

ABSTRACT

Assuring the performance, integrity, and security of the application infrastructure is the central function of IT in the modern enterprise. Network managers have struggled for years with their part of this effort, piecing together increasingly unwieldy collections of point solutions and largely device-oriented data sources reporting to a monolithic, central repository in an attempt to "defend the network" against "the network is slow" complaints. Nonetheless, too often, all lights are green but users are complaining about poor application response, and all the other management teams point at networking. Network application management addresses this problem by focusing on the network's role in infrastructure assurance while also giving network managers cross-domain insight into server and application performance and utilization as well. However, traditional centralized solutions based largely on data from devices and "dumb probes" cannot deliver network application management for global and other widely-distributed or complex application infrastructures. This requires a distributed, flow-based solution.



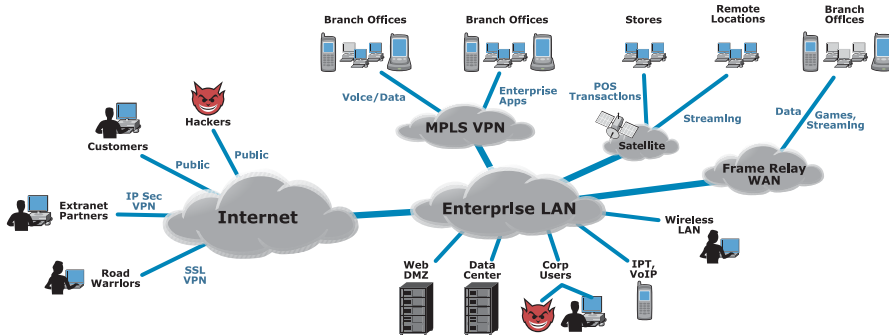
FULFILLING THE IMPOSSIBLE DREAM: MOVING BEYOND CENTRALIZED NETWORK MANAGEMENT

Assuring the performance, integrity, and security of the application infrastructure is the central function of IT in the modern enterprise. It no longer makes sense to treat network, application, server, and other management systems as separate entities, given that all of these are merely subsets of the fundamental purpose of the IT infrastructure: the delivery of application services to users, wherever, whenever, and however they need them. Managers and C-level executives alike are too familiar with the incomplete view, management inefficiencies, and lack of cross-

domain cooperation such a fragmented approach inevitably delivers.

Nowhere is the pain caused by this fragmented approach more apparent than among network managers. Their lament is familiar to anyone responsible for an enterprise network, or their supervisors: "I spend too much time 'defending the network.' I've instrumented my network and bought all the right tools, but users are still complaining about slow response, the application and server teams still say it's not their problem, and my tools still show 'all green.' So everybody blames the network, and it takes most of my time to figure out if they're right or not, and, if they're right, even more time to fix it."

The problem is that current network management solutions monitor the wrong things, and furthermore, monitor them the wrong way. By comparison to their cohorts on the server or application teams, who generally manage well-instrumented assets that are entirely under their control, network managers wrestle with an enormously complex network of networks, many of which, such as the Internet or



partner networks, are not under their control. Despite this, their traditional tools, in which they've invested heavily, bear a strong resemblance to their systems management counterparts: mainly device-level management systems that can't monitor those outside networks, and, on the parts of the network they can monitor deliver data that is difficult or impossible to relate to application response, user experience, or business impact.

Network managers have struggled with this dilemma for years, generally by trying to piece together increasingly complex and hard-to-manage collections

of point solutions and data sources—mostly devices and “dumb probes”—reporting to a monolithic, central repository. But these attempts were and remain doomed to failure: they overload managers with irrelevant data, can't deliver a real-time view, can't scale to handle an extended enterprise network, and don't match the natural, efficient division of labor needed for a responsive IT organization. What is needed is a distributed, flow-based solution based on intelligent edge appliances, in effect adopting the distribution model so successfully exploited by TCP/IP networks themselves.

Solving the Data Scale Management Problem

The central problem of network application management in a global enterprise or any highly complex infrastructure is one of data scale management: how to make sense of the sheer volume of data such a system must collect, aggregate, store, and analyze. A global enterprise network will likely deliver hundreds of applications to thousands of users across many tens of thousands of devices, coordinating operations among dozens of regional centers.

Baselining the Challenge

Real-world data suggest that up to 20 regional data aggregation and control points are not unlikely for a large global enterprise. 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. We will take this as a baseline for this white paper.

An effective network application management solution for such a global network should provide access to data and real-time alerts about end-to-end application response time (broken down by server, network, and application performance and delay to give cross-domain visibility), network utilization, route composition and quality, and ISP/AS performance and utilization. Data should represent one-minute averages, to enable both accurate peak utilization metrics and to detect transient or intermittent events, which are both common and hard-to-diagnose.

As well, we may wish to gather packet-level data, such as is offered by “sniffer” tools, for even deeper diagnosis.

The Four Challenges

The data scale management problem poses four challenges to supplying this information in a global, petabyte-scale network:

- **Challenge One: A Unified Logical and Physical View of the Network.** Correlating, de-duplicating, and visualizing data from multiple sources to present a unified logical (application, user-experience, and business context) and physical (infrastructure) view of the network at all scales.
- **Challenge Two: Real Time, End-to-End.** Monitoring all the network all the time: balancing network coverage against the network, database, and management-complexity cost of data collection
- **Challenge Three: Scalability.** Scale to manage the largest, most complex networks without imposing an overwhelming management burden, or excessive network or database impact.
- **Challenge Four: Management Agility.** Adapt easily to support the most efficient division of managerial oversight: global/regional, C-level/director/manager, security/operations, application/server/network, NOC/help desk, etc. Let users explore management data however they wish.

As this paper will show, the cost of overcoming these challenges with centralized, largely device-based solutions—those legacy systems whose heritage is systems management—is likely to be overwhelming. Trying to assemble a coherent view of the application infrastructure by aggregating data from thousands of management touchpoints into a monolithic master database is simply impractical for large or geographically-distributed networks. Consider the requirements for the hypothetical 20-regional-center network described above.

