THE GEEK STUFF

Linux | DB | Open Source | Web

- Home
- Free eBook
- Contact
- About
- Start Here

TCP/IP Protocol Fundamentals Explained with a Diagram

by Himanshu Arora on November 2, 2011

Have you ever wondered how your computer talks to other computers on your local LAN or to other systems on the internet?

Understanding the intricacies of how computers interact is an important part of networking and is of equal interest to a sysadmin as well as to a developer. In this article, we will make an attempt to discuss the concept of communication from the very basic fundamental level that needs to be understood by everybody.

TCP/IP PROTOCOL SUITE

Communications between computers on a network is done through protocol suits. The most widely used and most widely available protocol suite is TCP/IP protocol suite. A protocol suit consists of a layered architecture where each layer depicts some functionality which can be carried out by a protocol. Each layer usually has more than one protocol options to carry out the responsibility that the layer adheres to. TCP/IP is normally considered to be a 4 layer system. The 4 layers are as follows :

1. Application layer
2. Transport layer
3. Network layer
4. Data link layer

1. Application layer

This is the top layer of TCP/IP protocol suite. This layer includes applications or processes that use transport layer protocols to deliver the data to destination computers.

At each layer there are certain protocol options to carry out the task designated to that particular layer. So, application layer also has various protocols that applications use to communicate with the second layer, the transport layer. Some of the popular application layer protocols are :

- HTTP (Hypertext transfer protocol)
- FTP (File transfer protocol)
- SMTP (Simple mail transfer protocol)
- SNMP (Simple network management protocol) etc

2. Transport Layer

This layer provides backbone to data flow between two hosts. This layer receives data from the application layer above it. There are many protocols that work at this layer but the two most commonly used protocols at transport layer are TCP and UDP.

TCP is used where a reliable connection is required while UDP is used in case of unreliable connections.

TCP divides the data(coming from the application layer) into proper sized chunks and then passes these chunks onto the network. It acknowledges received packets, waits for the acknowledgments of the packets it sent and sets timeout to resend the packets if acknowledgements are not received in time. The term 'reliable connection' is used where it is not desired to loose any information that is being transferred over the network through this connection. So, the protocol used for this type of connection must provide the mechanism to achieve this desired characteristic. For example, while downloading a file, it is not desired to loose any information(bytes) as it may lead to corruption of downloaded content.

UDP provides a comparatively simpler but unreliable service by sending packets from one host to another. UDP does not take any extra measures to ensure that the data sent is received by the target host or not. The term 'unreliable connection' are used where loss of some information does not hamper the task being fulfilled through this connection. For example while streaming a video, loss of few bytes of information due to some reason is acceptable as this does not harm the user experience much.

3. Network Layer

This layer is also known as Internet layer. The main purpose of this layer is to organize or handle the movement of data on network. By movement of data, we generally mean routing of data over the network. The main protocol used at this layer is IP. While ICMP(used by popular 'ping' command) and IGMP are also used at this layer.

4. Data Link Layer

This layer is also known as network interface layer. This layer normally consists of device drivers in the OS and the network interface card attached to the system. Both the device drivers and the network interface card take care of the communication details with the media being used to transfer the data over the network. In most of the cases, this media is in the form of cables. Some of the famous protocols that are used at this layer include ARP(Address resolution protocol), PPP(Point to point protocol) etc.

