

# Data Routers

Just as *network* routers provide flexibility for connecting computers, a **data router provides flexibility between your real-time data sources and the tools that process the real-time data.**

You wouldn't hard-wire every workstation to every server... (You'd run out of space and money.) Instead, your network hubs, routers and switches provide efficiency and flexibility.

For similar reasons, you wouldn't want to hard-wire tools to your data sources (traps, logs, probes, agents, database queries, etc.). Doing so could bind your real-time data to only one use, and one vendor's solution.

## Why unbind your real-time data?

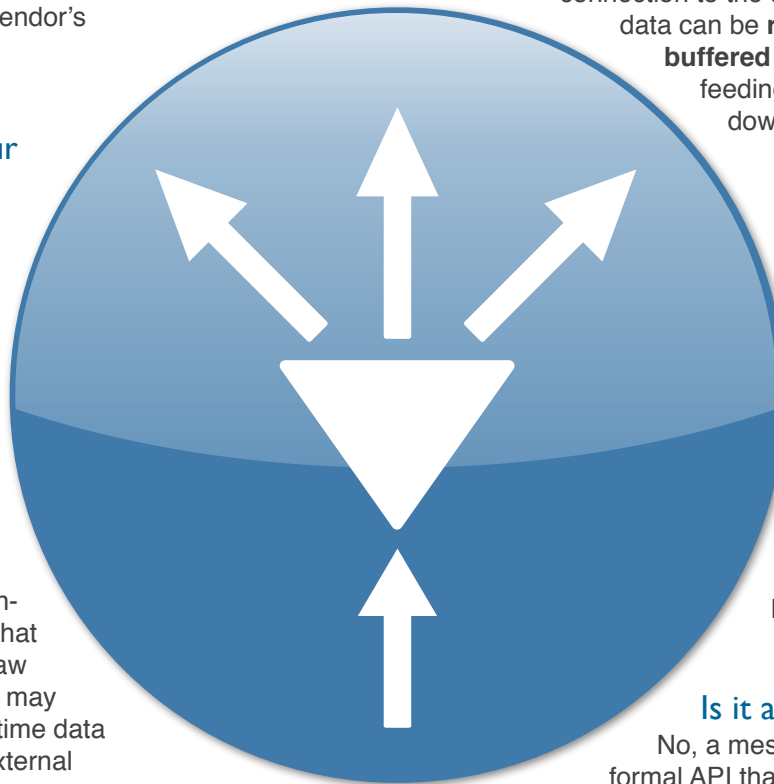
There are two main reasons:

- ① You may want to send the live raw data to multiple destinations. For example, you may have commercial management tools, custom tools, and in-house report tools that all need real-time raw data feeds. Or you may need to share real-time data with one or more external partners or customers.
- ② You can run alternative products in parallel, for breadth of features, or competition. **There's no better way to get fair vendor pricing than to regularly test their competition's products too.** And since you can test with your own live data, you are guaranteed to make meaningful comparisons, and discover bugs during testing rather than in production. Cutting over to a new product is less stressful, and a very real threat to ensure you always have your vendors' attention.

## What are your options?

You can't always just add more connections directly to your data sources. Depending on the source, its performance may bog down under the extra load, or there may be limitations due to protocols, interfaces, or your network capacity.

If you attach a data router as the single connection to the source, then the data can be **replicated** and **buffered** by the router for feeding any number of downstream listeners.



## Security

Access security for all your data is uniformly configured in a data router. Having it all in one place streamlines the administration effort, and helps avoid omissions or human errors.

## Is it a message bus?

No, a message bus has a formal API that both the data source and receiver must support. And the data must be encapsulated as individual messages.

A data router is entirely transparent, and therefore easier to deploy. It sees your data as just a continuous stream of bytes. The data router connects to your data source via its own native interface. At the other end, your data processors receive copies of the raw data stream, unaltered and real-time.

## Ubiquitous Access

If you have more than one type of data source, then you probably have more than one type of communications interface: SNMP, network sockets, serial ports, database queries, telnet/login/command, log files, and maybe even your own custom interfaces. Your downstream processors may not be equipped to easily handle all the possibilities you face.