A central, terabyte-level database and sufficient processing power to

- Receive and store network data at one-minute intervals: on the order of millions of managed objects
- Correlate and analyze that data in terms of impact on application performance and business priorities *in real time*
- Correlate/de-duplicate and display alerts
- Display and/or generate reports

Access to data from the following sources

- RMON2 or NetFlow data from all routers and switches
- Agent data (e.g., SNMP) from all routers servers, firewalls, load balancers, and similar devices for device status alerts
- Data from application response agents throughout the network
- Packet-level information (gathered using portable or fixed-installation Sniffer-like products).

As we shall see below, the largely device-based approach of traditional network management systems makes it difficult to provide a unified logical and physical view of the network. This is needed to display network and application metrics in the context of application response, user experience, or business impact while preserving the ability to drill down to the underlying infrastructure. A centralized solution does not scale and real-time access to the data needed is impractical due to the enormous network and

database load imposed. Finally, a solution dependent on a monolithic, central repository and analysis engine makes the proper distribution of managerial oversight and responsibility much more difficult than it should be.

A distributed, flow-based solution is far more practical and cost-effective. The Appendix discusses the two aspects of this technology in more detail; below we merely state its advantages.

Rather than using a centralized manager to process data from simpler device agents or probes, a distributed system moves much of its management power to intelligent appliances on the edge of the network, in effect adopting the "dumb network" model that has made TCP/IP networks so successful. Such distribution is only feasible with flow-based data, which, in addition to its other advantages, is far more compact than device-derived data. As we shall see, such a solution can manage our hypothetical networks with 20 intelligent appliances managed by a global coordinator using a federated database structure.

Flows are conversations between business entities (e.g., application and user) identified by IP address and port at each end. The management intelligence of a flow-based system is based on correlating data garnered from TCP, IP, BGP, and other distributed Layer 3 and 4 protocols with the related flow. These flow-correlated data deliver a unified logical and physical view of the network. Flow correlation can also serve as the backbone of business-relevance for management data in legacy systems (e.g., from SNMP-based solutions).

Flow-based data is largely independent of the number of devices through which the flow passes and can be gathered from large networks via a single, passive connection at a major data aggregation point. These factors make possible both scalability and end-to-end, real-time coverage.

Intelligently grouping and aggregating flow-correlated data within a distributed architecture promotes management agility across multiple domains, including cross-silo collaboration, global-regional coordination, and manager/C-level distribution.

Let's look at the challenges one-by-one to see just how the two solutions compare.

Challenge One: A Unified Logical and Physical View

Unifying a logical, business-centric view of the network with a physical view of the infrastructure is critical for network application management's contribution to the fundamental goal of IT management: to assure that the application infrastructure supports business goals. This unified view is necessary for efficient troubleshooting, trending, baselining, chargeback, and similar tasks.

A logical, business-centric view of the network must present data in three contexts:

- The application context, which highlights the business functions or processes impacted;
- The user experience context, which is how problems are perceived, judged, and reported; and
- The business context, the business entities involved: branch offices, departments, buildings, users, server clusters, business applications, and end-to-end business services, and so forth; critical for prioritizing problems and planning.

The physical view is needed to support drill-down from the symptoms, which are reported in and prioritized by the above contexts, to the devices responsible.

Equally important, a network application management solution must maintain these contexts of network data at all scales, from a global overview to details of individual IP conversations—without imposing an impossible management burden. To make this practical, such a solution must support two critical functions:

- Maintaining data context through network change. Are the business-network links brittle under change, requiring re-programming, or are do they adapt without user intervention? This is required for scalability, and will be discussed in that section, below.
- Maintaining data context through aggregation. Can the system reduce the volume of network management data many orders of magnitude, both to make it intelligible and manageable, and to reduce network and database load, without losing its application, user, or business context? This is required not only for scalability, but for real-time coverage of large, geographically-distributed networks, and will be discussed in the real time section, below.

Business-Network Integration

To understand the difference between the unifying ability of a centralized system vs. that of a distributed, flow-based solution, let's first consider how each links

network operations to their business impact: what we call business-network integration (BNI). This is key to the value of network management data.

Distributed, Flow-Based Solution

Since a flow is a conversation between two business entities, flow-correlated data is a natural representation of business relationships. With such data, it is possible to encode business relationships as sets of IP addresses, AS numbers, or ports. In the first two cases, such a set, which we call a business group, can represent a natural business grouping such as servers, data centers, ISPs, branch offices, departments, buildings, users, customers, partners, or any other business-related entity. Sets of ports can represent applications. User experience can be judged based on TCP performance metrics. Grouping flow-correlated data thus delivers the three critical contexts for network application metrics: application, user, and business.

Drill-Downs

Business groups and applications also enable drill-downs to more specific information, revealing even more about the business relationships flows represent. Some of the drill-downs that are possible in a flow-based system include:

- Members of a business group (endpoints within it); e.g., servers in a data center, users at a partner site;
- Non-members communicating with members of the group (endpoints outside the group), e.g., users of an application hosted on a server within a data center;

- Other groups connected to a group, e.g., a remote office using applications hosted on a server within the data center;
- Individual IP-IP conversations (enabling further detail), and
- Applications (by port). Even unknown application activity can be seen, and its behavior reveals much about it, including whether it is a rogue application used without permission or malware, such as a worm or virus.

Flow-correlated data delivers TCP application performance information for all these groups, and utilization information for all applications and groups. Route, ISP, and AS performance and utilization are likewise available. With the proper aggregation and

topping, further drill-downs can be made available. For instance, a manager receiving an alert on an application might drill down to the business groups affected, then drill down within it to the individual IP addresses of servers or clients to distinguish between a server or a network problem by decoding the flow-control data encapsulated in the TCP header.

