

OVERCOMING THE DATA-SCALE MANAGEMENT CHALLENGE IN LARGE, DYNAMIC NETWORKS

Dwight Barker
Director of Product Management
Network Physics Inc.

ABSTRACT

A large military network may deliver hundreds of applications to thousands of users across many tens of thousands of devices, under conditions of constant and unpredictable change. This poses insuperable problems for traditional network management systems, which attempt to assemble a coherent view of infrastructure operations and their impact on application performance and business processes by aggregating data from thousands of infrastructure elements, agents, and probes into a monolithic master database. Such an approach cannot address the fundamental data-scale management challenge posed by large, dynamic networks: making sense of the sheer volume of data that must be collected, aggregated, and stored to manage such networks; and there are significant additional obstacles to success. These problems can be overcome with a distributed, flow-based approach using techniques adapted from high-energy physics research to address the data-scale management challenge.

INTRODUCTION

The overseas deployment of just one Army division is the equivalent of constructing a Fortune-500 corporation in weeks instead of years. How does one design, implement, and manage the network for such an effort, in an environment that is more demanding and dynamic than any other conceivable?

That question was answered starting more than a quarter-century ago. As is well known, one goal of the ARPANET project, the genesis of the Internet and, indeed, all modern, large-scale networks, was to create a network that could survive the exigencies of a nuclear war. What emerged was a set of distributed technologies for linking disparate computer systems via a decentralized packet-switched network.

Generally referred to (somewhat redundantly) as the TCP/IP protocol suite, these technologies support conversations (data flows), based on an agreed-upon set of higher-level protocols (an application), between a server offering a resource and a client using that resource, across any number of nominally independent but connected networks.

Under TCP/IP, as long as each participating independent network follows a set of relatively simple rules, any two computer systems can communicate across the largest internetworks, even if some devices or networks fail, and even in the face of massive changes. The scalability and resilience of the Internet, perhaps the largest structure ever created by the human race, is a testament to the success of this model and of the ARPANET project. The advantages of this networking model for military networks have not, of course, been overlooked.

THE DATA-SCALE MANAGEMENT CHALLENGE

Unfortunately, the technologies for managing the individual networks of a TCP/IP internetwork have not kept up with either the pace of application innovation or the growth in the size and extent of networks delivered by the “dumb network” model. The fundamental protocols of TCP/IP network management, such as the Simple Network Management Protocol (SNMP) and Remote Network Monitoring (RMON, now RMON2), still reveal the DNA of the small, isolated networks that ARPANET merged to become the Internet.

Such networks, consisting of a limited number of easily-identified devices, could be centrally managed—indeed, were. But management systems based on SNMP and RMON2, which depend on data from agents in individual network elements or from network probes reporting to a centralized database, do not scale well. As the number of devices grows, the amount of information that must be collected, aggregated, transmitted, and analyzed grows non-linearly, as does the management burden of keeping the system up to date with network changes. The network load imposed by this data, the computational load on the central database, and the difficulty of managing thousands of agents and probes, combine to make it practically impossible to develop a real-time¹ overview of a large network. This is the “data-scale management challenge.”

This challenge is similar to that faced by high-energy particle physicists managing the data from particle

¹ For the purposes of this paper, “real-time” is defined as one-minute granularity.

collision experiments, which may generate hundreds of terabytes of data in a fraction of a second. Algorithmic approaches developed to aggregate and top that data in accordance with the nature of the problems addresses can condense it up to two orders of magnitude without losing salient information. These approaches can be adapted to deal in similar fashion with the flood of data from a large, dynamic TCP/IP network. However, this requires the use of flow-correlated data in a distributed management system, rather than device and probe-derived data in a centralized system.

Consider a large, globally-distributed network with 20 regional data aggregation and control centers. If each center generates 250 Mbits/sec, the aggregate data rate of such a network is 5 Gbits/sec, which works out to over 400 petabits/day (50 petabytes). This might represent upwards of 2 million packets/second and 2 million flows/minute. This hypothetical network will be used as a baseline for comparison in this paper.

