

EDA095 - Nätverksprogramering

Måns Ansgariusson, C14

2016-05-13



1 Lager

1.1 Applikationslagret

För att två program ska kunna kommunicera, så krävs det att de använder sig av samma protokoll. Det finns flera lager inom nätverkskommunikation som har olika uppgifter och använder sig av olika protokoll.

1.2 Transportlagret

Transportlagret bidrar till att applikationer ska kunna skicka meddelanden mellan varandra. Med hjälp av portnummer, som är en rent logisk abstraktion, mappas meddelanden till rätt applikation. De två vanligaste protokollen är TCP (Transmission control protocol) och UDP (User datagram protocol).

TCP har en fast (logisk) uppkoppling över nätverket. Dataström, dvs. data behöver inte delas upp i paket. Automatisk felkontroll/omsändning. Garanterad leverans och inbördes ordning

UDP använder sig av datagram. Upp till 65507 byte stora datagram (IPv4) Garanterat korrekta meddelanden. Ej garanterad leverans eller inbördes ordning.

1.3 IP-lagret

Internetlagret bidrar till att program ska kunna koppla upp sig till rätt nätverksenhet. Detta sker med hjälp av IP (internet protocol), och är en adressering som kan jämföras med gatuadress. Idag finns det två varianter av IP, IPv4 och IPv6.

IPv4 använder sig av 4 bytes och kan exempelvis skrivas 130.235.16.34.

IPv6 använder sig av 16 bytes, exempel på samma adress som ovan i IPv6: 2001:fe0c::db:1dc0.

IPv6 har tillkommit på grund av att IPv4 numren börjar ta slut. Båda varianterna kommer att användas parallellt i övergångsfasen. Moderna datorer stödjer båda varianterna. Mycket dåligt stöd bland ISPs. Har lite inverkan på utvecklare, undvik dock antagandet att IP skrivs på x.x.x.x.

Domännamn kan användas istället för IP-nummer, med hjälp av DNS (domain name server) så översätts domännamnen till IP-nummer. `www.cs.lth.se` = 130.235.16.34

2 Strömmar

En ström är en sekvensiell följd av bytes (tecken). In- och utmatning sker oftast i form av strömmar: Inmatning från tangentbordet, utskrift till ett terminalfönster. Vi kan skriva bytes till strömmar och vi kan läsa bytes från strömmar. Generell abstraktion. Vi kan upprätta strömmar över ett nätverk: TCP

InputStream representerar inkommande ström, t.ex. läsa från fil.

OutputStream representerar utgående ström, t.ex. skriva till fil.

Buffring av strömmar ökar effektiviteten, högre genomströmningshastighet. Mer data som sänds/tas emot -> färre systemanrop/diskaccesser/nätverkspaket. Klasserna `BufferedInput/OutputStream` implementerar buffring. Glöm inte anropa `flush`!

`Reader/Writer` implementerar möjligheten att skicka/ta emot tecken utan att omvandla dessa till bytes manuellt.

3 URL

URL = Uniform resource locators.

En URL identifierar en viss resurs på nätet.

3.1 URI

URI = Uniform Resource Identifier.

4 REST

I en REST-arkitektur använder man HTTP-metoder för att interagera med en server. REST-arkitekturen bygger på HTTP, URI/URL och XML/HTML.

- POST skapar data.
- GET hämtar statements från repo.
- PUT uppdaterar data, skriver över.
- DELETE raderar statements från repo.

5 Busy-Wait

Busy-wait innebär att en tråd förbrukar all tillgänglig CPU-tid för att vänta på att en händelse ska inträffa, t.e.x att en annan tråd ändrar ett värde. Det leder till dåligt processorutnyttjande och dåliga prestanda.

RMI - The Java Remote Method Invocation

RMI, är en Javastandard för anrop av distribuerade objekt och komponenter.

5.1 Registry

Ett registry är ett separat serverprogram som används för att en RMI-klient och en RMI-server ska kunna koppla ihop sig med varandra. Det fungerar som en förteckning över vilka tjänster som olika RMI-servrar erbjuder.

5.2 Serialisering

Serialisering innebär att javaobjekt kodas om till en seriell ström av bytes – typiskt i syfte att överföras via ett nätverk. På mottagarsidan gör man sedan den omvända operationen

5.3 stub/skeleton

Stub(klient)/skeleton(server) är de programkomponenter på klient- respektive serversidan som implementerar RMI-anropen. På klienten representerar en s.k. stubb det distribuerade objektet. När metoder anropas på stubben omvandlas dessa till nätverksanrop som tas emot av ett s.k. skeleton på serversidan. Detta senare objekt gör sedan det verkliga metदानropet på det distribuerade objektet.

