Applying Network Policy Control to Asymmetric Traffic: Considerations and Solutions

An Industry Whitepaper

Contents

Executive Summary ................................... 1
Introduction to Routing Asymmetry ............... 2
Types of Routing Asymmetry ...................... 3
  Flow Asymmetry .................................. 3
  IP Asymmetry ..................................... 3
Considerations and Potential Solutions .......... 4
Evaluation Criteria ................................. 4
Potential Solutions ................................. 5
  Ignore the problem ............................... 5
  Network/external approaches .................. 6
  Intersect all paths with one box .............. 7
  Share some state between multiple boxes . 7
  Ensure core affinity ............................. 10
Conclusion .......................................... 12
Summary of Approaches to Overcome Routing Asymmetry ......................... 12
Additional Resources ............................. 13

Executive Summary

Traffic asymmetry is widespread and pervasive throughout networks, and broadly speaking takes two forms: Flow Asymmetry, occurring when a flow’s packets traverse different links; and IP Asymmetry, occurring where multiple flows from the same IP address traverse different links.

Routing asymmetry increases network efficiency and redundancy, but also complicates some other matters.

In the context of network policy control, routing asymmetry causes serious challenges that must be overcome so that CSPs can implement critical use cases such as accurate charging, policy-based measurements, and congestion management.

To be a viable policy control platform, a system deployed in the network’s data path must be able to operate in the presence of all types of asymmetry, for all types of traffic. If these conditions aren’t met, then the CSP will not be able to make full, consistent use of policy control.

There are a handful of approaches to overcoming asymmetry, but only three actually preserve complete functionality: deploying only where traffic is guaranteed to be symmetric, intersecting all possible paths with one (large) platform, and clustering to preserve processor core affinity.

Of these three approaches, only clustering offers a combination of versatility, efficiency, and an attractive redundancy model.
Introduction to Routing Asymmetry

The major innovation of packet-switched networks was removing the path concept that existed in earlier telephone and telegraph-style networks. Packet networks are entirely based on per-hop behavior and addressing, and each packet carries all of the addressing information required for each network transport device to move it to the ‘next hop’. This design is in contrast to circuit-switched networks such as those based on Asynchronous Transfer Mode (ATM), in which an end-to-end path is defined and information flows along predefined links.

In practice, an approach is commonly used where, when there is more than one choice of where to send a packet a hash is performed of the packet’s addressing and the link chosen is thus a relative constant. Commonly used hash mechanisms are source-only, destination-only, and source XOR destination. The overall field of such packet balancing is called Equal Cost Multi-Path (ECMP) routing. These hash mechanisms are stable as long as there is no change in the number of links, or routing tables, at each hop. As a consequence of the per-hop hashing, it is highly likely that a single TCP or UDP flow from a single source to a single destination will traverse a variety of paths. That is, traffic flow on any given point in a path is not symmetric (i.e., that point does not see both the upstream and the downstream for all packets in a flow) - the routing is said to be asymmetric.

Figure 1 shows an example of typical routing asymmetry in which a consumer application interacts with a destination server through two different routers; each router has links to three separate transit networks. As a result, each and every packet will take one of six different links between the consumer and the server.

A recent study\(^1\) shows asymmetry is widespread and pervasive throughout networks, with the studied network links showing up to 90% of packet traffic to be asymmetric. In a typical network, with multiple layers of aggregation, only the network edge (i.e., the last piece of the path) is free of asymmetry.

---

\(^1\) “Observing routing asymmetry in Internet traffic” · Dusi Maurizio · Universita' degli studi di Brescia, Brescia, Italy and John Wolfgang · Chalmers University of Technology, Gothenburg, Sweden. [www.caida.org/research/traffic-analysis/asymmetry/](http://www.caida.org/research/traffic-analysis/asymmetry/)
Types of Routing Asymmetry
Modern data networks exhibit different kinds of routing asymmetry; the types that are present vary from network to network, and it is useful to have some definitions.

Flow Asymmetry
In the context of network policy control, a flow is defined as a group of 5-tuple classifiers (i.e., source IP address, destination IP address, source port number, destination port number, and the protocol ID), but it can also include such things as MPLS tunnels. Flow asymmetry occurs when a flow’s packets traverse different links.

- **Full flow asymmetry**: occurs when each packet in a flow may take any one of several links in either direction (i.e., upstream or downstream)
- **Consistent partial flow asymmetry**: occurs when all the packets in a flow in a given direction are on one link; in other words, all upstream packets are on one link and all downstream packets are on another link

IP Asymmetry
We can also examine routing asymmetry as a function of addressing, where multiple flows from the same IP address traverse different links.