NETWORK LOAD

The network load for a centralized management system based on device-oriented protocols is not easy to assess, but even rough estimates highlight a fundamental problem. To supply one-minute averages, such a system, in receiving data from the thousands of probes, agents, and devices in the network, would be required to collect, aggregate, store in a central database, process, and analyze millions of variables every minute.

In SNMP each variable is a managed object represented by an OID in a MIB. Polling one million objects a minute one at a time is the equivalent of 9 Gbps, an obviously ridiculous figure representing about 180% of the aggregate data rate in the hypothetical network outlined above. But even assuming 100 variables reported per packet (from each of only 10,000 devices), aggregate management traffic is 90 Mbps for every million variables via SNMP.

Another source of data might be router metrics such as delivered by S-Flow or NetFlow. In the latter case, each packet can report 18 variables on 30 flows (there is no performance data). To match the flow data rate of the hypothetical network (two million flows/minute), the centralized system would have to gather this type of data at a rate of 800 Mbps (assuming no polling but simply scheduled data).

And even with both SNMP and router metrics (or RMON2), there will still be little or no performance

data, so there will be an data load from performance agents throughout the network.²

There are a number of tactics that can be used to reduce the network load to a manageable level:

- Reduce the number of devices monitored and/or agents reporting, impairing the system's ability to see all traffic.
- Reduce the sampling rate, impairing the system's ability to see transient events.
- Top the data at each device, i.e., select only the top N datasets with reference to a particular metric, such as throughput, which results in data loss due to the aggregation/topping problem discussed below.
- Aggregate the data using intermediate devices; e.g., Cisco NetFlow Collection Engines to collect and average NetFlow data from routers.

In reality, of course, a centralized management system uses all of these tactics. An RFI for such a system would typically specify 10-15 minute data granularity, will collect data from only a subset of the enterprise's own devices, and will definitely use intermediate aggregation points. It is interesting to note that even when using all these strategies, many large financial institutions, for instance, find it is still necessary to transmit management data from the network on an entirely separate network because of the load.

AGGREGATION AND TOPPING ORDER

Even in such networks, however, there is a critical problem that results in the loss of important management information. This is due to the relationship between aggregation (grouping) and topping, and the importance of the order in which they take place.

In an IP network, the identity of endpoints (clients/users and servers/resources) is specified by IP addresses, the identity of networks is specified by AS number, and the identity of applications by port. A combination of IP addresses and ports specifies a unique conversation, or flow, which is the basis of all IP communications. Groups of IP addresses, AS number, ports, and flows can thus specify any and all sets of resources and users and the communications between them on the network. For instance, a range of IP addresses might represent the servers hosting a critical C3I or logistics application;

² As discussed below, performance metrics can be derived from data supplied by the TCP flow-control algorithms. This information is not available from either SNMP or RMON-2 agents and must be gathered by specialized "probes."

other ranges of IP addresses might represent the computers assigned to particular companies, battalions, or other command elements. Aggregating or grouping data in accordance with these identities is crucial to network management, not least because it permits the rapid prioritization of problems as determined by the mission significance of a given resource or set of users.

It is critical that aggregation take place before topping, but this is impossible with SNMP and RMON2: neither protocol offers a way of creating or storing such identity information. Mission-significance information is simply not available at the device level. Topping in a centralized system dependent on such protocols inevitably takes place before aggregation/grouping. This destroys significant information, for the device is discarding data in accordance with metrics (e.g., throughput) that have no necessary relationship to the significance of that data for the network manager and the mission the network supports. For instance, the computer used by the colonel in brigade HQ might not generate a lot of traffic, so it would not be included in topped data, yet its priority for troubleshooting would be quite high.

A tiered approach can be used to overcome this in part, but at the cost of a greatly increased management burden, as noted below.