6 XML

XML, EXtensible Markup Language, är en specifikation för att specialdesigna så kallade märkspråk, som du sedan kan använda till att spara och dela data på ett strukturerat och välformaterat sätt.

Parser: DOM

DOM bygger upp ett träd som representerar XML-dokumentet på vilket man sedan kan utföra godtyckliga operationer (på hela eller delar av trädet).

DOM normalt sett kraftfullare då ett komplett parseträd byggs upp på vilket man kan göra mera omfattande analyser eller transformationer.

Parser: SAX

SAX inte sparar någon fullständig representation av dokumentet. I det senare fallet kan man utföra operationer efterhand som parsern stöter på de olika taggarna.

SAX är oftast snabbare än DOM.

7 TextStandarder

7.1 Unicode

Unicode är en industristandard som låter datorer hantera text skriven i världens alla skriftsystem.

Unicode är en familj av kodningsstandarder som bygger på en gemensam teckentabell innehållande en stor mängd med tecken ur olika alfabet. Varje tecken motsvaras av en särskild kod. Unicode säger dock inget om exakt hur ett tecken ska representeras på bit-nivå. För detta behöver man en specifik

kodningsstandard, t.ex. UTF-8 som i detalj specificerar hur den teckenkod i Unicode som representerar ett visst teckna ska lagras.

Unicode består av en repertoar med mer än 100 000 skrivtecken. Ett av de viktigare målen är att alla tecken i världens alla skriftsystem ska finnas med: bokstäver, siffror, skiljetecken, matematiska symboler, och så vidare.

Unicode består även av ett antal metoder för att lagra tecknen i datorer, bland annat UTF-8 och UTF-16.

UTF-8

I UTF-8 lagras ett tecken som en följd av en eller flera bytes. De vanligaste tecknen representeras av en enkel byte medan de mera ovanliga representeras av en sekvens av upp till fyra bytes.

UTF-8 har valts som huvudsaklig teckenkodning i internetprotokoll: nya protokoll måste ge stöd till denna teckenkodning, om det inte av speciella skäl är olämpligt.

7.2 MIME-tag

Serverar sätter MIME header i början av varje webböverföring. Klienter använder denna typ av innehåll eller mediatyp rubrik för att välja en lämplig "player" ansökan om vilken typ av data rubriken anger. Några av dessa spelare är inbyggda i webbklienten eller webbläsare (till exempel, nästan alla webbläsare komma med GIF och JPEG-bild spelare samt förmågan att hantera HTML-filer); andra spelare kan behöva laddas ner.

8 Protokoll

8.1 HTML - Hypertext markup language

HTML är ett märkspråk för hypertext och utgör tillsammans med TCP/IP och HTTP den grundläggande standarden för World Wide Web. Webbssidor skrivs i allmänhet som HTML och överförs över Internet med HTTP.

HTML ger möjlighet att ange ett dokumentets struktur (rubriker, styckeindelning m.m.), metainformation (språk, författare, plats i en hierarki) och i viss mån hur dokumentet skall visas.

HTML tillåter också att man infogar information av andra typer, eventuellt i andra filer, från början framförallt bilder, senare till exempel datorprogram (java, javascript) och instruktioner för hur sidan skall visas (CSS).

Det finns två parser som vi använt i kursen: HTMLEditorKit och jsoupe. HTMLEditorKit är en eventbaserad parser och är ganska begränsad i sin funktionalitet samt dokumentation.

Jsoupe är baserat på ett yttre libery och det är fullt funktionellt och lätt att använda men det kräver som sagt att man måste hämta hem ett externt libery.

HTTP - Hypertext Transfer Protocol

HTTP är det kommunikationsprotokoll som används för att överföra webbsidor på informationsnätverket World Wide Web till en webbklient.

Det ursprungliga syftet med HTTP var att tillhandahålla en metod för att överföra HTML-sidor från webbservrar till webbklienter.

