Real-Time Network Monitoring with Iftop

Traffic Watch

The iftop utility looks simple, but this versatile tool can provide a wealth of network monitoring information. *By Saikat Goswami*

In today's interconnected world, network monitoring has become an essential skill. With the exponential growth of cloud computing, IoT devices, and remote work solutions, understanding network traffic patterns is more critical than ever before. Among the plethora of Linux monitoring tools available, iftop stands out as one of the most powerful and versatile command-line utilities for real-time network bandwidth monitoring. Unlike system monitoring tools like htop or atop that focus on CPU, memory, and process metrics, iftop provides unparalleled visibility into your network traffic patterns, helping you identify bandwidth hogs, troubleshoot performance issues, and optimize your network configuration.

Whether you're managing enterprise servers, optimizing application performance, or simply curious about network activity, mastering iftop will give you valuable insights into your system's network behavior.

Understanding iftop

iftop is a console-based network bandwidth monitoring tool that displays real-time bandwidth usage on individual network interfaces. What sets iftop apart from basic network monitors is its ability to provide a granular view of network activity at the connection level. Most administrators are familiar with tools that show overall bandwidth consumption, but iftop takes this several steps further by revealing exactly which hosts are communicating and how much bandwidth each connection is consuming.

Key advantages of iftop include:

- Connection-level visibility – Shows bandwidth usage per socket connection rather than just aggregate interface statistics
- Real-time monitoring – Updates the display continuously to reflect current network activity
- Protocol identification – Can distinguish between different types of network traffic
- Interactive interface – Allows sorting and filtering of connections while running
- Minimal resource usage – Lightweight compared to GUI-based monitoring tools
- Historical context – Provides short-term bandwidth usage trends through a multi-timeframe display

Under the hood, iftop uses the pcap library to capture packets moving through the network interface. As packets flow through the interface, iftop analyzes them to determine:

- Source and destination IP addresses
- Port numbers
- Protocols (TCP/UDP)
- Data transfer rates in both directions
- Total data transferred per connection

This approach lets iftop show bandwidth usage at the connection level, something that tools reading from `/proc/net/dev` (like nload) cannot provide. The tool maintains running averages of bandwidth usage across three different timescales (2, 10, and 40 seconds by default), giving you both immediate and trending views of your network activity.

Iftop is useful for answering questions like:

- Which external hosts is my server communicating with right now?
- What specific connections are consuming the most bandwidth at this moment?
- Is there any suspicious network activity I should investigate immediately?
- Which services or ports are generating the most traffic on my system?

See the box entitled "The Other Tools" for a look at how iftop compares with other monitoring utilities.

The Other Tools

Understanding how iftop compares to other tools will help you determine when to choose it over another solution:

- vs nload – Although nload shows overall interface statistics with simple graphs, it lacks the connection details that make iftop so powerful for troubleshooting.
- vs nethogs – nethogs shows bandwidth per process but doesn't show the network connection details that iftop provides.
- vs vnstat – vnstat maintains historical usage data but doesn't provide the real-time connection monitoring that iftop specializes in.
- vs bmon – bmon provides interface statistics and graphs but not the per-connection breakdowns that iftop offers.
- vs iptraf – Although iptraf offers similar functionality, iftop provides a cleaner, more focused interface for bandwidth monitoring.

Installing iftop

One of iftop's advantages is its wide availability across different Linux distributions. The installation process is straightforward, though it might vary slightly depending on your package manager. For Debian or Ubuntu-based systems, enter

```
sudo apt update
sudo apt install iftop
```

For RHEL or CentOS:

```
sudo yum install epel-release -y
sudo yum install iftop -y
```

For Fedora:

```
sudo dnf install iftop -y
```

For Arch Linux:

```
sudo pacman -Syu iftop
```

For other distros, see your package manager's documentation.

You'll need root privileges to run iftop, because it needs access to network interfaces for packet capture:

```
sudo iftop
```

iftop offers several useful command-line options to customize its behavior right from startup. These options allow you to tailor the tool's operation to your specific monitoring needs without having to interact with the interface. For instance, use the `-i` option to specify which network interface to monitor (e.g., eth0, wlan0, ens3).