DATABASE LOAD

The database load can be calculated in a similar fashion, based on the data requirements of a leading SNMP-based device-management system (Hewlett Packard OpenView), and of router-based NetFlow data. OpenView requires 24 bytes of space per MIB variable stored. Storing one day's worth of one-minute data on one million MIB variables would thus require at least 35 gigabytes of disk space. NetFlow router metrics datagrams require 20 bytes/flow, so storing one day's worth of flow data at one-minute granularity for the example network would require about 58 gigabytes. Thus, a centralized system would be storing close to 100 gigabytes/day; adding application performance data from probes would of course require even more.

MANAGEMENT BURDEN

As noted above, a centralized system needs a tiered architecture (i.e., must use intermediate aggregation points) to deal with the flood of device-level and other management data generated by the network. This results in at least two levels of management complexity:

- Device-level: managing the dataflow from each device. This is incremental to the already-extant

task of managing the devices. However, in many cases, device-level management is a local function (“Is my access router running?”), whereas managing the data from the device is necessarily a centralized function. This complicates management by dividing it across organizational boundaries.

- Aggregation-point management: managing the intermediate tiers that aggregate data. The problem here is one of ensuring that mission-significance data necessary for troubleshooting (“Who is affected? What application?”) are not lost to topping.

Regardless, a centralized solution will, by its very nature, have thousands of management touchpoints.

ADDITIONAL CONCERNS

Beyond the data-scale management challenge, centralized device-oriented systems pose additional problems, as well.

Device-oriented protocols also don't easily cross organizational boundaries. Such data tends to be tightly-held because obtaining it requires access to mission-critical resources: the devices themselves. Yet, no matter how much control IT managers exert over their own network, as part of an internetwork its performance will always depend to some degree on other networks not under their control. This problem is quite familiar to enterprise network managers, who are dependent on the functioning of networks such as the Internet and customer or partner networks from which they can obtain no device-level data at all.

A similar problem obtains between the various management “silos” of an IT organization. Managers of the data-center portions of the application infrastructure—e.g., servers, databases, and applications—generally enjoy the luxury of managing a relatively limited set of assets within organizational boundaries and tend to regard the network as a cloud, or pipe. For network managers, server load and performance are likewise blank spots in their vision. Thus within an organization the “fingerpointing” resulting from the limited viewpoint of each silo is a massive drag on efficiency, and most of the fingerpointing is at the network team due to the relative lack of network data.

Worse, device-level data gives very little information about the actual performance or identity of the applications running on the network, nor about the

identity of the users. Linking device-level data to this necessary application and user context requires programmed “maps” whose information is rapidly rendered obsolete by changes in the network. RMON2 data, with its flow-based information, can overcome this to some extent, but the topping problem discussed above remains.

DISTRIBUTED, FLOW-BASED INFRASTRUCTURE MANAGEMENT

A distributed, flow-based approach can overcome both the data-scale management challenge and the other problems that dog IT managers dependent on legacy solutions. We will review briefly the nature of such a system and its basic advantages, look in more detail at how it can overcome the data-scale management challenge using algorithmic approaches adapted from high-energy particle physics research, and then compare its network, database, and management load to the centralized system outlined above.

The fundamental advantage of a flow-based approach is that distributed protocols of TCP/IP themselves—*by their very nature*—contain much of the data needed to manage application services end-to-end across both organizational and hierarchical boundaries. The management intelligence of a flow-based system is based on correlating data garnered using TCP, IP, BGP, and other protocols in the context of the related flow, which is a user-application conversation and thus implicitly contains the user and application context needed to understand mission significance. Thus, a flow-based management system delivers all metrics in the context of the mission the network supports.

In particular, consider the TCP operations involved in establishing, maintaining, and terminating a conversation between a server and client (Figure 1). The sequence and timing of SYN, ACK, and FIN packets used by server and client contain a plethora of information about the various components of user-perceived application performance: TCP connection setup time, network latency, data transmission time (size of transmission), server/application response time, and more.