HTTP oftast beroende på GET eller POST definierar åtta åtgärder som en klient kan begära utförande av på en fil på en HTTP-server:

- GET – Ber servern att skicka den utpekade filen till klienten. Detta är i särklass det mest använda HTTP-kommandot.
- HEAD – Ber servern att skicka information om den utpekade filen, men utan att skicka själva innehållet i filen.
- POST – Sänder någon form av information, till exempel användarens namn och lösenord till servern.
- PUT – Om tillåtelse ges, laddar upp en utpekad fil från klienten till servern för lagring.
- DELETE – Raderar den utpekade filen. Detta kommando används sällan och många webbservrar har inget stöd för det.
- TRACE – Ber servern att skicka tillbaka klientförfrågan precis i det skick som den anlände till servern. Detta kommando kan användas för att kontrollera om någon tredje part mellan klient och server har gjort några ändringar i förfrågan.
- OPTIONS – Returnerar en lista över de HTTP-kommandon som servern stöder.
- CONNECT – Används med proxy-servrar som kan fungera som SSL-tunnlar.

Metoderna GET och HEAD är definierade som säkra, med andra ord avsedda uteslutande för informationshämtning. Icke-säkra metoder (som POST, PUT och DELETE) bör presenteras i webbklienten på något särskilt sätt (till exempel som knappar istället för länkar), så att användaren blir medveten om de potentiella effekterna av deras användning.

RTP-Real time Transport Protocol

RTP är specificerat för att bära dataströmmar i realtid. Exempel är ljud eller video. Vid första specifikationen var den tänkt att användas som ett multicast-protokoll. I senare utgåva tillkom RTCP som är kontrollprotokollet till RTP. Idag används RTP oftast ovanpå UDP för att bära telefonsamtal över Internet, även kallat VoIP.

RTCP - Real time Control Protocol

ett systerprotokoll till Real-time Transport Protocol som hjälper till med hantering av underhållsinformation och används av båda sändaren och mottagaren. Från sändaren kommer exempelvis info om vad klockan var när ett visst RTP-paket skickades. Från mottagaren kommer information om till exempel antalet tappade paket och jitter¹.

¹Jitter kallas den effekt som uppstår då ett förlopp som skall upprepas med jämna intervall inte sker i stabil takt

RTSP - Real Time Streaming Protocol

RTSP är ett standardprotokoll för strömmande media.

RTSP är ett kontrollprotokoll som fungerar tillsammans med dataleveransprotokollet RTP för att tillhandahålla innehåll för klienter. Används för att etablera och synkronisera multimedieströmmar. Det är inget transportprotokoll utan förlitar sig på UDP eller RTP för transport.

Kan liknas vid en sort fjärrkontroll som kan starta eller pausa en ström.

UDP - User Datagram Protocol

UDP är ett förbindelseöst protokoll i transportskiktet för att skicka datagram över ett IP-nätverk. Med förbindelseöst (eng. stateless eller connectionless) menas att ingen session upprättas mellan sändare och mottagare av protokollet i sig. Härvid kan inte sändare garantera² att mottagaren får paketet. Mottagaren kan heller inte veta att den fått alla paket, eller att den fått paketen i rätt ordning. För att kunna garantera att paket kommer fram och i rätt ordning så bör TCP användas istället för UDP.

UDP lämpar sig bra på livestreaming utan playback eller för online dataspel där reaktion är allt (Tänk Cs) då information om vad som hände för 1 sekund sen inte är intressant utan endast den senaste informationen är relevant till användaren.

TCP - Transmission control protocol

TCP är ett förbindelseorienterat dataöverföringsprotokoll som används för huvuddelen av all kommunikation över Internet. TCP tillhandahåller en relativt pålitlig dataström mellan två datorer och används för exempelvis HTTP, FTP och e-post. TCP är mindre lämpligt i situationer där dess felkorrigeringsegenskaper kan orsaka oönskade fördröjningar, exempelvis i online datorspel. TCP startar upp en stabil förbindelse mellan två användare till skillnad från UDP som kör efter *best effort delivery principen*³

Stuff to know

En tråd får inte anropa någon av operationerna `wait()`, `notify()` eller `notifyAll()` såvida den inte först låst objektet i fråga genom att gå in i ett `synchronized-block` eller anropa en operation som är deklarerad `synchronized`.

Om man deklarerar metod `synchronized` i Java medför det att endast en tråd kan exekvera den metoden samtidigt. Andra trådar kan dock exekvera samma `synchronized-metod` i ett annat objekt.