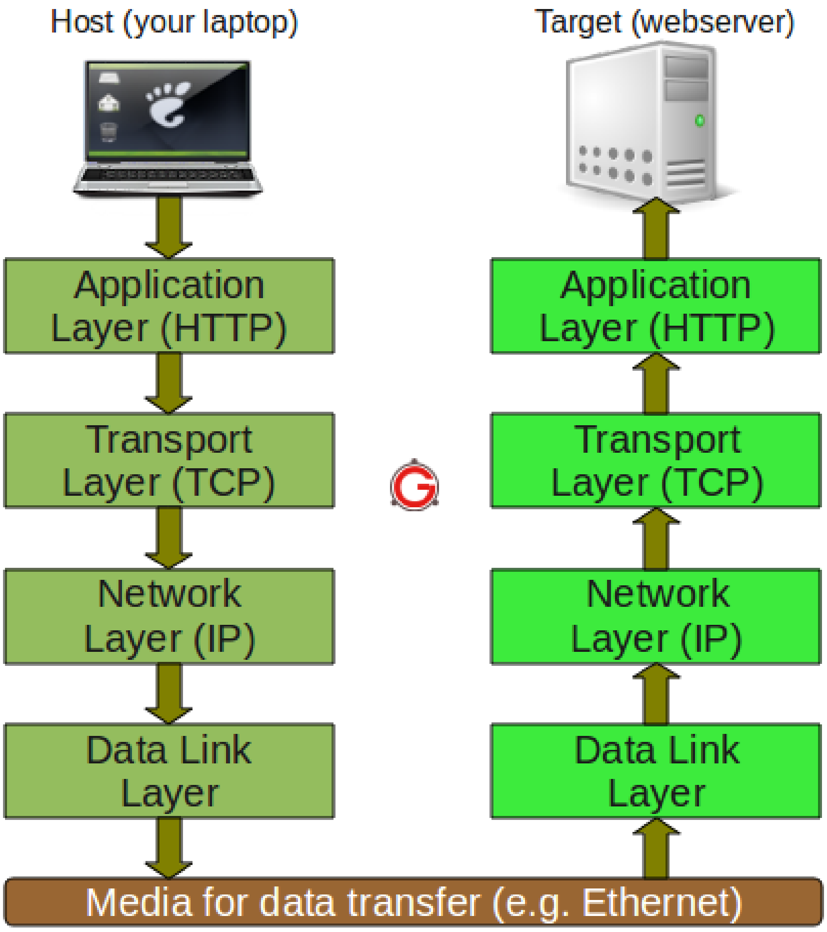
TCP/IP CONCEPT EXAMPLE

One thing which is worth taking note is that the interaction between two computers over the network through TCP/IP protocol suite takes place in the form of a client server architecture.

Client requests for a service while the server processes the request for client.

Now, since we have discussed the underlying layers which help that data flow from host to target over a network. Lets take a very simple example to make the concept more clear.

Consider the data flow when you open a website.



As seen in the above figure, the information flows downward through each layer on the host machine. At the first layer, since http protocol is being used, so an HTTP request is formed and sent to the transport layer.

Here the protocol TCP assigns some more information(like sequence number, source port number, destination port number etc) to the data coming from upper layer so that the communication remains reliable i.e, a track of sent data and received data could be maintained.

At the next lower layer, IP adds its own information over the data coming from transport layer. This information would help in packet travelling over the network. Lastly, the data link layer makes sure that the data transfer to/from the physical media is done properly. Here again the communication done at the data link layer can be reliable or unreliable.

This information travels on the physical media (like Ethernet) and reaches the target machine.

Now, at the target machine (which in our case is the machine at which the website is hosted) the same series of interactions happen, but in reverse order.

The packet is first received at the data link layer. At this layer the information (that was stuffed by the data link layer protocol of the host machine) is read and rest of the data is passed to the upper layer.

Similarly at the Network layer, the information set by the Network layer protocol of host machine is read and rest of the information is passed on the next upper layer. Same happens at the transport layer and finally the HTTP request sent by the host application(your browser) is received by the target application(Website server).

One would wonder what happens when information particular to each layer is read by the corresponding protocols at target machine or why is it required? Well, lets understand this by an example of TCP protocol present at transport layer. At the host machine this protocol adds information like sequence number to each packet sent by this layer.

At the target machine, when packet reaches at this layer, the TCP at this layer makes note of the sequence number of the packet and sends an acknowledgement (which is received seq number + 1).

Now, if the host TCP does not receive the acknowledgement within some specified time, it re sends the same packet. So this way TCP makes sure that no packet gets lost. So we see that protocol at every layer reads the information set by its counterpart to achieve the functionality of the layer it represents.

PORTS, SERVERS AND STANDARDS

On a particular machine, a port number coupled with the IP address of the machine is known as a socket. A combination of IP and port on both client and server is known as four tuple. This four tuple uniquely identifies a connection. In this section we will discuss how port numbers are chosen.

You already know that some of the very common services like FTP, telnet etc run on well known port numbers. While FTP server runs on port 21, Telent server runs on port 23. So, we see that some standard services that are provided by any implementation of TCP/IP have some standard ports on which they run. These standard port numbers are generally chosen from 1 to 1023. The well known ports are managed by Internet Assigned Numbers Authority(IANA).

While most standard servers (that are provided by the implementation of TCP/IP suite) run on standard port numbers, clients do not require any standard port to run on.

Client port numbers are known as ephemeral ports. By ephemeral we mean short lived. This is because a client may connect to server, do its work and then disconnect. So we used the term ‘short lived’ and hence no standard ports are required for them.