Even further drill-down is possible given integration of a device-level solution based on data such as SNMP. A flow-based system can identify and isolate devices by IP address or via BGP or traceroute data, which is often enough to send the problem on to the appropriate team or service provider. The addition of SNMP can supply internal device data not available from flow data, with the flow data supplying the needed contexts.

Centralized, Device-Based Solution

By contrast, much of the device and agent data upon which a traditional network management system depends does not naturally represent business relationships. A router may carry traffic representing hundreds or thousands of transactions-which of them are critical to the business? A server or application may be accessed by many different users, business partners, or customers. Which of them is most important? To answer these questions, which are critical for prioritizing problem response as well as for planning new network initiatives, traditional, largely

device-oriented systems depend on hand-coded "maps" to supply context for the data: the relationship between a set of devices and the applications, users, and business entities that they support. Any drill-downs offered are likewise dependent on the map for context. As we shall see below, the map-based approach is neither scalable nor capable of real-time coverage.

Now let us consider the problems of maintaining data context through network change and under aggregation for these two solution types.

Maintaining Data Context Through Network Change

Are the business-network links brittle under change, requiring re-programming, or do they adapt without user intervention?

Distributed, Flow-Based Solution

Flows are end-to-end phenomena, so the business *relationship* they represent is unaffected by changes in the nature, number, or status of the devices through which the flow passes. Thus, the business-network links encoded in business groups, applications, and their related drill-downs are largely independent of infrastructure changes.

The only infrastructure changes requiring manual intervention in a flow-based system are those affecting the endpoints. For instance, adding a server to a server cluster might require adding the IP address of the new server to the group representing the cluster. On the other hand, replacing an access router at a branch office would not affect the data obtained from the group of IP addresses representing that office.

Centralized, Device-Based Solution

The business-network integration maps required in a traditional system are brittle in the face of infrastructure change. Changes in any device supporting a given application or process, or addition of a new device, requires re-programming of the map, contributing to management complexity. Some more advanced systems attempt to use various forms of

discovery (based on device-level data such as SNMP) to automate the updating of such maps, at the cost of greater system complexity. Either way, for a traditional, centralized system, the burden of maintaining those maps across the data from thousands of devices would be extremely high.

Maintaining Data Context Through Network Change

Can the system reduce the volume of network management data many orders of magnitude, to make it intelligible and manageable, and reduce the network and database load, without losing its application, user, or business context? This requires sophisticated data-aggregation algorithms, and, as we shall see, is perhaps the biggest strike against any centralized system, which by its very nature cannot produce accurate aggregations of network application data in context.

Aggregation comprises three processes: grouping, topping, and temporal aggregation (which is really a form of grouping, but is here considered separately for clarity's sake).

Grouping produces statistical aggregates within a particular data context. For instance, average server response time for each of the servers in the Chicago data center. Topping truncates this data set by selecting the top N data sets ranked by an appropriate

metric, such as total throughput; in this case, preserving average server response time for the top N servers by throughput, and averaging all the others into an "Other" category. Temporal aggregation further reduces the size of the data set by averaging it across larger and larger time periods as it ages: for instance, preserving one-minute averages for a week (for troubleshooting), and then aging it to five minutes, one hour, one day, and so forth at longer intervals.

Now we can see the crux of the problem: *accurate aggregation requires grouping before topping*. If you top the data before grouping it, you are truncating the data without knowing which data sets are important in a given context. Any statistical digests of such topped data are inherently inaccurate; information has been irretrievably lost. Looking at the difference between a distributed, flow-based solution and a centralized one will make clear just what is lost.

Distributed, Flow-Based Solution

Distributed intelligence and flow-correlated data both contribute to solving the data aggregation crunch.

Flow-Correlated Data

The ease with which a flow-based solution encodes business 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 important to the user. 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 business.

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 business-relevance information implicit in the groups and application specified by the user, which are by definition important to the business. Although the flows in a global enterprise network may represent conversations between millions of IP addresses and hundreds of applications, business biasing gives preference to the IP addresses and AS numbers in groups, and likewise for applications, thus preserving the data most important for troubleshooting (prioritization) and business impact.

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 and will show up as a connected IP in the drill-downs for that group. It should be possible to override this assumption to account for critical yet low-throughput applications.

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. It should be possible to override topping to account for critical yet low-traffic IP conversations.

Temporal Aggregation

To preserve network application data for long-term tasks such as baselining, planning, and chargeback, 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.

Distributed Intelligence

Using intelligent edge appliances coordinated by a global management console into a federated database overcomes the grouping/topping problem. 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.

Centralized, Device-Based Solution

A centralized system would need to apply the same kind of aggregation. However, as we shall see below in the section on real-time coverage, it is impossible to send all the network data generated by thousands of probes (even flow-based ones), agents, and devices to a central, monolithic database. The data must be topped at the agent, probe, or device level to reduce the network and database loads, but the information needed for grouping (context) is only available in the central system. Grouping already-topped data guarantees inaccurate statistical digests that lack true application, user, and, especially, business context.

Intermediate aggregation of data in a way that would maintain its context is impossible in a true centralized system, because the aggregation points, by definition, lack the intelligence to do so. If they were intelligent, it would be a distributed system. If they were not intelligent, then there is the additional management burden involved in making the proper aggregation

decisions so that important data is not lost. This burden would be insupportable, especially in the face of network change.

With a great deal of effort and expense, it might be possible to use SNMP, NetFlow or RMON data, and data from application response agents to create groupings analogous to the business groups and applications possible with a flow-based solution. One could even create intermediate aggregation points that would use those groupings to intelligently aggregate data, in effect creating a distributed system. However, such intermediate devices would have to understand a wide variety of MIBs as well as the any proprietary data models from response agents, so not only would the cost of managing such devices likely be very high, but their coverage would tend to lag behind as new devices were introduced to the network. By contrast, a flow-based solution, being largely-device independent, requires no such effort.