- **Full IP asymmetry**: occurs when all flows from a given subscriber IP may be on different links
- **Partial IP asymmetry**: occurs when flows between the subscriber IP and a given endpoint are on the same link, but flows to a different Internet endpoint IP are on a different link
- **Consistent IP asymmetry**: occurs when all upstream traffic from an IP traverses one link, and all downstream traffic to an IP traverses another link
Considerations and Potential Solutions

The benefits of routing asymmetry (e.g., least-cost paths, increased redundancy, etc.) far outweigh the slight negatives (e.g., packets occasionally arrive out of order), so routing asymmetry should be considered a permanent characteristic of all networks.

In the context of network policy control, communications service providers (CSPs) need to implement critical use cases such as accurate charging, policy-based measurements, and congestion management in the presence of this widespread asymmetry:

- Charging requires the accurate counting of volume, duration, and events; the solution needs to be able to count these metrics even when the traffic takes a variety of routes, over the full duration of a flow.
- To be useful and actionable, business intelligence must be accurate; the solution must be able to deliver all metrics (e.g., volume, quality of experience scores, application- and subscriber-awareness, etc.) consistently and correctly in the presence of asymmetry, over the full duration of a flow.
- Policy enforcement must be applied accurately to avoid negative consequences; for instance, to achieve precisely targeted congestion management, only the traffic corresponding to users on a congested resource should be managed.

Any systems that are deployed in the network’s data path must be able to function in the presence of asymmetry - this requirement is fundamental.

Evaluation Criteria

First and foremost, a network policy control solution must work when faced with routing asymmetry; that is:

- the functions available when deployed in an environment with routing asymmetry should be exactly the same as those functions available if there was perfect symmetry.
- the functions should behave precisely the same, and be as effective, in an asymmetric routing environment as they do and are with perfect symmetry.

Fundamentally, these functions include but are not limited to:

- traffic identification (e.g., application, protocol, video provider, etc.)
- traffic measurements (e.g., duration, event counting)
- advanced metrics (e.g., video quality of experience)
- billing and charging (e.g., counting volume, duration, events) for both prepaid and postpaid scenarios
- policy enforcement (e.g., quality of service marks, shaping and rate-limiting, session management)

These conditions should be true for all types of asymmetry (as defined above), for all types of traffic (e.g., TCP and UDP, tunneled/encapsulated and not, etc.), and must also apply for long-lived flows that can switch routes multiple times.

---

2 For instance, consider a long-lived video flow. During the course of a two-hour movie, the video might switch routes (e.g., different routing paths from a single source, or different sources/CDNs for a single movie) many times.
If these conditions aren’t met, then the CSP will not be able to make full, consistent use of policy control, and will have to choose which use cases to pursue. Perhaps worse, a CSP might conclude that a policy control solution works in all cases when it really only works for a subset; a simple network infrastructure change could inadvertently render the policy control solution completely ineffective.

Secondary considerations include cost and complexity:

- How much will the solution cost to deploy and maintain over its lifetime? Considerations include: amount of equipment, resilience to network change, redundancy models, upgradeability, rack space efficiency, etc.
- How complex is the solution to deploy and maintain? Is additional cabling involved? Can the solution be easily managed?

Potential Solutions

There are a range of potential solutions that must be evaluated against the criteria outlined above.

Ignore the problem

In this approach, policy control platforms are deployed at various locations and each independently does its best (Figure 2). As a result, traffic identification (the foundation of policy control) is performed with whatever traffic is available to a box; practically, this means that an individual box only seems some packets associated with a flow.

Some vendors include an “asymmetry mode” in their equipment which purports to deliver reasonable functionality even when only parts of a flow are visible, but the reality is that such modes cannot overcome the realities of traffic formats.

To illustrate why an asymmetry mode is doomed to failure, consider the case of the HTTP protocol, which has a ‘GET’ from the client to the server and a response from the server to the client. Generally, each side can be identified as HTTP, but some information on the forward path is also needed on the reverse path.
In the example shown by Figure 2 and Figure 3, Box-1 will see the client-to-server packets, and Box-3 will see the server to client packets. Both can identify the flow as HTTP but only Box-3 will know it is a jpeg image, and only Box-1 will know the host or URL. If the CSP has a policy to zero-rate all images from advertising domains, then it is not possible to achieve this policy in this scenario. It is worth bearing in mind that HTTP makes up the majority of Internet traffic.