Also, since clients need to know the port numbers of the servers to connect to them, so most standard servers run on standard port numbers.

The ports reserved for clients generally range from 1024 to 5000. Port number higher than 5000 are reserved for those servers which are not standard or well known.

If we look at the file ‘/etc/services’, you will find most of the standard servers and the port on which they run.

```
$ cat /etc/services
systat          11/tcp          users
daytime         13/udp
netstat         15/tcp
qotd            17/tcp          quote
msp             18/udp
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp             21/tcp
ssh             22/tcp
ssh             22/udp
telnet          23/tcp
...
...
```

. . .

As you see from the /etc/services file, FTP has port number 21, telent has port number 23 etc. You can use 'grep' command on this file to find any server and its associated port.

As far as the standards are concerned, the following four organizations/groups manage the TCP/IP protocol suite. Both the IRTF and the IETF fall under the IAB.

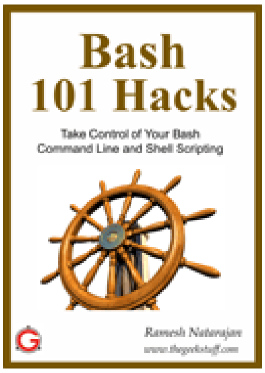
1. The Internet Society (ISOC)
2. The Internet Architecture Board (IAB). The IAB falls under the ISOC.
3. The Internet Engineering Task Force (IETF)
4. The Internet Research Task Force (IRTF)

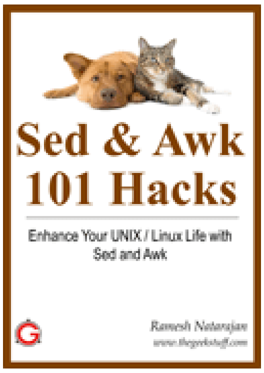
g+1 50 Tweet 28 Like 89 > Add your comment

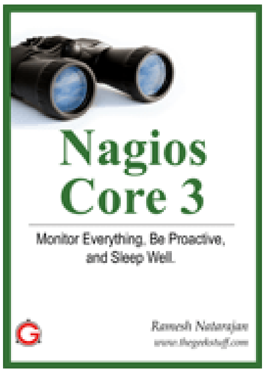
If you enjoyed this article, you might also like..

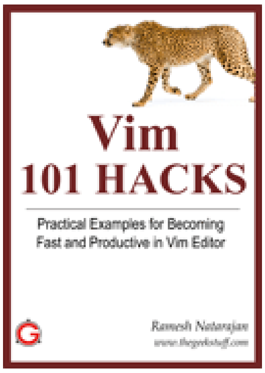
1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)
3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook Free

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples
- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons








Vim
101 HACKS
Practical Examples for Becoming Fast and Productive in Vim Editor
Ramesh Natarajan
www.thegeekstuff.com

{ 51 comments… read them below or add one }

1 Devin November 2, 2011 at 9:02 am

You left out the most popular application layer protocol – HTTP!

2 krishnan November 2, 2011 at 9:09 am

Excellent job. Short and informative

3 jalal hajigholamali November 2, 2011 at 12:10 pm

Hi,

very nice article
thanks a lot…

4 Júlio Hoffimann Mendes November 2, 2011 at 1:52 pm

Thank you for the clear explanation! I've always wanted to learn a little about TCP/IP. Keep the good work!

Regards,
Júlio.

5 Walyonaire November 2, 2011 at 3:17 pm

This reminds me of my Cisco Lessons, thanks for the "refresh".
Excellent Article.

6 Narendra Kangralkar November 2, 2011 at 10:21 pm

Very good explanation !!!
Keep writing.

7 Ramesh Natarajan November 2, 2011 at 10:27 pm

@Devin,

Yes. HTTP is the most popular application layer protocol. This is why the "TCP/IP Concept Example" section explains it using the HTTP protocol.

Anyway, thanks for pointing it out, HTTP is added to the "Application Layer" section to make the list complete.

8 Joe November 3, 2011 at 3:37 am

Thank you for the effort.

Well written.

9 Vonskippy November 3, 2011 at 2:54 pm

Seems redundant to the OSI model.

10 ATUL KHACHANE November 4, 2011 at 8:32 pm

Yes. Vonskippy 😉

Great articles here always……………this is one stop resource…..and you will find almost everything you need…

11 Himanshu November 5, 2011 at 12:48 pm

Thank you all for your valuable comments.