Challenge Two: Real-Time, End-to-End

One of the biggest problems in designing a network management system for a large network is the necessary trade-off between network coverage and the impact of data collection. Data collection has three costs associated with it:

- Network load. How much additional traffic must the network carry from the management data sources to the central or top-level management database?
- Database processing load. Similarly, how much data must the database be able to process and store per second, minute, hour, or day?

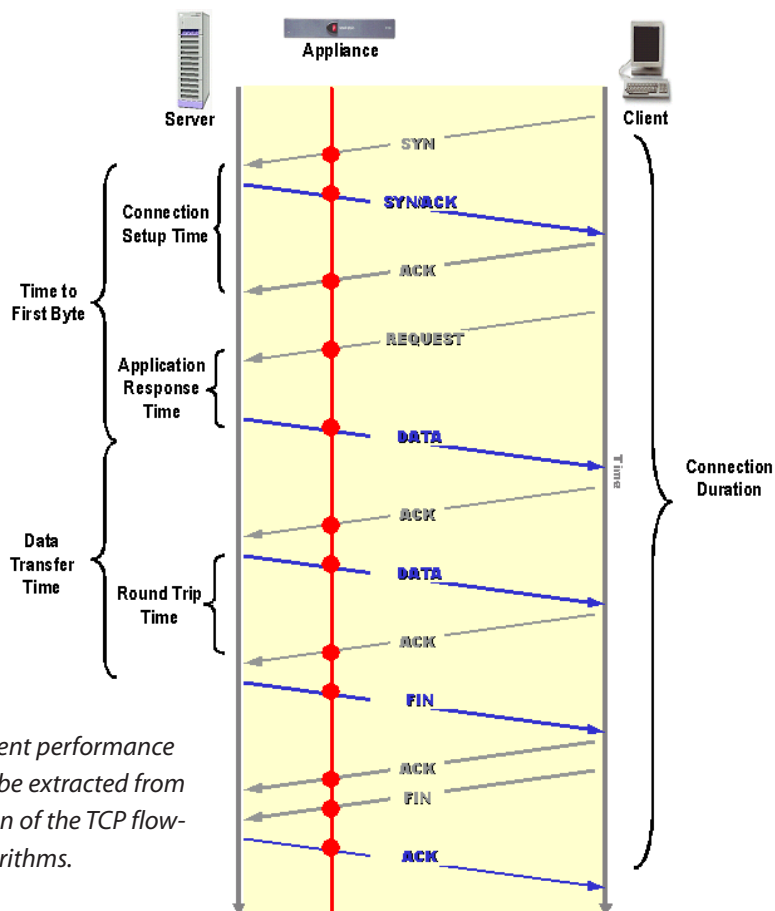
- Management complexity. Most important, how much additional time and effort is required to manage the management system and all its data sources?

All of these weigh much more against a centralized, traditional system than a distributed, flow-based system, making it practically impossible for the former to deliver real-time coverage of a large enterprise network. To see why, let's compare how these factors impact the NetSensory Enterprise Architecture vs. a centralized, traditional management system when considered in the context of the hypothetical 20-regional-center network

Network Load

How much additional traffic must the network carry from the management data sources to the central or top-level management database?

Distributed, Flow-Based Solution



Many different performance metrics can be extracted from the operation of the TCP flow-control algorithms.

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 1×10^{11} 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 (business 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 or 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.

The network load for a centralized management system is not easy to assess, but even rough estimates highlight a fundamental problem. To supply the one-minute averages we have specified, 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 our hypothetical network. But even if we assume that we can get 100 variables reported per packet (from each of only 10,000 devices), we're still looking at 90 Mbps in management traffic 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 our 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), we still have little or no performance data, so we must add in the data load from performance agents throughout the network.

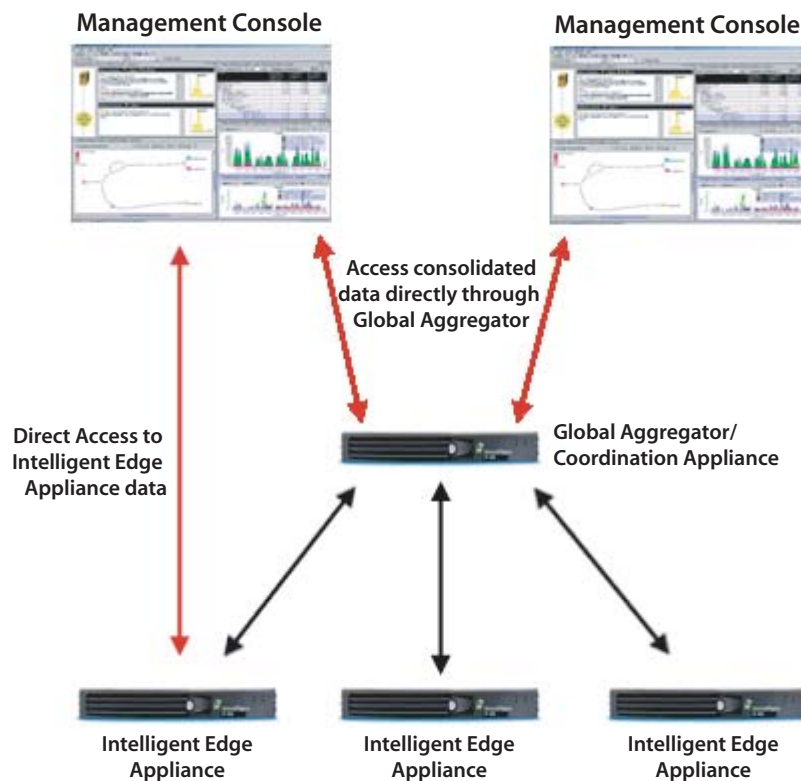
There are a number of tactics that can be used to reduce the network load to a manageable level we will need to:

- Reduce the number of devices monitored and/or agents reporting, impairing the system's ability to see all traffic. Since it is dependent on data not available from public or third-party networks, its oversight is already limited; this will further reduce its ability to troubleshoot problems.
- Reduce the sampling rate, impairing the system's ability to see transient events. (Based on actual data from a major Internet portal, for 70% of Internet traffic experiencing congestion, the episode lasts less than two minutes).
- Top the data at each device, which, as it necessarily entails topping before grouping, results in the loss of data, as explained above.
- Aggregate the data using intermediate devices; e.g., Cisco NetFlow Collection Engines to collect and average NetFlow data from routers, which entails increased management complexity and, again, loss of data from topping before grouping.