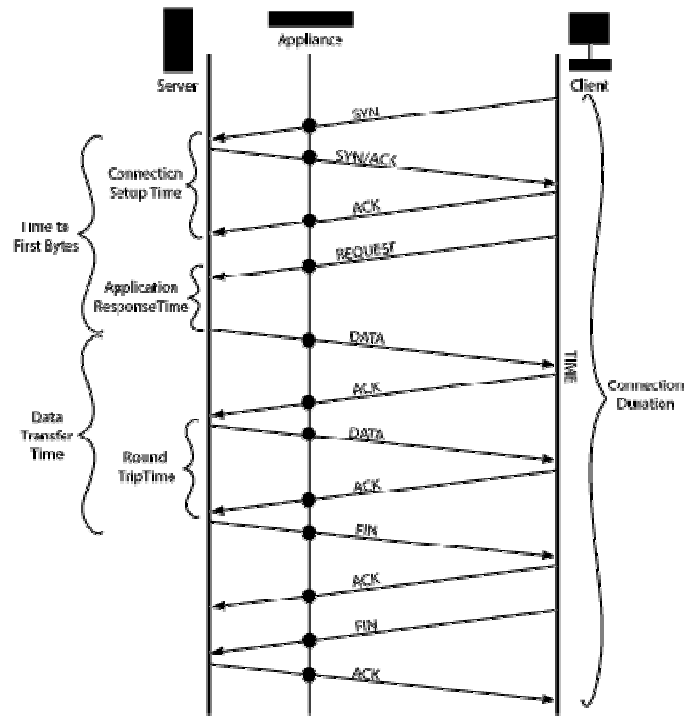


Figure 1: (U) Typical metrics derived from TCP flow-control protocol.

The table shows the types of metrics derivable from various protocols.

Table 1: (U) Flow-based metrics

Metric Type	Flow-Correlated Metrics (Flows + BGP + Traceroute)
Usage and Utilization	Packets or bytes in/out, total or throughput, by IP address or by TCP port (application)
Network Performance	Packet loss, round trip time, data transfer time
Server/Application Performance	Server response time, connection rate, number of connections
ISP Performance	Autonomous System information, peering point performance, trans-ISP performance
Route Performance	Traceroutes and route history

Flow-correlated data delivers the following:

- Detailed information about the utilization, performance, origin, and users of every application running across the network, even illicit, rogue, and hostile ones
- The relative contribution of various application performance components, giving managers cross-domain insight
- The identity of users and resources (based on sets of IP addresses, AS numbers, and ports), essential for prioritization of problem response based on mission significance
- Route quality and other information about the structure of the internetwork, essential for monitoring application performance and utilization across networks not directly instrumented
- Support for drill-down to device identity. This can be extended by integration with SNMP or RMON2 data to give detailed oversight at Layers 2, 3 and 4.

This data need be collected from very few points (Figure 2). Major data aggregation points, such as HQ, regional data centers, or other major concentrations of resources are the optimal place for flow-based data collection: a SPAN/mirror port or tap is sufficient. Provision for a BGP feed to reveal internetwork structure and appropriate tracerouting for route visibility complete the picture flow-based management can deliver. This makes flow-based systems more scalable and inherently easier to manage than device-oriented ones, since the data they deliver is largely independent of the number of devices supporting the flows, and stable through network change: maps are unnecessary.

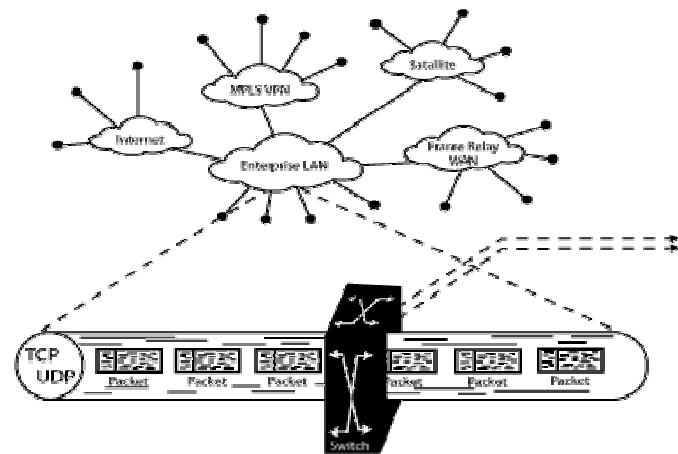


Figure 2: (U) Flow-based management data for large networks can be collected from a single point.