Use the `-N` option to specify port numbers or the `-P` option to show both port numbers and IP addresses. A lowercase `-n` tells iftop to disable hostmane resolution, which will speed up processesing by reducing DNS traffic. See the box entitled "More Options" for other useful command-line switches.

For example, to monitor interface `eth0` without hostname resolution and showing port numbers:

```
sudo iftop -i eth0 -nP
```

Or to monitor only HTTP/HTTPS traffic on interface wlp3s0:

```
sudo iftop -i wlp3s0 -f ➥
  'port 80 or port 443'
```

Understanding the iftop Interface

When launched, iftop presents a clean, text-based interface divided into logical sections that work together to provide a comprehensive view of network activity (Figure 1).

The top section shows total bandwidth usage. The middle section lists active connections (source to destination) with bandwidth usage. The bottom section shows RX and TX. RX (Receive) represents incoming traffic, measures the amount of data your machine is receiving from other hosts (in bytes per second). TX (Transmit) represents outgoing

More Options

Other useful command-line options include:

- `-B` – Displays bandwidth in bytes instead of bits (more intuitive for many users)
- `-F net/mask` – Filters to show only traffic to/from a specified network (CIDR notation)
- `-f filter` – Applies a BPF filter expression to monitor specific traffic
- `-t` – Uses the text interface without ncurses (useful for scripting)

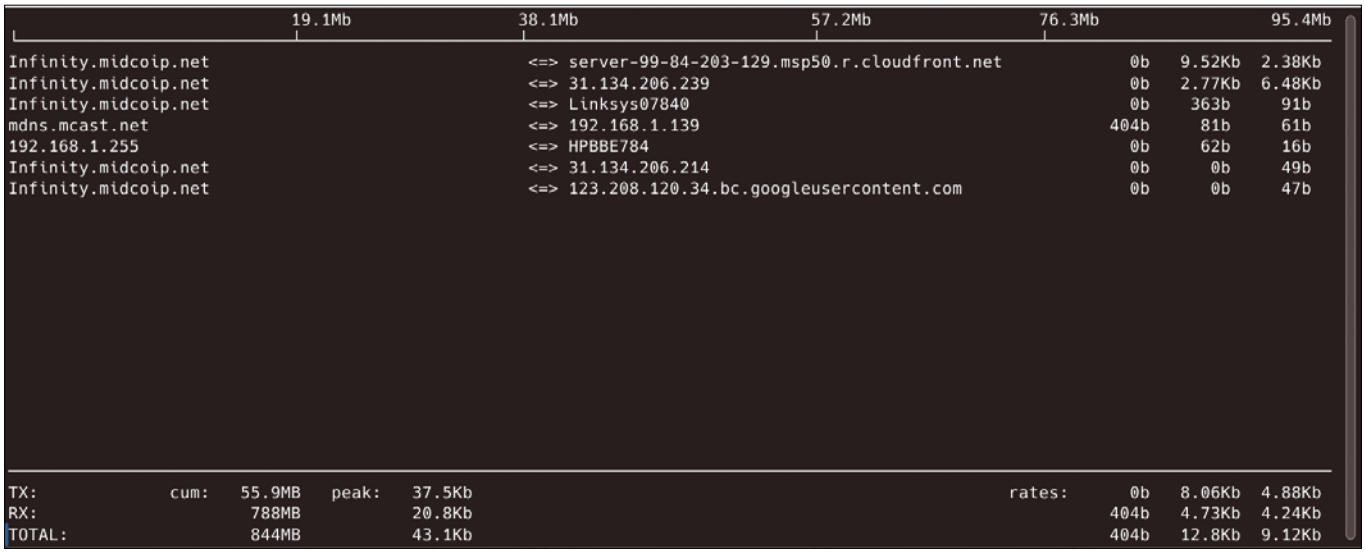

Figure 1: Iftop's text-based interface offers a view of current connections and overall network activity.

traffic, measures the amount of data your machine is sending to other hosts (in bytes per second).