The problem gets even more complex when we examine the case of ‘tracker’ protocols - for example, those that exchange some information on a ‘control plane’ to describe future flows on a ‘data plane’. SIP, FTP, and RTMP all fall into this category, as do many others. Some of these protocols have no signatures for the ‘data plane’ protocols (i.e., it consists entirely of unrecognizable data packets), and the traffic type or condition cannot be identified and tracked. With FTP, for example, the data flow is the raw file transferred and there is no signature possible.

In summary, the approach of ignoring routing asymmetry fails the first criterion: it is completely ineffective, even if the platform has a supposed “asymmetry mode”.

Network/external approaches
In this approach, the asymmetry issue is addressed outside of the policy control platform. There are two means by which this objective can be theoretically achieved:

1. Deploy the policy control platforms where there is no asymmetry; practically, this means deploying at the extreme subscriber edge
2. Redesign the network to extend the symmetric path inward from the subscriber edge, so that there are more locations at which to deploy the policy control platform in a symmetric environment

Both these options should preserve policy control functionality, so they pass the first criterion. However, both suffer upon further examination.

The primary drawback of the first option is the massive cost associated with having a policy control platform on every last path (and subsequently maintaining those many platforms over time). We can therefore conclude that this approach is only favourable where the cost/benefit economics of the deployment make sense.
Applying Network Policy Control to Asymmetric Traffic: Considerations and Solutions

The primary drawback of the second approach is that it lowers network resilience and redundancy (and incurs massive cost to do so), so the CSP would be forced to undermine the entire network to accommodate the policy control platform. Remember, asymmetry is a solution to several network problems, so it is a good thing and should not be removed.

**Intersect all paths with one box**
It is reasonable to ask if a single box can intersect all the possible traffic paths, thereby overcoming the problem of routing asymmetry.

![Figure 4 - A single policy control platform spans all potential paths](image)

The main drawbacks to this approach are scaling limitations and issues of reliability.

First, in many networks the sheer scale required by this approach is unattainable by a single box, thereby rendering it moot.

Second, the network engineers designed the network with redundant routers and links for a reason, and a single attached/spanning element introduces a single-point of failure. For double the cost, the CSP can get some (but not entire) comfort with redundancy. This approach is not appropriate for any CSP that wishes to avoid the real potential for a high-impact and widespread traffic outage. It undermines the entire concept of carrier-grade, falling well below the minimum level of robustness and redundancy required for today’s networks.

However, if a CSP is comfortable with the added risk, and is confident that scaling isn’t an issue now and, importantly, won’t be in the future, then this approach is an option.

**Share some state between multiple boxes**
This approach, and the next one to be considered, seek to overcome the practical limitations of the “Intersect all paths with one box” by emulating the behavior of that single box across many.

---

3 Externally-speaking, anyway; if the platform is based on a particular family of Intel® chips, then to maximize performance the platform still needs to prevent the internal asymmetry that causes processor migration. More information about this topic, which is really quite interesting and important, is available in the Sandvine whitepaper *QuickPath Interconnect: Considerations in Packet Processing Applications*. 

---

7
Applying Network Policy Control to Asymmetric Traffic: Considerations and Solutions

In this approach, known in the industry as state-sharing, the packet flow remains unchanged (which has the benefit of the network behaving exactly as it would without the policy control devices) and the policy control devices exchange information about packets and flows they observe. In a common implementation of state-sharing, the first device that observes a flow (e.g., 5-tuple) informs the other boxes; in this manner the other boxes become aware of the flow’s existence.

When a packet corresponding to this flow passes through another box in the policy control deployment, that box communicates to the one that saw the initial packet. From that point on these two boxes exchange information and coordinate on policy enforcement, measurements, etc. In ideal conditions, state-sharing amounts to an information overhead of 2% to 6% of total traffic volume. This scenario is depicted in Figure 5. In this example, Box 1 sees the first chunk (Chk1) of a video stream and informs the other boxes. Box 3 sees the second chunk (Chk2) and informs Box 1. From that point on, Box 1 and Box 3 coordinate by sharing state information about the flow. Box 2 eventually stops paying attention (i.e., frees up state memory), either by timing out or due to an explicit notification from the other boxes.