In reality, of course, a centralized management system uses all three of these tactics. An RFI for such a system would typically specify 10-15 minute data granularity, it won't be able to get device information from public and third-party networks and 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 find it is still necessary to transmit management data from the network on an entirely separate network because of the load. Most enterprises cannot afford such a solution.

How much data must the database be able to process and store per second, minute, hour, or day?

Distributed, Flow-Based Solution



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, the global aggregator, 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 aggregator in our hypothetical network to less than 10,000 records/minute from each appliance, amounting to about 6 gigabytes a day.

Centralized, Device-Based Solution

An appliance based on an open-source database can store on the order of 100,000 records/minute. A centralized database attempting to store the same amount of data as 20 intelligent appliances would thus need to process 2 million records/minute.

Another way to look at this is in terms of raw data storage, although a centralized system would be storing data in a different format from our flow-based example, based on different data. Consider the data requirements of a leading SNMP-based device management system, and of router-based data. The SNMP system 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. Typical router metrics datagrams require 20 bytes/flow. Storing one day's worth of flow data at the rate collected by the NetSensory EA would thus require about 58 gigabytes. Thus, without any application

performance or route/ISP quality data and very little AS data, a centralized system would still be storing close to 100 gigabytes/day.

Either way, while an enterprise may be willing to invest in this big a database for CRM, it's unlikely the IT department would be willing to face such a high expenditure for network management.

Then there's the processing cost involved with turning that data into information useful for troubleshooting application performance problems and assessing their business impact. Knowing the status of a device, or how many packets it has processed in the last minute, or a summary of flow data, or even detailed application performance data is of little use unless you know who is affected and what impact this has on the business. As discussed above, business relevance is considerably easier to obtain with a distributed, flow-based solution.

How much additional time and effort is required to manage the management system and all its data sources?

A simple metric for measuring management complexity is the number of management touchpoints a solution requires.

Distributed, Flow-Based Solution

This is perhaps the biggest win for a distributed, flow-based network application management solution. 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.

Centralized, Device-Based Solution

As noted above, a centralized system would need a tiered architecture to deal with the flood of device-level and other management data generated by the network. So we have 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. To this add the burden of managing various application response agents.

- Aggregation-point management: managing the intermediate tiers that aggregate data. The problem here is one of ensuring that business-relevance data necessary for troubleshooting (Who is affected? What application?) is not lost in aggregation. As we have seen, this is best accomplished by pushing the necessary intelligence out to edge appliances.

The result: a centralized solution would, by its very nature, have thousands of management touchpoints.

Challenge Three: Scalability

A flow-based, distributed network application management solution enjoys the same advantage in

scalability as it does in real-time, and for much the same reasons discussed in the previous sections.

Distributed, Flow-Based Solution

A distributed, flow-based solution avoids these scalability issues the same way that TCP/IP does: by pushing intelligence—in this case, data collection, analysis, storage and, most important, action—to the edge. Intelligent edge appliances apply global business rules to locally assign business relevance to data, aggregate it in accordance with that relevance, and transmit digests to a global coordinator appliance that delivers a unified view of the network. As we shall see below, this gives a distributed system tremendous

"management agility," which makes possible the distribution of management functions and actions in accordance with business needs: global/regional, C-level/director/manager, application/server/network, etc.

Flow-correlated data does its part here: such metrics are largely infrastructure independent, yet allow drill-down to identify specific device misbehaviors, and millions of flows can be monitored from a single point,

rather than depending on the complexity of multiple SNMP, RMON and application response agents. Flow

data also contains information about the server and network components of application response.

Centralized, Device-Based Solution

Traditional network management solutions, with their device-oriented, system management heritage, are inherently difficult to scale. This is especially so given the autonomous nature of TCP/IP networks, which manage much of their own flow rate and routing, so that a given application may be supported by many different network devices, depending on the interactions of network traffic. Each device added thus represents another management touchpoint, the transmission of another management data stream

across the network, and a potentially enormous number of data variables representing not only the device's internal state but the hundreds of different applications and thousands of users it may support—information necessary for business relevance.

As a result, the management burden represented by thousands of devices, agents and probes reporting to a central console are impractically high, and there is no economy of scale—these problems just get worse the larger the network.

Challenge Four: Management Agility

Does the solution adapt easily to support the most efficient division of managerial oversight: global/regional, C-level/director/manager,

application/server/network, security/operations, NOC/help desk, etc. Does it let users explore management data however they wish?

Distributed, Flow-Based Solution

A distributed flow-based solution promotes cross-organizational dialogue, local/global coordination, and executive dialog.

Cross-Organizational Dialogue

No single traditional network management solution can supply all the types of metrics needed to manage a large enterprise network: usage/utilization, network performance, server performance, client performance, ISP performance, and route performance. Instead, each type of system is capable of seeing only certain kinds of problems, and is blind to problems arising elsewhere along the end-to-end path that defines application response and thus user experience.