Distributing the intelligence required to do this correlation to edge devices coordinated by a central aggregator, in accordance with the basic distributed nature of TCP/IP, enables real-time coverage by making it possible to overcome the data-scale management challenge, as discussed below. It also promotes a more fluid division of management responsibilities across organizational boundaries: e.g., between server and network managers, or local and global managers.

OVERCOMING THE DATA-SCALE MANAGEMENT CHALLENGE

Although a distributed, flow-based approach overcomes many of the problems implicit in centralized, device-oriented systems, its scalability is just as dependent on proper exploitation of the mission significance implicit in the sets of IP addresses, AS numbers, or ports used by a flow-based system to represent resources, applications, and users. This requires a new approach to aggregation and topping.

LEVERAGING FLOW-CORRELATED DATA

The ease with which a flow-based solution encodes mission significance and user/resource relationships using simple sets of IP addresses, AS numbers, or ports, offers a lever for the necessary aggregation algorithms by allowing them to apply a contextual bias favoring groups and applications specified by the user, and thus, by definition, important to the mission. This requires an appropriate balance between grouping, topping, and temporal aggregation to keep more granular data (shorter time averages) available longer for the network entities and relationships that are more important to the mission.

GROUPING BIAS

In a flow-based network application management solution, grouping can be biased using a simple, common-sense assumption: IP addresses or AS numbers in groups, and specified applications (identified by port), are more important than others not so included or specified.

This assumption assures that aggregation is driven by the mission-significance information implicit in the groups and application specified by managers, thus preserving the data most important for troubleshooting (prioritization) and mission impact. For instance, an enterprise data center might include the corporate web site as well as business-critical applications accessed by branch offices—each office would be represented by a

group specifying its assigned IP addresses. Grouping bias would assure that the IP addresses of employees at branch offices using business applications would have more performance and utilization data stored than would those of general Internet users accessing the web site.

TOPPING BIAS

Topping bias can be applied in many contexts; the simplest and most fundamental is throughput, on the assumption that IP addresses with higher total throughput are more important than those with less.

Not only is this a common-sense assumption, but it makes the algorithms dynamic. For instance, an IP address not part of a group that is nonetheless generating a lot of traffic to a business group will have more data recorded about. It should be possible to override this assumption to account for critical yet low-throughput IP conversations.

For instance, a flow-based system might preserve the top 4000 IP addresses, ranked by total throughput, that are members of defined groups, distributed between those groups.

TEMPORAL AGGREGATION

To preserve network application data for long-term tasks such as baselining and planning, further aggregation involving temporal aging must be applied to preserve disk space. After a period determined by the management data rate and an appropriate "horizon" for high-granularity troubleshooting data, one-minute data can be averaged to five minutes, then one hour, then one day, and so forth. Again, this is done in the context of groups and applications, so data is kept longer for groups and applications with more mission significance.

DISTRIBUTED INTELLIGENCE

Using intelligent edge appliances coordinated by a global management console into a federated database overcomes the aggregation/topping problem discussed above. The global console can distribute grouping information to all the appliances, enabling them to apply the system grouping, topping, and temporal aggregation algorithms to local data using the correct contexts. Thus, grouping can be accomplished before topping, leading to accurate group averages.

From those aggregates, the appliances can create digests (also in accordance with the system algorithms) for transmission to the global console. Because grouping

was applied before topping, the averages represented by these digests are also completely accurate.

Using a federated database design, drill-downs from those digests can transparently connect a global manager user to the more detailed data stored on the appropriate local appliances.