In the ideal scenario presented in Figure 5, state-sharing is a great solution to the problem of routing asymmetry: functionality is preserved, costs are reasonable, and complexity is manageable. However, beneath the surface there are some severe limitations that become clearer upon deeper examination. One thing that must be understood about state-sharing is that it provides an approximation of precise policy control, but not perfect precision. A simple example makes this clear: if Box 1 and Box 3 are jointly implementing a policy of “shape this flow to 20 Mbps”, then they must be in constant synchronization to approximate the 20 Mbps as closely as possible. Since the packets for the flow are passing through both boxes in real-time, it is not possible to perfectly apply a 20 Mbps shaper unless the packets are held up inside the boxes. Plus, the more precise coordination that the CSP wants (i.e., the more accuracy the use case demands), the more overhead information must be exchanged, and the boxes venture above the advertised 2% to 6% amount. While traffic shaping is a common use case,

Figure 5 - State-sharing at work

4 According to the marketing material found online; the number of messages per second per box is a function of the number of new flows per second multiplied by two, plus some overhead for the early phase broadcasts
perhaps complete accuracy isn’t terribly important; what about prepaid charging? To avoid revenue leakage, the CSP must demand complete accuracy.

Additionally, the state-sharing story continues to break down when one considers stateless protocols such as DNS and UDP in general. With these protocols, flow-exhaustion is an ever-present danger: consider that some protocol flows such as DNS, DHT\(^5\), AAA, and DHCP are only two packets, which means the performance overhead of tracking individual flows is 100%. High velocity protocols such as DNS and DHT can be especially sensitive to latency, and in the case of a DNS attack, with state-sharing there is a real possibility of exhausting the solution’s ability to maintain state due to the underlying design.

Those shortcomings are why, in practice, flow synchronization applies only to TCP connections. It is important to note that in most networks, the majority of the network’s connections are UDP\(^6\).

In those examples, it all comes down to trade-offs - gain accuracy at the expense of overhead - and this reality might be tolerable. However, state-sharing has an Achilles heal\(^7\).

Figure 5 showed ideal flow behavior from the perspective of a state-sharing solution; Figure 6 shows a more common flow behavior. This scenario picks up where Figure 5 left off, and tracks the subsequent packet flows. Box 1 sees the third chunk of the video stream (Chk3) and Box 3 sees the fourth (Chk4); so far, everything is fine. But the fifth and sixth chunks (Chk5 and Chk6) go through Box 2.

Recall that Box 2 has disavowed all knowledge of this flow - as a result, neither Chk5 nor Chk6 are counted against the flow. And the ultimate result is that the policy control is fundamentally broken:

- All measurements pertaining to the flow are now invalid (e.g., volume, duration, etc.) because packets are missing
- All metrics based on those measurements are now invalid (e.g., quality, duration, etc.)
- The amount of unrecognized traffic for the overall deployment will increase since Box 2 cannot identify the packets that are part of the connection
- All policy enforcement (e.g., shaping) relating to this connection (i.e., either alone or in aggregate) fails, as do charging use cases
- Redirection to enable value-added services (VAS) and service function chains is no longer possible, since much of the traffic is no longer detected\(^8\)

Basically, the foundation of all network policy control uses cases - correctly identifying as much traffic as possible - is gone.

\(^5\) The distributed hash table (DHT) protocol (e.g., used by BitTorrent) provides a lookup service similar to a hash table - any participating node can retrieve the value associated with a given key
\(^6\) Take a look at your own network; typically we see UDP account for 50% to 70% of all connections
\(^7\) i.e., a critical, devastating, fatal flaw
\(^8\) Enabling value-added services with service function chains is examined in detail in the Sandvine whitepaper Value-Added Services and Service Chaining: Deployment Considerations and Challenges
Applying Network Policy Control to Asymmetric Traffic: Considerations and Solutions

Figure 6 - State-sharing fails completely when three or more elements see packets from the same connection

This issue results because Box 2 has no knowledge of the flow, so a potential solution is for Box 2 to maintain knowledge. However, this comes at the cost of consumed state memory; to truly overcome this issue, every box would need to maintain state on every flow, so this potential solution is not at all feasible.

As a result, state-sharing only works as advertised in an ideal scenario in which traffic can only take a maximum of two paths through the network.

It should also be noted that a consequence of the state-sharing approach is an N:2N redundancy model; that is, every single box depicted in the figures above is actually a pair of boxes. Since each box has a unique flow state table, each must be held on a redundant box.

Ensure core affinity

This approach bears similarities to state-sharing, and also seeks to overcome the practical limitations of the “Intersect all paths with one box” by emulating the behavior of that single box across many.

In this approach, often termed clustering, the packet flow remains unchanged (as with state-sharing); however, while state-sharing attempts to coordinate between processors, the clustering approach ensures that all packets associated with a particular flow, session, and subscriber are processed by the same processor core.