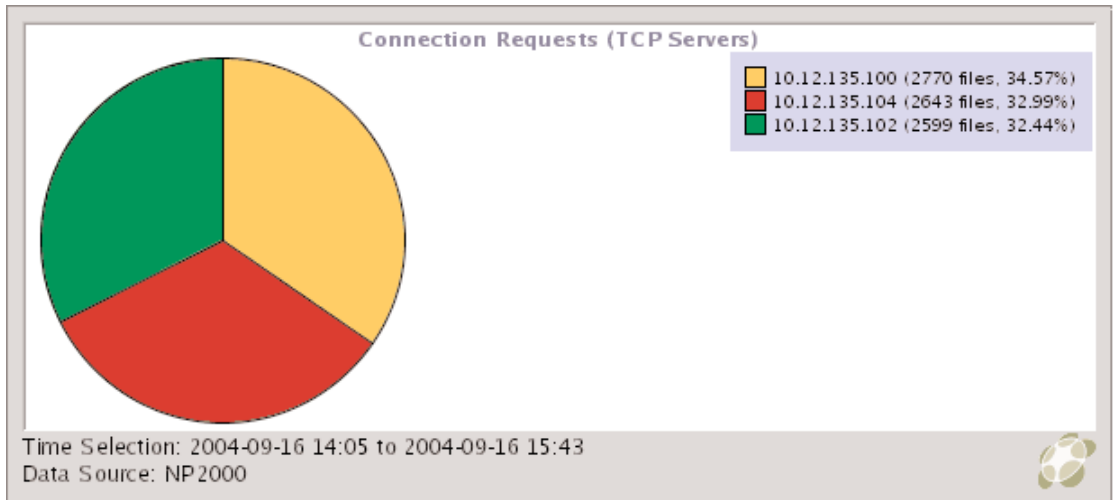
For this reason, each type is the favored tool of different management teams: device management systems for networking teams, and application response time systems for application teams. The result is the bane of CIOs everywhere: fingerpointing, lack of cooperation, and far too much time wasted by teams troubleshooting problems that turn out not to be in their management silo.

Each intelligent edge appliance in a distributed, flow-based network application management solution monitors end-to-end TCP or UDP flows—conversations between servers and clients—by aggregating the data from TCP/IP packet headers as they pass through a data aggregation point. By itself, this flow data supplies detailed information on application performance and network utilization; other information, from BGP and traceroute daemons, is correlated with the associated flows to integrate route and ISP/AS data as well. With all this flow-correlated information available in one place, a flow-based system can immediately identify a problem as originating in the server or network, enabling the related teams to zero in on it, using their more specific tools if necessary. This capability alone can greatly increase coordination between the Help Desk and the NOC.

Even greater application granularity can be supplied by monitoring at multiple points along the data flow of a multi-tier architecture, enabling immediate localization of application response problems to the database, application server, load balancer, or other device responsible.

Finally, because a flow-based system sees all application activity (by port), whether or not anyone knew said application was running on the network, it can quickly reveal malware and rogue applications or users, making it an ideal tool for security troubleshooting and greater coordination between network operations and security teams.

Flow-based solutions are often used to validate the operation of a load balancer in a multi-tier architecture.



Local/Global Coordination

There is a further problem in large networks: finding the balance between local and global control, so that global managers are not overwhelmed by local details, and local managers always have access to local data even when connectivity to the centralized system is lost—as may well be the case during a network problem!

In a distributed, flow-based solution, most of the system's intelligence and stored data resides in each of the local edge appliances, available in full detail to local administrators even if the connection to the global coordinator is lost. Yet that data is available transparently on demand to central administrators when they need. Integration at the local level with smaller, more manageable SNMP-based solutions, can also give a Layer-2 drill-down to the managers with day-to-day responsibility for the devices involved.

Executive Dialogue

The business-network integration advantages of a distributed, flow-based system also enable it to promote executive dialog. Because application response and network data are always presented in a user, application, and business context, it is relatively easy to create and distribute reports that tell C-level

executives, line of business managers, and other managerial constituencies just how network operations are impacting their spheres of concern. Such reporting is also a critical component of efficient network planning, baselining/trending and chargeback.

Centralized, Device-Based Solution

A centralized, device-based solution cannot easily facilitate cross-organizational dialogue or executive

dialog, and by its very nature tends to impede local/global coordination.

Cross-Organizational Dialogue

Although a centralized, device-based Solution can theoretically deliver all the metrics needed for efficient cross-organizational dialogue as described above, the network and database load, and even more the management burden of correlating different kinds of data from so many different devices make it

impractical for any large network. Such solutions are more practically deployed on a local basis under the aegis of a global, distributed flow-based solution that can supply the context needed for coordinating across organizational IT boundaries.

Local/Global Coordination

this challenge is one that a centralized system, by its very nature, must fail. The majority of events and alarms taking place in a data center, for instance, likely affect only the immediate environment, and do not require the attention of managers with global oversight. Indeed, bringing all that to their attention will swamp them with unnecessary information and cripple their ability to manage the global aspects of the network. But a centralized system, since it must process all events and alarms centrally, will do just that. At the very least, some programming must be in place to suppress, or relay back to the related local center, the events and alarms that are of only local interest. The same goes for data that is of only local interest: central administrators will be confronted with data overload, severely hampering their ability to focus on what matters.

In addition, as noted earlier, the process of aggregating local data to transmit it to the central console inevitably results in a loss of knowledge that could have been used more effectively "on the spot" to solve problems. And, with a centralized system, local administrators must log into the central system (with all the security management headaches that entails) to troubleshoot a problem, perhaps accessing a data store thousands of miles away to learn the condition of a router in the next room—data that is not available if the problem affects their link to the central system! The result is extremely inefficient use of management resources, and additional management burdens.

Executive Dialogue

Again, the difficulty with with a traditional solution supplies application, user, and business context to network data makes it an inefficient choice for

promoting executive dialog. The programming burden involved is necessarily much higher than with a flow-based system..

Summary

A distributed network application management solution based on flow-correlated data collection and analysis is needed to manage application response issues in large, complex, distributed networks. Such a solution presents data in its application, user and business context to promote efficient troubleshooting as well as tasks such as planning, baselining, and chargeback, and provides real-time coverage at a practical cost. It can also serve as the backbone of business-relevance for management data in legacy

systems (e.g., from SNMP-based solutions). A distributed, flow-based solution scales to the largest networks, and allows the efficient division of management responsibilities across several axes: global/regional, server/application/network, C-level/manager, and others.