CHARACTERISTICS OF A DISTRIBUTED, FLOW-BASED SYSTEM

Below is a summary of the likely network, database, and management load of a distributed flow-based system for the hypothetical 50 petabyte/day network outlined above. This summary is based on data from actual production systems in use at global enterprises.

NETWORK LOAD

A flow-based system will read the TCP/IP header from every packet carried by the network. Based on an average IP packet size of 576 bytes (4608 bits), the aggregate network rate represents about 1 x 10¹¹ packets/day (1 terapacket) aggregate, or about 50 gigapackets/day/appliance; the header portion of that traffic amounts to about 2 terabytes of raw data per day per appliance or probe, including BGP and traceroute data, to name two data sources that may be fruitfully correlated with IP flows to reveal AS and route performance and utilization information.

However, the flow-based information derived from this data is far more compact. Each flow represents multiple packets; contextual aggregation (IP address/AS groups and applications) and storing the data as one-minute averages further reduces the volume of data recorded. Actual operational systems of this type store on the order of 10-15 gigabytes of management data per appliance per day at a network rate of 250 mbps. However, it's easy to see that even so, that would consume most of a dedicated T1 line (about 1.4 Mbps) from each regional center, or a total of more than one-half of an T-3 (28 Mbps) to the central console, just to handle the management traffic. Even with flow-based data, a distributed design based on a federated database is still necessary to further reduce the network and database load while maintaining data context.

DATABASE LOAD

In a distributed network application management solution based on a federated database design, the most detailed data is stored in the local appliances. Only summaries of data are transmitted to the coordinating

appliance, but transparent access to that more detailed data is available on demand. Actual experience suggests that a federated design using flow-based context-driven aggregation and grouping as discussed above can reduce the database load on the global management console in our hypothetical network to less than 10,000 records/minute from each appliance, amounting to about 6 gigabytes a day: less than 30 kilobits/second from each appliance.

MANAGEMENT BURDEN

Given the ease with which flow-correlated data can be assigned business relevance using sets of IP addresses, AS numbers, and ports, it is a relatively simple task to push these sets out from a global coordination appliance to intelligent appliances, which handle the task of correlating, grouping, topping, and transmitting data digests and alerts back to the global coordinator. A federated database design allows access from the coordinator to the more detailed data stored on each edge appliance.

This makes it possible for a distributed, flow-based solution to manage a global network comprising 20 regional centers, hundreds of remote sites, and thousands of users with a single global coordinator appliance and 20 intelligent edge appliances: 21 management touchpoints.

Note that the global coordinator can itself apply the same types of relevance and aggregation algorithms to the digests it receives. This creates the potential for a multi-level architecture with even greater scalability.

SUMMARY

A distributed, flow-based management system exploits the distributed nature of fundamental TCP/IP protocols to overcome the problems inherent in managing large, dynamic networks. Such a system is complementary to existing approaches: integration of distributed flow data with SNMP and RMON2 data can provide detailed oversight of global networks at layers 2, 3, and 4 by presenting all network data in the context of its significance to the missions the network supports.

SUMMARY COMPARISON

Problem	Centralized, Device-Oriented System	Distributed, Flow-Based System
Network load	Impractically high (hundreds of megabits per second) for full one-minute-average coverage of 50 petabit per day network	~ 30 kilobits per second from 20 distributed appliances for same network
Aggregation-Topping	Topping before aggregation destroys significant information about mission-significance and priorities	Aggregation before topping preserves information about mission-significance and priorities
Database load	~50-100 gigabytes per day in centralized database	~ 6 gigabytes per day in global aggregation database, transparent access to 15-20 gigabytes per day in federated databases in distributed appliances
Management Load	thousands of management touchpoints; difficult to appropriately distribute global and local responsibilities	21 management touchpoints; natural distribution of global and local responsibilities
Mission significance (needed for problem prioritization)	Must be synthesized from device information; fragile under change	Naturally represented by flows; automatically maintained through changes
Cross-organizational coordination	Difficult: device data does not easily cross organizational boundaries	Flow data easily shared across organizational boundaries