A data router is designed to connect to anything. It usually comes equipped for connections to all common protocols, and will likely support a plug-in API to extend its functionality. More importantly, it streams the live data via a ubiquitous interface, usually a network socket. So connecting your downstream processors is simplified. Usually, your data processors just open a socket in order to start receiving data.

Administrators will appreciate the simplicity, and you can save money and complexity by avoiding a swath of protocol licenses from your vendors.

## Terminology & Architecture

A data router manages each data source as a *channel*. Your data processing tools are usually connected as *listeners*. You can usually configure access controls and limit the number of simultaneous listeners per channel

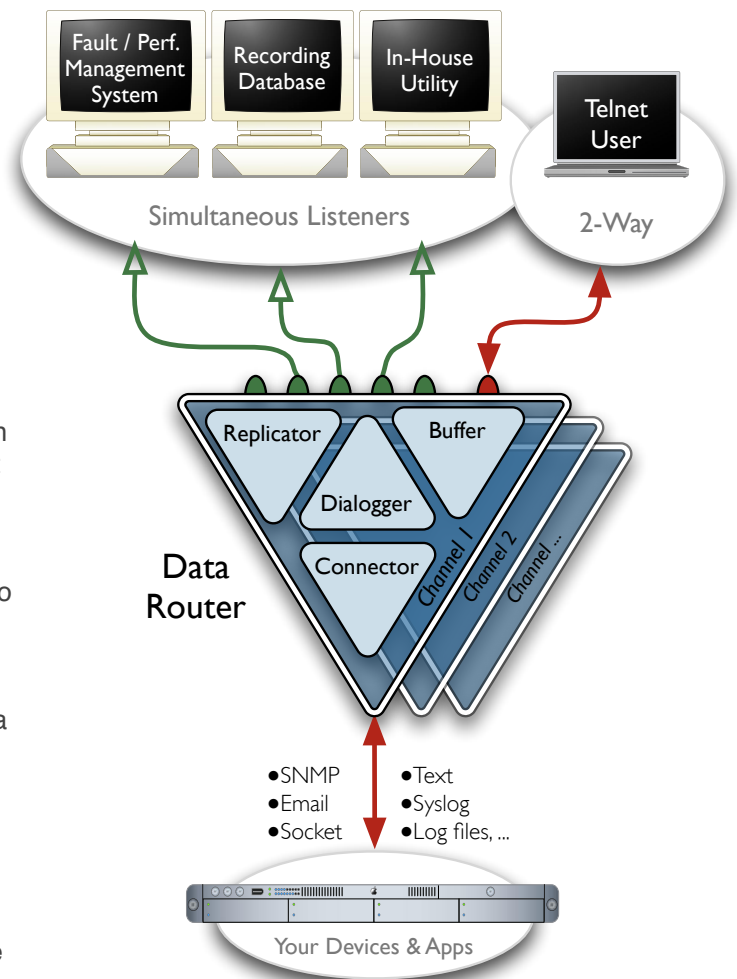
As the name implies, a listener can only read data. However, some channels may also support a *2-way* interactive session. A *dialogger* can support scripted interactions. A live interactive client is called a *talker*. Usually each channel will assign separate socket *port* numbers for connections from listeners and talkers.

For example, if a channel is connected to a console port on your equipment, a talker can send commands, and then both the talker and any listeners will simultaneously see the results of the commands.

A single data router can typically manage hundreds of channels and thousands of listeners and talkers simultaneously.

## Central Logging

A data router can record all the raw data that passes through it. Talker logs can be specially annotated for audits. This centralized logging



provides a convenient resource for log reporting tools.

## How do you recover after an outage?

You may not have to. A data router isolates your listeners from your data sources. If a data source fails, the data router periodically tries to reconnect. Meanwhile, any listeners stay connected to the channel, and therefore you avoid a domino effect of restarts.

## Socket Station™

Augur Systems develops the Socket Station brand of data router. It is a software solution that runs on any modern operating system. Socket Station currently manages thousands of data channels and listeners for its customers. Download the free trial at: <http://www.AugurSystems.com/>