

JAC444 - BTP400 Course Object-Oriented Software Development II - Java

Principles of Network Programming in Java

Segment 1



Network Programming



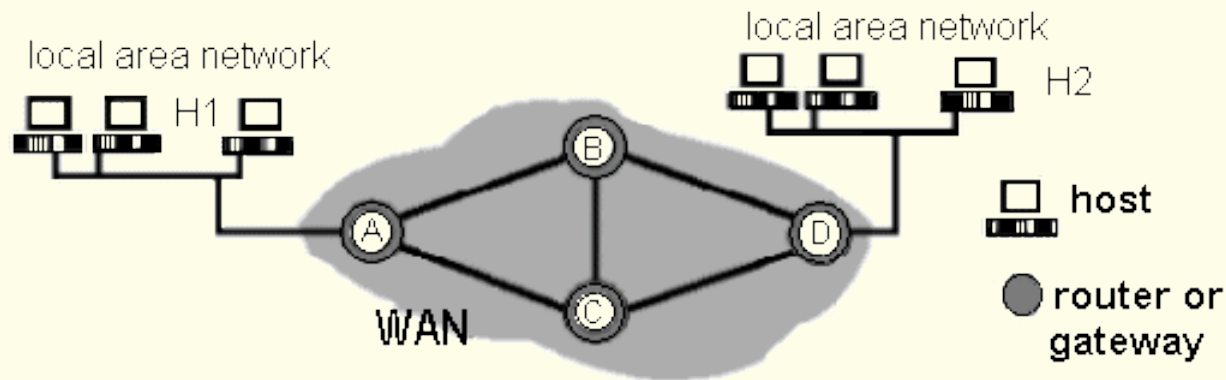
In this lesson you will be learning about:

- Layering and protocols
- Java Network Programming
- URL Processing

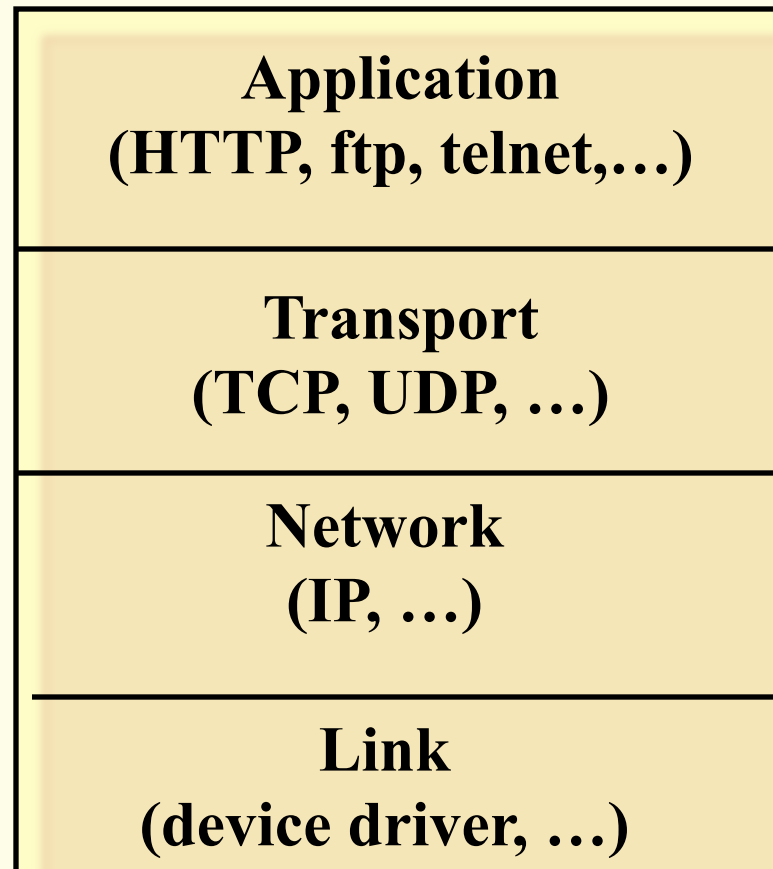


Elements of a Network

- Communication links:
 - point-to-point (e.g., A-to-B)
 - broadcast (e.g., Ethernet LAN)
- Host: computer running applications which use network (e.g.: H1)
- Router: computer routing packet from input line to output line. (e.g., C)
- Gateway: a router directly connects networks (e.g. A)