The connection list in the middle section is the most important part, showing for each connection:

- Source host (left side, with port if enabled)
- Destination host (right side, with port if enabled)
- Current bandwidth usage (middle bar, length represents utilization)
- 2-second average bandwidth rate (first numerical column)
- 10-second average bandwidth rate (second numerical column)
- 40-second average bandwidth rate (third numerical column)

By default, connections are sorted by their 40-second average bandwidth usage, with the heaviest connections at the top of the list. This default sorting helps immediately identify which connections are consuming the most bandwidth over a meaningful time frame.

Interactive Controls

Iftop becomes particularly powerful when you use its interactive controls to customize the display in real-time. These keyboard commands allow you to adapt the view to your immediate troubleshooting needs without restarting the tool. Type a keyboard key and watch the display react. You can toggle the port number display (`p`), display or hide the source address (`s`) or destination address (`d`), or display total bandwidth (`t`). Type `n` to toggle hostname resolution, which is helpful for reducing DNS lookups. Type `P` to freeze the current view and pause the display. Enter a 1, 2, or 3 to sort columns 1, 2, or 3 of the bandwidth timescales. The `l` option lets you enter a BPF filter expression.

Iftop's controls allow you to quickly focus on the most relevant connections for your current troubleshooting needs. For example, when diagnosing a bandwidth saturation issue, you might start with the default view to identify the heaviest flows, then press `S` to group by source if a particular host seems responsible. From there, press `p` to show the ports in order to identify the services responsible. Enter `j/k` to scroll through all connections from the host.

Filtering Traffic with BPF Expressions

One of iftop's most powerful features is its ability to filter traffic using Berkeley Packet Filter (BPF) expressions. This packet filtering language allows you to focus on specific types of traffic while excluding irrelevant data from the display. BPF filters can be applied either at startup via command line or interactively while iftop is running.

Common Filtering Scenarios

Common filtering scenarios include view-only HTTP traffic (port 80):

```
sudo iftop -f 'port 80'
```

monitor traffic to/from a specific host:

```
sudo iftop -f 'host 192.168.1.100'
```

exclude SSH traffic (port 22) from display:

```
sudo iftop -f 'not port 22'
```

monitor traffic between specific subnets:

```
sudo iftop -f 'net 192.168.1.0/24 ➥
  and net 10.0.0.0/8'
```

and view-only UDP traffic (useful for VoIP or streaming analysis):

```
sudo iftop -f 'udp'
```

You can also apply these filters interactively by pressing `l` and entering the filter expression. This allows you to quickly change what you're monitoring based on what you see in the initial display. For example, you might start with a broad view to identify interesting traffic. If you notice heavy traffic on port 443, press `l` and enter port 443 to focus just on HTTPS traffic. Then press `S` to sort by source to see which hosts are generating this traffic.

The ability to dynamically apply these filters makes iftop exceptionally flexible for drilling down into network issues.

Measuring Network Speed Between Hosts

While iftop is primarily a monitoring tool, creative users can leverage it to measure actual network speeds between specific hosts. The basic approach is to observe the bandwidth usage during a file transfer or other network operation and compare the measured rate against expected network capacity.

For example, to test upload speed to a specific server, you could initiate a large file upload to the target server and then enter

```
sudo iftop -i eth0 -f ➥
  'host target_server'
```

Then observe the outbound bandwidth rate during the transfer. This technique can help identify bottlenecks or confirm that you're getting the expected throughput from your network infrastructure. It's particularly valuable for troubleshooting speed issues where the theoretical network capacity doesn't match actual observed performance.

Identifying Bandwidth Hogs

One of the most common and valuable uses for iftop is identifying processes or connections consuming excessive bandwidth. On busy systems or networks, unexplained bandwidth usage can lead to performance degradation, quota overages, or even service disruptions. The cause for the excessive usage could be an unexpected backup, unauthorized media streaming, or even malware activity. Iftop provides the visibility needed to track down these bandwidth hogs.