12 grouch0 November 13, 2011 at 9:15 pm

Some books say the 5000 upper limit for client (dynamic/ephemeral) ports was a misprint in W. Richard Stevens' "TCP/IP Illustrated Vol. 1″ and should have been 50000. The issue doesn't seem to have affected the progress of the protocol in taking over the world.

13 Himanshu November 13, 2011 at 11:30 pm

@grouch0
OK. Can you provide some references?

14 Devin November 28, 2011 at 8:37 pm

You forgot the physical layer.

15 Himanshu November 29, 2011 at 12:08 am

Well, I think you are talking about the OSI model where physical layer is the last layer. In TCP/IP model Data link layer is the layer 1. Sometimes this layer is also referred to as "Network Interface Layer" as as to avoid confusion with the data link layer of the OSI model. This Network interface Layer encapsulates physical layer also.

16 Sravan December 15, 2011 at 2:48 am

Thanks. Short and informative article.

17 deepak kumar January 3, 2012 at 2:23 pm

nice …..very helpfull
thnx

18 Reece April 17, 2012 at 5:04 am

thanks for the information, it helped me understand TCP/IP

19 Dharmesh May 13, 2012 at 9:42 pm

Thanks a lot!!!

It was a very short and good explanation…

20 Asif May 13, 2012 at 10:08 pm

Simple nd Superbbbbbbbbbbb

21 rohan June 16, 2012 at 1:15 am

an application uses ephemeral port and it uses http as application layer protocol. Now when the packet is sent to destination, will it contain both ephemeral destination port and standard http destination port no. How this packet will be handled by transport layer and application layer at destination?

22 Tariq August 9, 2012 at 3:02 pm

Nice.

A small demo with
wireshark or any other tool could be used to show "how" this happens in real time.

23 raju September 13, 2012 at 8:54 am

Thanks a pile
It's fabulous artical and very knowledgeable for me,

24 malik usman October 3, 2012 at 9:15 am

Thanks a lot!!!

It was a very short and good explanation…

25 swathi November 24, 2012 at 10:13 am

Its is short but soooo…… informative,thank u very much

26 Ritu Kushwaha January 26, 2013 at 11:12 am

Useful Information in Short. Well Done!!
Keep Writing…

27 Umar Badu February 7, 2013 at 10:28 am

Useful information,i really appreciate this……Thanks

28 poonguzhali February 14, 2013 at 7:37 am

Its very easy to understand.

29 Nagendra Bhat February 20, 2013 at 8:20 pm

Hi..
Explanation is good and if there any blogs related to this subject please mention it or please mail me..

30 monica March 15, 2013 at 3:10 am

Very well explained in simple language

31 pemo May 13, 2013 at 2:13 am

nice thanks a lot…

32 Ameen June 12, 2013 at 9:49 am

tanx 4 ur explanation.

33 anand July 31, 2013 at 12:43 am

very good to understand

34 brig (retd) ashwani sharma August 20, 2013 at 9:47 am

An extremely clear, concise and lucid explanation of TCP/IP.
Great effort. Keep it up.

Best Wishes

35 hana September 2, 2013 at 12:05 am

good understand
thanks

36 khanabusd September 28, 2013 at 6:00 am

very helpful………………………………

37 zug October 16, 2013 at 10:29 pm

We'll written in a language that I find easy to read and understand. Thank you.

38 justauser October 27, 2013 at 2:41 pm

great article! the perfect balance of details and general view 🙂
thanks!

39 maurice November 7, 2013 at 2:51 am

it very helpful…….in simple language

40 jaco November 8, 2013 at 9:00 pm

Thank u
u manage to save me
as i don't yet have a text book to read
this gave light into my understanding of TCP/IP im able to
peruse my dreams 🙂

41 Mehbub November 19, 2013 at 6:00 am

very good explanation,u save me also……………………….thank u vry much

42 arun mehta December 8, 2013 at 8:28 pm

thanks

43 Mintu Krishna December 12, 2013 at 12:09 pm

very nice, in short & informative explanation. Useful for those who have keen interest in Internet.
Thanks a lot.

44 Ben January 24, 2014 at 5:57 am

I found this very useful! Thank you very much 🙂

45 Karm Nath February 12, 2014 at 3:23 am

good explanation …….

46 joe March 19, 2014 at 2:22 am

thanks for this article. Quite good.

47 amar March 24, 2014 at 3:15 am

Hi,
Explanations are very easy to read and understand. I would like to know more depth about encapsulation and decapsulation process of pkts in source side and destination side.