With clustering (Figure 7), regardless of the link on which a particular packet arrives at the entire deployment (i.e., at the cluster), that packet is carried to a specific processor core. If that core is on the box at which the packet arrived, then the redirection is carried out on switching fabric; if that core is on a different box, then the packet traverses the cluster links to arrive at the box housing the core. Once processed, the packet is returned to the exit interface corresponding to the original link, making clustering transparent to the network.
Clustering completely, rather than approximately, resolves all functional issues and ensures complete consistency with a fully symmetric deployment - everything works: policy enforcement, measurements, VAS-enablement, etc.

Critically, clustering works in any deployment, with any number of asymmetric links, with any forms of traffic - that means CSPs are able to make network changes without concern that the policy control will be impacted.

Additionally, because there is no shared state (i.e., clustering uses a “shared nothing” architecture), the redundancy model becomes N:N+1, which is significantly more cost effective than state-sharing. Practically, rather than keeping the “+1” unit sitting idle, it becomes part of the cluster and participates in the processing. In the event core, processor, or unit failure, the traffic is simply rebalanced to the remaining N units.

From a complexity standpoint, clustering is no more or less complex than state-sharing, as both approaches require connecting multiple units together. The cost incurred in providing the larger cluster links is offset by the increased efficiency (e.g., the N:N+1 redundancy model requires lower capital and makes much more efficient use of rack-space than state-sharing).
Conclusion

By design, all broadband networks exhibit routing asymmetry of one form or another; that is, traffic packets relating to the same flow can take different routes through the network. This asymmetry presents significant challenges for equipment (e.g., policy control platforms) that measure this traffic and take action.

First and foremost, a policy control solution must work when faced with routing asymmetry:

- the functions available when deployed in an environment with routing asymmetry should be exactly the same as those functions available if there was perfect symmetry
- the functions should behave precisely the same, and be as effective, in an asymmetric routing environment as they do and are with perfect symmetry

The criteria of cost and complexity are important, but are secondary to the fundamental issue or whether or not the solution actually works.

There are a handful of approaches to overcome asymmetry, but only three actually preserve complete functionality: deploying only where traffic is guaranteed to be symmetric, intersecting all possible paths with one (large) platform, and clustering to preserve processor core affinity.

Of these three approaches, only clustering offers a combination of versatility, efficiency, and an attractive redundancy model.

Summary of Approaches to Overcome Routing Asymmetry

The table below summarizes the approaches examined in this paper.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore routing asymmetry</td>
<td>This approach is not viable; even with an “asymmetry mode” enabled, it is technically impossible to accurately identify a significant portion of Internet traffic (e.g., HTTP content, tracker-based protocols including SIP, FTP, and many peer-to-peer clients, etc.)</td>
</tr>
<tr>
<td>Network/external approaches</td>
<td>Deploy only where traffic is guaranteed symmetric (e.g., at the extreme subscriber edge): this approach is often cost-prohibitive, but is favourable where the cost/benefit economics of the deployment make sense. Extend symmetry deeper into the network, then deploy there: this approach incurs great cost to undo much of the resilience and redundancy of the network, so is not viable.</td>
</tr>
<tr>
<td>Intersect all paths with one box</td>
<td>This approach is not appropriate for any CSP that wishes to avoid the real potential for a high-impact and widespread traffic outage. However, if a CSP is comfortable with the added risk, and is confident that scaling isn’t an issue now and, importantly, won’t be in the future, then this approach is an option.</td>
</tr>
<tr>
<td>State-Sharing: share some state between multiple boxes</td>
<td>State-sharing is an attractive story, but it cannot withstand the deployment requirements of real network. State-sharing does not work with UDP traffic, which frequently makes up the majority of total connections on a network, and does not work when traffic can take three or more paths through the network - this latter shortcoming severely restricts CSPs, as they must be very careful of network updates and changes when state-sharing is deployed. Additionally, because state is shared between units, this approach requires an</td>
</tr>
</tbody>
</table>
Applying Network Policy Control to Asymmetric Traffic: Considerations and Solutions

| N:2N redundancy model; as a consequence, a CSP must deploy twice as many boxes as are required for scale, and must consume valuable and expensive rack-space to house the redundant units. |

| Clustering: ensure core affinity | Clustering completely resolves all functional issues and ensures complete consistency with a fully symmetric deployment. Critically, clustering works in any deployment, with any number of asymmetric links, with any forms of traffic. This approach also provides a cost-effective and rack-space efficient N:N+1 redundancy model. |

Additional Resources

In addition to the resources cited in the footnotes throughout this document, please consider reading the Sandvine Technology Showcase *Policy Traffic Switch Clusters: Overcoming Routing Asymmetry and Achieving Scale*.