If you notice slow network performance or unexpectedly high bandwidth usage, launch iftop and sort by bandwidth usage. You can identify the top consumers by examining the connection list. Then cross-reference the result with other tools like nethogs or lsof to zero in on the offending process.

Troubleshooting Network Performance Issues

Iftop serves as an excellent first-line tool for diagnosing various network performance problems. Its real-time connection view helps correlate user-reported issues with actual network behavior. You can use iftop to uncover uneven traffic distribution, troubleshoot unexpected traffic patterns, and diagnose protocol issues.

By observing traffic patterns during periods of poor performance, you can often pinpoint the root cause much

faster than with trial-and-error approaches. The ability to see both the big picture of overall bandwidth usage and the micro view of individual connections makes iftop uniquely suited for this kind of troubleshooting.

Monitoring for Suspicious Activity

Security-conscious administrators can use iftop as part of their basic security monitoring toolkit. Iftop is not a replacement for dedicated security tools, but you can use it to help identify several types of suspicious activity, including:

- Unexpected external connections – Detect calls to unknown external IPs that might indicate malware or data exfiltration
- Port scanning activity – Identify multiple rapid connections to different ports on your host
- Data exfiltration attempts – Spot large outbound transfers to unexpected destinations
- DDoS participation – Notice many connections to a single external service
- Unauthorized services – Find unexpected listening ports receiving connections

Run iftop with appropriate filters to exclude known-good traffic, then look for connections to or from unexpected IP addresses or countries. Look for unusually high bandwidth to any single host, and watch for patterns that suggest scanning (many ports to one host). Investigate any suspicious findings with deeper tools.

Although iftop won't provide complete security visibility, it serves as an excellent "canary in the coal mine" to alert you to potential issues worth deeper investigation with tools like tcpdump, Wireshark, or a security information and event management (SIEM) system.

Verifying Network Configuration Changes

If you make a change to your network, iftop can provide immediate feedback to verify the result. This real-time validation is invaluable for ensuring changes have the intended effect without unexpected side effects. You can use iftop to confirm that the intended traffic is allowed or blocked following a change to firewall rules. Iftop also lets you verify bandwidth limits and prioritization for QoS implementations. If you make a change to a router, iftop will let you ensure that the traffic is flowing as intended.

For example, after implementing QoS to prioritize VoIP traffic, you could run iftop with the focus on common SIP/RTP ports:

```
iftop -f 'port 5060 or ➥
  portrange 10000-20000'
```

Then initiate test calls while generating background traffic and verify that VoIP traffic maintains priority during congestion. This immediate feedback loop helps avoid situations where a configuration error becomes apparent at a critical moment.

Configuration File Options

Although iftop works well with default settings, power users can customize its behavior through a configuration file, typically located at `~/.iftoprc`. This file allows persistent customization that applies across all iftop sessions without needing command-line options. See the iftop manpage for more on iftop configuration file settings.

Conclusion

The iftop utility is far more than a simple bandwidth monitor – it's a real-time window into network activity, empowering administrators to understand, diagnose, and optimize network performance with minimal setup and zero packet capture overhead. This article explored how you can use iftop for deep network insight and proactive management.

As network environments grow increasingly complex – spanning the cloud, containers, and virtualized systems – the need for simple command-line tools that provide real-time situational awareness remains vital. Iftop continues to prove its value, balancing simplicity with depth and speed with precision.

Whether you're troubleshooting a slow VPN, confirming QoS policies, or hunting for unusual traffic patterns, iftop gives you the ability to see what's really happening right now. For any sys admin, DevOps engineer, or network specialist, iftop remains one of the most practical and indispensable tools in the Linux networking toolkit. ■■■

Author

Saikat Goswami has more than 20 years in the IT industry, where he has worked on Linux, Java, and related technologies. He holds a master's in Industrial Engineering from Clemson University. Like everyone, he is fond of Linux and always wants to write on it. When he is not in his front of laptop, he is strumming his guitar or maybe travelling to mountains.