The Table below summarizes the advantages of a distributed, flow-based solution over a centralized, device-based approach.

Challenge	Centralized, Device-Based Solution	Distributed, Flow-Based Solution
Unified view of the network	Device-based data does not naturally represent application, user, and business performance or relationships; unified view must be manually programmed and maintained. Difficult to deliver business relevance at all scales. Topping before grouping loses information.	Flow-correlated data naturally represents application, user, and business performance or relationships. Unified logical (application, user, and business) and physical view of the network easily created and automatically maintained. Business relevance at all scales through intelligent grouping and topping.
Real-time coverage	Network and database load imposed by data from thousands of devices makes real-time coverage impractical; sampling error of typical 10-15 minute granularity prevents accurate peak usage statistics. Limited network coverage (public and third-party networks are invisible).	Flow-based data makes one-minute granularity possible in global network for accurate peak usage statistics, no sampling error, full network coverage across all network types: WAN, LAN, VPN, MPLS, Internet, and third-party (e.g., partner or customer).
Scalability	Thousands of management touchpoints. Information delivered highly dependent on number of devices in network, extremely sensitive to infrastructure changes.	Minimal number of management touchpoints; information delivered largely independent of number of devices in network or changes in infrastructure.
Management Agility	Necessity of manual programming for unified view make both cross-organizational and executive dialogue difficult. Centralized structure cripples local oversight, overloads global managers.	Unified view of metrics and business relevance promotes both cross-organizational dialogue across management silos and executive dialogue, empowers local oversight while avoiding global overload for better local/global coordination, and executive dialog.

Appendix: Distributed, Flow-Based Technology

There are two requirements for an effective network application management solution; flow-based, correlated network and application data, and

distributed "intelligence." We will briefly review each of these before reviewing the ability of the two different solution types to meet the four challenges.

Flow-Based, Correlated Network and Application Data

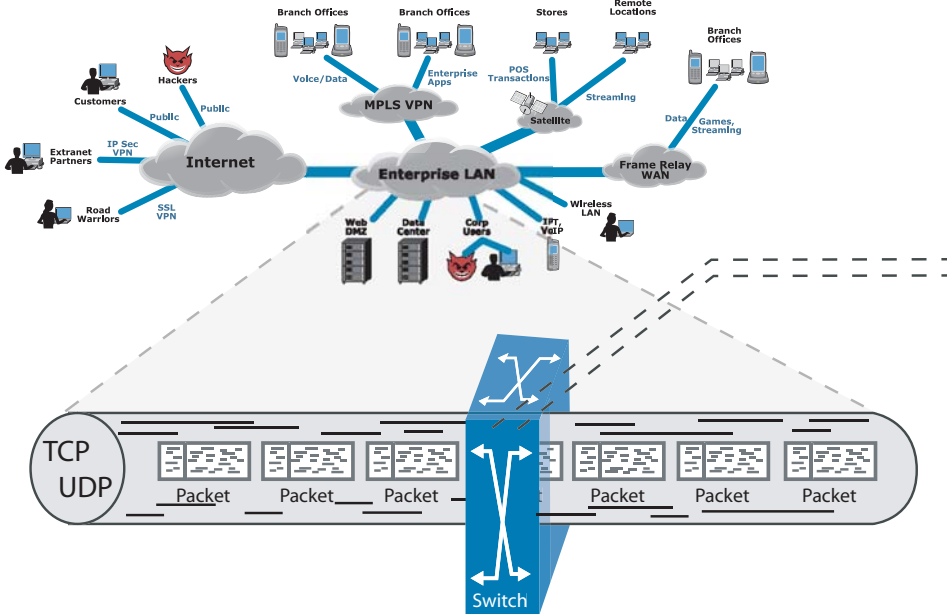
The management intelligence of a flow-based system is based on correlating data garnered from TCP, IP, BGP, and other distributed protocols with the related

flow, which delivers the five different types of network information necessary for network application management.

Metric Type	Flow-Correlated Metrics (Flows + BGP + Traceroute)
Usage and Utilization	Packets or bytes in and out, by IP address (end point) or by TCP port (application)
Network Performance	Packet loss, round trip time, throughput, network transfer time
Server Performance	Server response time, connection rate, number of connections
Client Performance	Fetch time, reset rate, payload size
ISP Performance	Autonomous System information, peering point performance, trans-ISP performance
Route Performance	Traceroutes and route history, time to live (hops)

Flow-correlated data is a far more compact representation of network state than device-oriented data, since it is independent of the number of devices through which the flow passes. A flow-based system

can also monitor very large networks from a single, passive connection at a major data aggregation point. These factors make possible both scalability and end-to-end, real-time coverage.



NETWORK FLOWS

Any IP Connection

- > End-to-end, Layer 4
- > Source, destination
- > Application or service port

Represents Business Relationships

- > Conversation between business entities
- > Business resource provider (e.g., server)
- > Business resource consumer (e.g., user)

Non-Invasive Monitoring

- > Via switch spanning port or tap
- > No agents, no SNMP, no polling, no synthetic transactions