Thanks

48 kamal April 9, 2014 at 9:29 am

I really found useful article coz…before the exam usually I don't read the book I just look and search the info…in the net….

49 ravi June 5, 2014 at 6:25 am

short n sweet 😃

50 kul sidhu June 10, 2014 at 8:11 am

Plz give difference between osi and tcp/ip with diagram

51 Asamoah Apau Daniel August 21, 2014 at 6:55 am

1. what is a transport protocol?
a. explain with diagrams different between TCP and UDP
2. Explain secure Protocol
a. explain with diagrams different between FTP and SSH

Leave a Comment

Name

E-mail

Website

□ Notify me of followup comments via e-mail

Submit

Previous post: 20 OpenOffice Calc Spreadsheet Tips and Tricks

Next post: Connect with The Geek Stuff

- RSS | Email | Twitter | Facebook | Google+

Search

-

- **COURSE**
  - Linux Sysadmin CentOS 6 Course - Master the Tools, Configure it Right, and be Lazy
- **EBOOKS**
  - Free Linux 101 Hacks 2nd Edition eBook - Practical Examples to Build a Strong Foundation in Linux
  - Bash 101 Hacks eBook - Take Control of Your Bash Command Line and Shell Scripting
  - Sed and Awk 101 Hacks eBook - Enhance Your UNIX / Linux Life with Sed and Awk
  - Vim 101 Hacks eBook - Practical Examples for Becoming Fast and Productive in Vim Editor
  - Nagios Core 3 eBook - Monitor Everything, Be Proactive, and Sleep Well
-

  The Geek Stuff

  Like

  7,482 people like The Geek Stuff.

  Facebook social plugin
- **POPULAR POSTS**
  - 12 Amazing and Essential Linux Books To Enrich Your Brain and Library
  - 50 UNIX / Linux Sysadmin Tutorials
  - 50 Most Frequently Used UNIX / Linux Commands (With Examples)
  - How To Be Productive and Get Things Done Using GTD
  - 30 Things To Do When you are Bored and have a Computer
  - Linux Directory Structure (File System Structure) Explained with Examples
  - Linux Crontab: 15 Awesome Cron Job Examples
  - Get a Grip on the Grep! – 15 Practical Grep Command Examples
  - Unix LS Command: 15 Practical Examples
  - 15 Examples To Master Linux Command Line History
  - Top 10 Open Source Bug Tracking System
  - Vi and Vim Macro Tutorial: How To Record and Play

- Mommy, I found it! -- 15 Practical Linux Find Command Examples
- 15 Awesome Gmail Tips and Tricks
- 15 Awesome Google Search Tips and Tricks
- RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams
- Can You Top This? 15 Practical Linux Top Command Examples
- Top 5 Best System Monitoring Tools
- Top 5 Best Linux OS Distributions
- How To Monitor Remote Linux Host using Nagios 3.0
- Awk Introduction Tutorial – 7 Awk Print Examples
- How to Backup Linux? 15 rsync Command Examples
- The Ultimate Wget Download Guide With 15 Awesome Examples
- Top 5 Best Linux Text Editors
- Packet Analyzer: 15 TCPDUMP Command Examples
- The Ultimate Bash Array Tutorial with 15 Examples
- 3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id
- Unix Sed Tutorial: Advanced Sed Substitution Examples
- UNIX / Linux: 10 Netstat Command Examples
- The Ultimate Guide for Creating Strong Passwords
- 6 Steps to Secure Your Home Wireless Network
- Turbocharge PuTTY with 12 Powerful Add-Ons

CATEGORIES

- Linux Tutorials
- Vim Editor
- Sed Scripting
- Awk Scripting
- Bash Shell Scripting
- Nagios Monitoring
- OpenSSH
- IPTables Firewall
- Apache Web Server
- MySQL Database
- Perl Programming
- Google Tutorials
- Ubuntu Tutorials
- PostgreSQL DB
- Hello World Examples
- C Programming
- C++ Programming
- DELL Server Tutorials
- Oracle Database
- VMware Tutorials

Ramesh Natarajan

Follow

About The Geek Stuff

My name is **Ramesh Natarajan**. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about Ramesh Natarajan and the blog.

- **Support Us**

Support this blog by purchasing one of my ebooks.

Bash 101 Hacks eBook

Sed and Awk 101 Hacks eBook

Vim 101 Hacks eBook

Nagios Core 3 eBook

- **Contact Us**

Email Me : Use this Contact Form to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

Follow us on Google+

Follow us on Twitter

Become a fan on Facebook