSize() + MSS_BYTES *(MSS_BYTES/getCongWinSize()))
             }
      }
}
```

Problema 2 (30 puntos)

Una empresa tiene una red A (LAN Ethernet con direccionamiento privado) con 15 servidores y 70 puestos de trabajo, y se desea que puedan acceder a un servidor web en Internet con dirección IP 219.41.31.21 (ver figura). Para ello el router R1 implementa NAT utilizando su dirección IP pública.



Se pide:

a) Especifique la tabla de enrutamiento (routing) de R1 y del Host H.

b)

i. Suponiendo que la tabla ARP del Host H está vacía, indicar los mensajes ARP (MAC de origen, MAC de destino y descripción del contenido) intercambiados en la Red A cuando el Host H inicia su comunicación con Servidor Web (asuma que Host H conoce la dirección IP de Servidor Web).

ii. Indique las direcciones MAC de las tramas y las direcciones IP de los paquetes correspondientes a un flujo de datos entre el Host H y el Servidor Web considerando ambos sentidos (considere únicamente la Red A).

c) Especifique las modificaciones que realizará R1 en los paquetes de datos que viajan entre Host H y Servidor Web. Considerar ambos sentidos.

d) Se adquiere para la Red A el bloque de direcciones IP públicas 100.100.100.0/24, y los administradores deciden aislar la red de servidores de los puestos de trabajo. Especifique las subredes del bloque mencionado que cumplan con este requerimiento y permitan dejar la mayor cantidad posible de direcciones IP para uso futuro. Mencione las modificaciones necesarias en el router R1 y su nueva tabla de enrutamiento.

Solución:

a) TABLA de enrutamiento del host H

Red destino / Máscara	Gateway / next hop
192.168.1.0/24	C – Directamente conectado
0.0.0.0/0	192.168.1.1

TABLA de enrutamiento de R1:

Red destino / Máscara	Gateway / next hop
192.168.1.0/24	C – Directamente conectado
220.10.10.28/30	C – Directamente conectado
0.0.0.0/0	220.10.10.29

b) i) Primero el host H envía el siguiente mensaje ARP:

MAC origen: MAC 2
MAC destino: FF:FF:FF:FF:FF:FF
Tipo de ARP: ARP Request preguntando por la MAC asociada a la dirección IP 192.168.1.1

El router R1 le responde al HOST H:

MAC origen: MAC1
MAC destino: MAC 2
Tipo de ARP: ARP Reply respondiendo que la MAC asociada a la dirección IP 192.168.1.1 es MAC 1.

b) ii) Para los datos originados en H y destinados al servidor web :

MAC origen: MAC 2
MAC destino: MAC 1

Dirección IP origen: 192.168.1.25
Dirección IP destino: 219.41.31.21

Para los datos originados en el servidor web y destinados a H:

MAC origen: MAC 1
MAC destino: MAC 2

Dirección IP origen: 219.41.31.21
Dirección IP destino: 192.168.1.25

c)

Sentido host H → Servidor Web

Datagramas con dirección IP de origen 192.168.1.25 y dirección IP de destino 219.41.31.21.

R1 sustituye la IP de origen por 220.10.10.30, y el puerto de origen (port #) por un puerto nuevo (new port #) asignado por R1.

El router R1 tendrá una tabla de NAT con los siguientes datos:

(Dirección IP origen, port #), (220.10.10.30, new port #)

Sentido servidor Web → host H

Datagramas con dirección IP de origen 219.41.31.21, dirección IP de destino 220.10.10.30, y puerto de destino new port #.

A partir de la tabla de NAT, R1 sustituye la IP de destino por 192.168.1.25, y el puerto de destino por port #.

d)

Subred para 70 puestos de trabajo:

70 direcciones IP + 1 para el router R1
(Además siempre se deben considerar la direcciones de red y de broadcast)
→ 100.100.100.0/25

Subred para 15 servidores:

15 direcciones IP + 1 para el router R1
(Además siempre se deben considerar la direcciones de red y de broadcast)
→ 100.100.100.128/27

El router R1 se conecta a cada subred y no es necesario que implemente NAT, ya que las estaciones de trabajo y los servidores tienen direcciones IP públicas.

Nueva tabla de enrutamiento de R1:

Red destino / Máscara	Gateway/ next hop
100.100.100.0/25	C – Directamente conectado
100.100.100.128/27	C – Directamente conectado
220.10.10.28/30	C – Directamente conectado
0.0.0.0/0	220.10.10.29