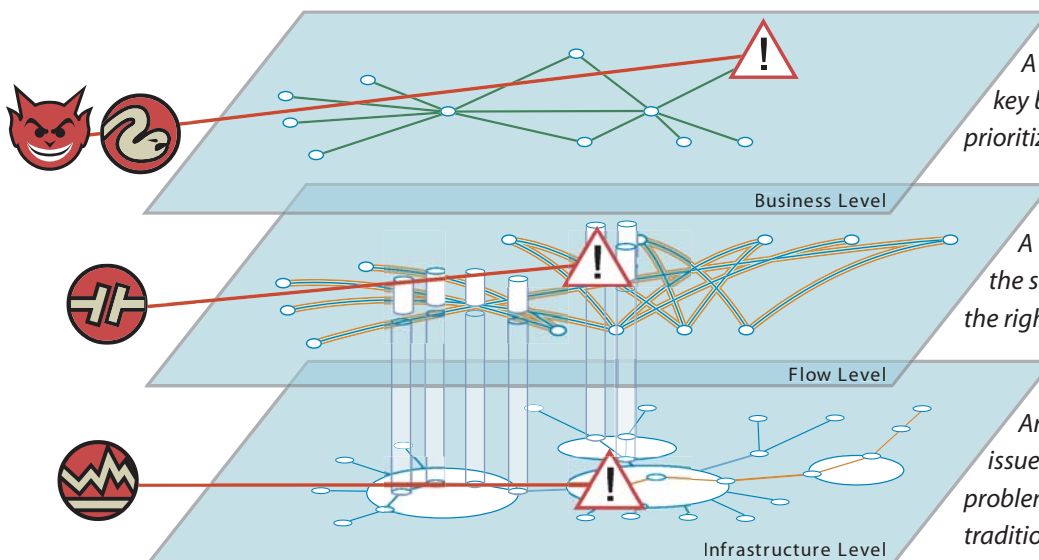
All the Traffic, All the Time

- > No sampling error

Correlation also delivers application, user, and business context at all scales for greater management agility across multiple domains, including cross-silo

collaboration, global-regional coordination, and manager/C-level distribution.

A flow-based system delivers three important views of the network:



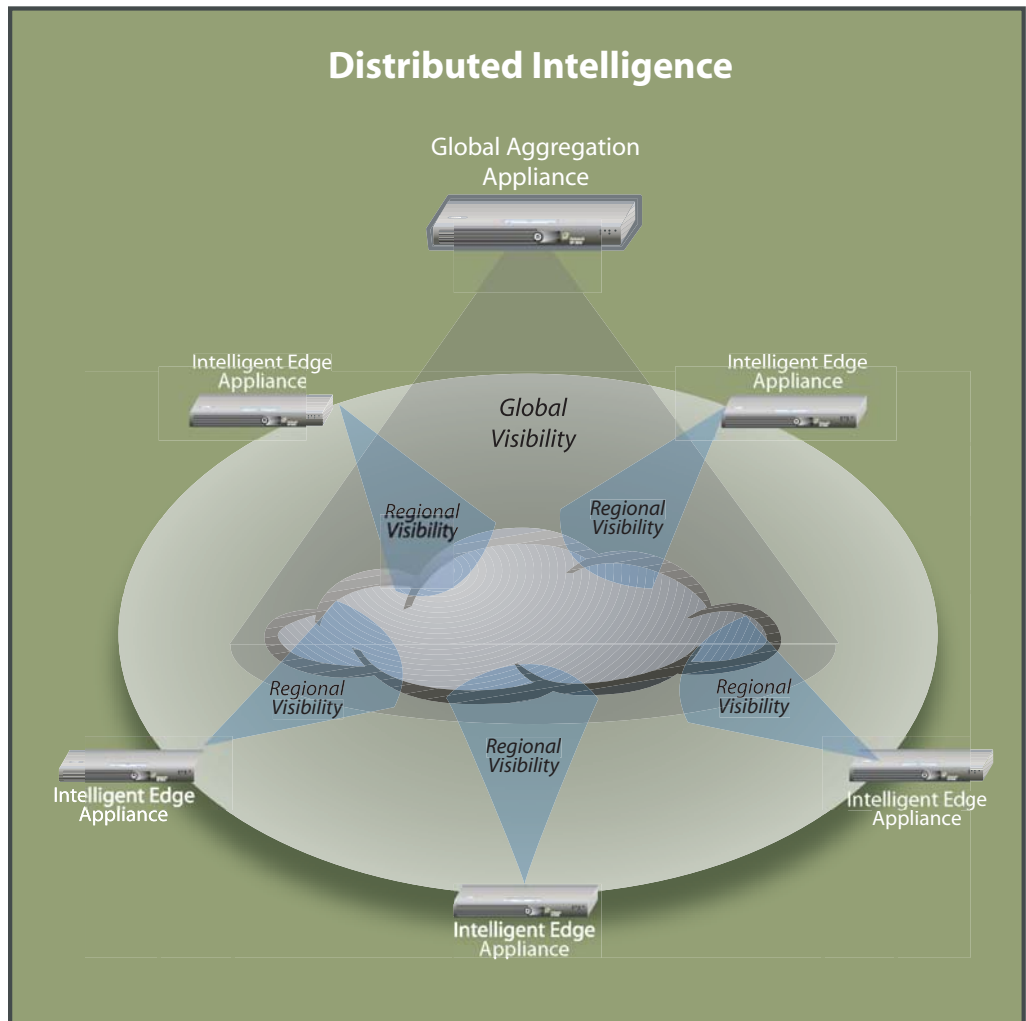
A **business view**, to report performance for key business functions, sites, and services and prioritize troubleshooting;

A **flow view**, to quickly isolate problems to the server, application, or network and assign the right management team; and

An **infrastructure view**, to pinpoint cloud issues such as peering, route, BGP, and MPLS problems that cannot even be seen by traditional solutions.

Rather than using a centralized manager to process data from simpler device agents or probes, a distributed system moves much of its management power to intelligent appliances on the edge of the network, in effect adopting the "dumb network" model that has made TCP/IP networks so successful. Such distribution makes real-time coverage feasible by

transforming data to more compact knowledge—data in context—in the edge appliances and transmitting it to a global aggregation appliance, reducing the network and database load and making the solution inherently more scalable. It also adapts more easily to the chosen management structure of the enterprise, especially the balance between global and regional.



Such a distributed system is also poised for cooperation with other solutions in a "management grid," using a variety of protocols. SNMP, of course, is already in use for this purpose; other protocols such as XML will also play a part in such grids. This not only provides user, application, or business context-aware

drill-down to other types of network data (such as specific device parameters in the case of SNMP) for troubleshooting, but is a key element in initiatives such as autonomic computing, grid computing, and the like.