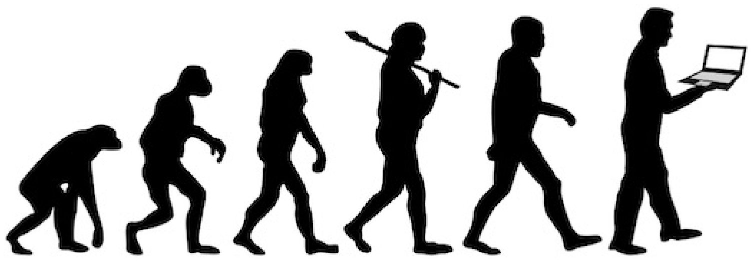
The Evolution of BGP NetFlow Analysis, Part 1

BY JIM MEEHAN, SOLUTIONS ENGINEER · MAR 7, 2016

Enabling comprehensive, integrated analytics for network visibility



Clear, comprehensive, and timely information is the essential prerequisite for effective network operations. For Internet-related traffic, there's no better source of that information than NetFlow and BGP. But while these protocols have been around for a couple of decades, their potential utility to network operators was initially unrealized, and the process of exposing more value has been a long, gradual evolution. The journey started with simply making the data available. It then progressed to the development of collection and analysis techniques that make the data useable, but with real limitations. The next leap forward has been to transcend the constraints of legacy approaches, both open source and appliance, using big data architecture. With a distributed, multi-tenant HA datastore in the cloud, Kentik has created a SaaS that enables network operators to extract far more practical value from NetFlow and BGP information. In this series we'll look at how we got from the first iterations of NetFlow and BGP to the fully realized network visibility systems that can be built around these protocols today.

In the beginning...

Border Gateway Protocol (BGP) was first introduced in 1989 to address the need for an exterior routing protocol between autonomous systems (AS). By 1994 BGP4 had become the settled protocol for inter-AS routing. Then in 1996, as the Internet grew into a commercial reality and the need for greater insight into IP traffic patterns grew, Cisco introduced the first routers featuring NetFlow. Support for BGP was added in 1998 with NetFlow version 5, which is still in wide use today.

Support for BGP in NetFlow v5 enabled the export of source AS, destination AS, and BGP next hop information, all of which was of great interest to engineers dealing with Internet traffic. BGP next hop data provided the possibility for network engineers to know which BGP peer, and hence which neighbor AS, outbound traffic was flowing through. With that insight, network engineers could better plan their outbound traffic.

A key use case for next hop data arises when determining which neighbor ASes to peer with. If both paid-transit and settlement-free peering options are available, and those options all provide equivalent and acceptable traffic delivery, then you'll want to maximize cost savings by ensuring that the free option is utilized whenever possible. Armed with BGP next hop insights, engineers can favor certain exit routers by tweaking IGP routing, either by changing IGP link metrics or (with a certain more-proprietary protocol) by employing weights.

Kick AS and take names

While NetFlow 5's BGP support was helpful with the above, simple aggregation of the supported raw data left many use cases unaddressed. Knowing ASNs is a first step, but it's not that helpful unless you can also get the corresponding AS_NAME so that a human can understand it and take follow-up action. In addition, engineers wanted more visibility into the full BGP paths of their traffic. For example, beyond the neighbor AS, what is the 2nd hop AS? And how about the source and destination ASes? NetFlow's v5 BGP implementation didn't offer that full path data, and while v9 introduced greater flexibility it still provided only a partial view.

In the early 2000's, a first generation of vendors figured out how to address this gap by collecting BGP routing data directly and blending it with NetFlow. This was done by establishing passive BGP peering sessions and recording all of the relevant BGP attributes. A further enhancement came from integrating information from GeoIP databases to augment the NetFlow and BGP data by providing source and destination IP location. Now, with a GUI tool, network engineers could make practical use of NetFlow and BGP information.

NetFlow's BGP gaps were addressed by direct collection of BGP routing data.

These enhancements helped engineers with a number of use cases. One was DDoS detection. Looking at a variety of IP header and BGP data attributes on inbound traffic, you could use pattern-matching to detect volumetric as well as more-nuanced denial of service attacks. Another use case was to find opportunities for transit cost savings, including settlement-free peering, by looking at traffic going through 2nd and 3rd hops in the AS_PATH. For companies delivering application traffic to end-users, the ability to view destination AS and Geography helps in understanding how best to reach the application's user base.

Struggling to keep up

The integration of fuller BGP data with NetFlow and other flavors of flow records created a combined data set that was a huge step forward for anyone trying to understand their Internet traffic. But at the same the overall volume of underlying traffic was skyrocketing. Constrained by the technologies of the day, available collection and storage systems struggled to keep up, and network operators were prevented from taking full advantage of their richer data.

One key issue was that the software architecture of Netflow-base visibility systems was based on scale-up assumptions. Whether the software was packaged commercially on appliances or sold as a downloadable software-only product, this meant that any given deployment had a cap on data processing and retention. With most of the software functionality written to optimize single-server function and performance, stringing together a number of servers only yielded a sum-of-the-parts in aggregate price performance.

Another issue was that the databases of choice for early NetFlow and BGP analysis implementations were either proprietary flat files, or, even worse, relational databases like MySQL. In this scenario, one process would strip the headers off of the NetFlow packets and stuff the data fields into one table. Another process would manage the BGP peering(s) and put those records into another table. A separate process would then take data from those tables and crank rows of processed data into still more tables, which were predefined for specific reporting and alerting tasks. Once those post-processed report tables were populated, the raw flow and BGP data was summarized, rolled up, or entirely aged out of the tables due to storage constraints.

With legacy NetFlow datastores, running non-standard reports was possible but painfully slow.

While it was possible in some cases to run non-standard reports on the raw data, it was painfully slow. Waiting 24 hours to process a BGP traffic analysis report from raw data was not uncommon. In some cases, you could export that raw data, but given the single-processor nature of the software deployment, and considering all of the other processes running at the same time, it was so slow to do so that 99% of users never did. You might have to dedicate a server just to run those larger periodic reports.

Steep deployment costs

Yet another major issue was the cost of deployment. NetFlow and BGP are both fairly voluminous data sets. NetFlow, even when sampled, produces a lot of flow records because there are many short-lived flows. Whenever a BGP session is established or experiences a hard or soft reset, the software has to ingest hundreds of thousands of routes. Plus, you have the continuous flow of BGP UPDATE messages as route changes propagate across the Internet.

Using single-server software, you may end up needing a bunch of servers to process all of that data. If you buy those servers pre-packaged with software from the vendor, you'll pay a big mark-up. Consider a typical 1U rackmount server appliance from your average Taiwanese OEM. Raw, the cost of goods sold (COGS) may be anywhere from $1,000.00 to $2,000.00, but loaded with software, and after a basic manufacturing burn-in, you can expect to pay a steep $10K to $25K, even with discounts. And even if you're buying a software-only product that isn't pre-packaged onto an appliance, you still have the cost of space, cooling, power, and — most importantly — overhead for IT personnel. So owning and maintaining your own hardware in a scale-up software model is still expensive from a total cost of ownership (TCO) point of view.

Considering the limited, inflexible reporting you get for these high costs, most users of legacy NetFlow and BGP analysis tools have been left hungry for a better way. In part 2 of this post, we'll look at an alternative approach based on big data SaaS, and consider how this new architecture can dramatically boost the value of BGP and NetFlow in network visibility, operations, and management.

Ready for more information?
Please email us at info@kentik.com or visit us at www.kentik.com.