Networking Basics



java.net.InetAddress class

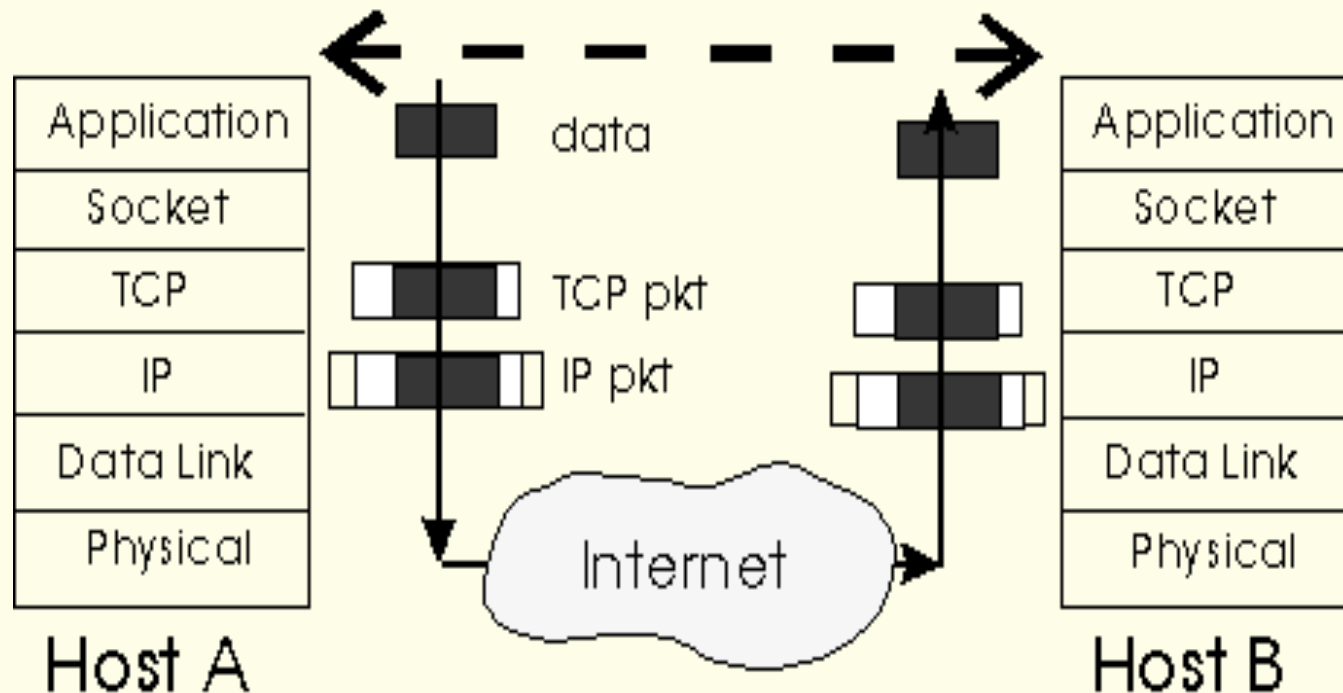


```
public static InetAddress getByName(String host)
    throws UnknownHostException
public static InetAddress[] getAllByName(String host)
    throws UnknownHostException
public static InetAddress getLocalHost()
    throws UnknownHostException
public boolean isMulticastAddress()
public String getHostName()
public byte[] getAddress()
public String getHostAddress()
public int hashCode()
public boolean equals(Object obj)
public String toString()
```



Protocol packets

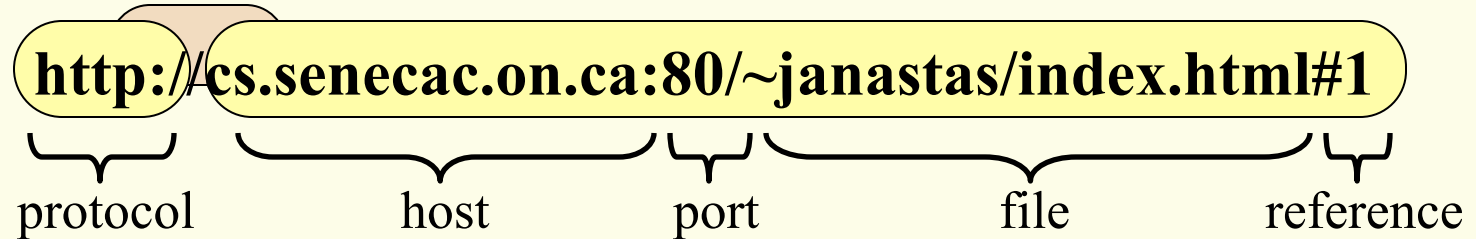
- Unit of data exchanged between protocol entities in a given layer.



Uniform Resource Locator



- URL is an acronym for *Uniform Resource Locator* and is a reference (an address) to a resource on the Internet.



```
import java.net.*; import java.io.*;
public class ParseURL {
    public static void main(String[] args) throws Exception {
        URL aURL = new URL("http://cs.senecac.on.ca:80/~janastas/index.html#1");
        System.out.println("protocol = " + aURL.getProtocol());
        System.out.println("host = " + aURL.getHost());
        System.out.println("filename = " + aURL.getFile());
        System.out.println("port = " + aURL.getPort());
        System.out.println("ref = " + aURL.getRef());
    }
}
```



Read URL Content



```
import java.net.*;
import java.io.*;
public class ReadSites {

    public static void main(String[] args) {

        for (int i = 0; i < args.length; i++) {
            try {
                URL u = new URL(args[i]);
                InputStream is = u.openStream();
                InputStreamReader isr = new InputStreamReader(is);
                BufferedReader br = new BufferedReader(isr);
                String theLine;
                while ((theLine = br.readLine()) != null) {
                    System.out.println(theLine);
                }
            } catch (MalformedURLException e) {
                System.err.println(e);
            } catch (IOException e) {
                System.err.println(e);
            }
        }
    }
}
```



Conclusion

After completion of this lesson you should know:

- How to write Java Programs using *java.net*.*
- Networking solution using Java packages.
- Client Server Paradigm.
- Socket, ServerSocket and URL Classes.