²Men det finns sätt att jobba runt detta i t.ex applikationer

³Best effort delivery är exakt vad det låter som. Det är ett leverans alternativ som inte garanterar att de paket som skickas kommer fram utan helt enkelt gör sitt bästa för att leverera en användares paket (OBS EXTREMT förenklad förklaring)

8.2 JSP - Java Server Pages

En JSP är en HTML fil med några extra element. JSP elementen aka tags anger var dynamisk html koden ska infogas i dokumentet och hur den ska genereras.

Förfrågningar i JSP:

När en klient begär JSP filen så tolkas den av servern som gör om den till en servlet om det är den första gången.

översättningen av en JSP fil:

Det som händer i en webbserver när en begäran anländer:

- begäran anländer till webbserver
- om det är förstagången som sidan begärs så laddas motsvarande JSP-fil in och kompileras om till en servlet av ett särskilt översättningsprogram. i annat fall går vi direkt till steg 4.
- den nyskapade servleten kompileras av servern och startas.
- metoden `service()` anropas av servern. metoden analyserar begäran och skapar html kod som svar på begäran.
- den producerade html koden laddas ner till klienten och webbbrowsern uppdateras.

9 Nätverksprogramering

9.1 Socket

`Java.net.Socket*`; `java.net.Socket` är en klass som skapar en end-to-end förbindelse mellan två användare.

```
String host;
int port;
try{
    Socket s = new Socket(host, port);

} catch (IOException e){ ...
} finally {
    socket.close();
} catch (IOException e) { ... }
```

1. Open a socket.
2. Open an input stream and output stream to the socket.
3. Read from and write to the stream according to the server's protocol.
4. Close the streams.
5. Close the socket.

MulticastSocket

9.2 InputStream

java.io.InputStream; Du läser från en inputstream och behandlar data som du läst från en inputstream. En InputStream är en abstrakt klass som är en superklass till alla klasser som läser information från en stream ⁴.

```
try{
URL u= new URL("http://www.apperture.se");
//Notera raderna nedan, detta skyddar InputStreamen från felanvändning och buggar samt ökar
InputStream in = u.openStream();
in = new BufferedInputStream(in);
InputStreamReader r = new InputStreamReader(in);

}
```

9.3 OutputStream

Du skriver till en OutputStream!

9.4 Trådar

En tråd i java är en definierad väg för att utföra ett program. De flesta program är skrivet med endast en "Tråd" men detta skapar problem när ett program försöker göra flera saker samtidigt.

För att skapa en tråd måste du göra en av följande:

- implementera interfacet Runnable.
- extend Thread.
- implements Callable

Fördelen med Runnable är att då kan du extenda en annan klass som du har gjort tidigare medan thread så kan du inte utöka en annan klass samtidigt. Runnable gör också att du kan skapa flera threads som refererar till samma objekt men thread fixar

9.4.1 Runnable

Är ett interface som definierar hur Runnable gör ditt program mer flexibelt.

⁴Tecken som är längre än en bit.

```

public class MyRunnable implements Runnable {
    public void run() {
        //Code
    }
}

```

9.4.2 Thread

```

public class MyThread extends Thread {
    public MyThread() {
        super("MyThread");
    }
    public void run() {
        //Code
    }
}

```

9.4.3 Callable

9.4.4 ClientThread

9.4.5 ServerThread

synchronized

Ömsesidig uteslutning innebär att endast en tråd kan vid varje tillfälle befinna sig inom en och samma s.k. kritiska region, dvs exekvera kod som skulle störa exekveringen av en annan tråds kod (t.ex. genom samtidig manipulering av data eller annan resurs).

Om en annan tråd försöker gå in i den kritiska regionen måste denna vänta tills den tråd som befinner sig inom regionen lämnar den.

Java tillhandahåller en mekanism som kallas objekt serialisering där ett objekt kan representeras som en sekvens av byte som innehåller objektets data samt information om objektets typ och vilka typer av data som lagras i objektet.

Efter en serie föremål har skrivits till en fil, kan det läsas från filen och deserialiseras det vill säga information och byte som representerar objektet och dess data kan användas för att återskapa objektet minnestyp.

Mest imponerande är att hela processen är JVM oberoende, vilket innebär att ett objekt kan serialiseras på en plattform och deserialiseras på en helt annan plattform.

10 Multicast

Ett objekt av typen `MulticastSocket` i Java kan användas för att skicka vanliga datagram (unicast) såväl som multicastdatagram. Det som styr huruvida det blir ett unicast- eller multicastmeddelande är endast IP-numret som angetts som meddelandets mottagare.