Towards a Predictive Internet A new world, with new challenges

Article · September 2021

CITATIONS
0

READS
37

1 author:

Jp Vasseur
Cisco Systems, Inc
22 PUBLICATIONS **3,396** CITATIONS

SEE PROFILE

Towards a Predictive Internet

JP Vasseur (jpv@cisco.com), PhD - Cisco Fellow, Engineering Lead – Release v1.1 - September 2021

Since the early days of the Internet (Arpanet in 1970), the topic of Routing Protocol Convergence Time (time required to detect and reroute traffic in order to handle a link/node failure) has been a top-of-mind issue. A number of protocols and technologies have been developed and deployed at a large scale with the objective of improving overall network reliability. Although such approaches have dramatically evolved, they all rely on a reactive approach: upon detecting a network failure, the traffic is rerouted onto an alternative path. In contrast, a proactive approach would rely on a different paradigm consisting in rerouting traffic before the occurrence of a predicted failure onto an alternate path that meets application Service Level Agreement (SLA) requirements.

Myth or reality? The notion of prediction refers to the ability to anticipate/forecast a network state (such as a dark/grey failure) that would impact the application experience, but also to determine whether an alternative path that is free of failures exists. This short white paper introduces the emergence of a Predictive Internet using learning technologies along with few results derived from the deployment of such technology at scale.

A new world, with new challenges

Network recovery has been a topic of high interest in the networking community since the early days of the Arpanet in 1970. Nonetheless, the paradigm has not changed much: first, a failure must be detected, followed by traffic rerouting along an alternate path; such a path can be either pre-computed (i.e., protection) or computed on-the-fly (i.e., restoration).

Let's first discuss "Failure detection". The most efficient approach is to rely on inter-layer signaling whereby lower layer may be able to detect a layer-1 failure (e.g., fiber cut) triggering a signal across layers. Unfortunately, a large proportion of failures impacting a link or a node are simply not detectable by lower layers. Thus, other techniques such as Keep Alive (KA) messages are being used. There is clearly no shortage of KA mechanisms implemented by routing protocols such as OSPF, ISIS or protocols such BFD. KA have their own shortcomings related to their parameter settings: aggressive timings introduce a risk of oscillations of traffic between multiple paths upon missing few KA messages, a real challenge on (lossy) links where packet loss is not negligible, which introduce high risks of oscillations. Once the failure is detected, a plethora of techniques can be used such as Fast IGP convergence (OSPF or ISIS using fast LSA/LSP generation, fast triggering of SPF, incremental SPF to mention a few), MPLS Fast Reroute (using a back 1-hop tunnel for link protection of multi-hop backup tunnels for node protection), IP Fast Reroute (IPRR) or other protection mechanisms used at lower layers (1+1 protection, 1:N, etc.) have proven their efficacy at minimizing downtime. Such recovery technologies have allowed for a fast convergence time in the order of a few milliseconds, while guaranteeing equivalent SLA other alternate paths (e.g., MPLS TE with bandwidth protection).

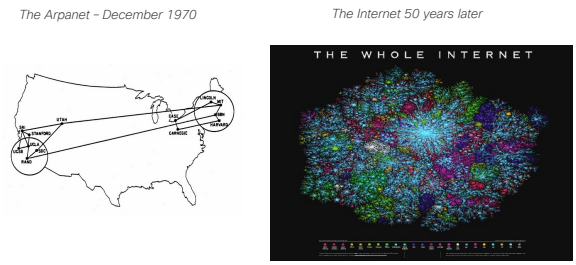

Figure 1 -Evolution of the Internet since the Arpanet in 1970

Unfortunately, there is a large category of failures highly likely to impact application experience that remain largely undetected. The notion of grey failures has been covered in a companion document [grey-failure]. These grey failures may have a high impact on application experience because of packet loss, delay or jitter without breaking the link/path connectivity (and thus they are not considered by the aforementioned technologies as "Failure"). In this case, most -- if not all -- KA-based approaches would fail, leaving the topology unchanged and the traffic highly impacted even though a preferable alternate path may exist.

Existing solutions such as Application Aware Routing (AAR) rely on the use of network-centric mechanisms such as BFD and HTTP probes to evaluate whether a path meets the SLA requirements of an application using a so-called SLA Template. The most common approach consists in specifying some hard threshold value for various network central KPIs such as the delay, loss and jitter averaged out over a given period of time (e.g., Delay average computed over 1h should not exceed 150ms one-way for voice). Such templates may be highly debatable and the use of statistical moments such as average and 90th percentile values have the undesirable effect of smoothing out the signal and lose the necessary granularity necessary to detect sporadic issues that impact the user experience. One may then shorten probe frequencies, use higher percentiles but the granularity is unlikely to suffice for the detection Grey failures impacting the user experience.

Still, AAR is a great step forward when compared with the usual routing paradigm, according to which traffic rerouting occurs only in presence of dark failures (i.e., path connectivity loss). Note that AAR is a misnomer since the true application feedback signal is never taken into account for routing; path selection still relies on other static network metrics and SLA templates are a posteriori assessed and verified as explained above. AAR is *reactive* (the issue must first take place to be detected and must last for a given period of time for a rerouting action to take place) with no visibility on the existence of a better alternate path: rerouting is triggered over alternate paths with no guarantees that SLA will be met once rerouting the traffic.

Path computation in the Internet

Several routing protocols used within Autonomous Systems (AS) also referred to as IGP (Interior Gateway Protocol) have been developed such as OSPF, IS-IS or EIGRP, whereas BGP (an Exterior Gateway Protocols) has been widely deployed to exchange routes between ASes for the past four decades. BGP has been scaling remarkably well and as

of 2021 routing tables comprise up to 860K IPv4 prefixes and 110K IPv6 prefixes.

How are paths/routes being computed in the Internet? This is the task of routing protocols. IGP such as ISIS or OSPF make use of a Link State DataBase (LSDB) to compute shortest paths using the well-known Dijkstra algorithm. Link weights are static reflecting some link properties (link bandwidth, delay, …). More dynamic solutions using control place Call Admission Control (CAC) such as MPLS Traffic Engineering allows for steering traffic along Labeled Switched Path (LSP) computed using constraint shortest paths using a distributed head-end driven Constrained SPF (Shortest Path computation). Alternatively, such TE LSPs may also be computed using a Path Computation Element (PCE) for Optical/IP-MPLS layers, intra and inter-domain. Paths *between* AS are selected using BGP policies.

A path between a source and destination across the Internet will likely cross a number of ASes, each managed by distinct administrative domains using disparate traffic engineering policies, making the optimization of end-to-end path extremely challenging, if not impossible.

In the context of this paper, the Predictive Internet refers to the path selection process at the *edge* the network. Consider the *Figure 2 Edge Path Selection in a Predictive Internet*: in this classic example, an remote site (called "Edge") is connected to a Hub via SD-WAN using several tunnels having various properties. It is also quite common for the edge to be connected to the public Internet via an interface sometimes called DIA (Direct Internet Access). In such a situation the traffic destined to (for example) a SaaS application S can be sent along one of the tunnels from the edge to the Hub (backhauling) or via the Internet using IP routing, or even using a (GRE or IPSec) tunnel to a Security Cloud

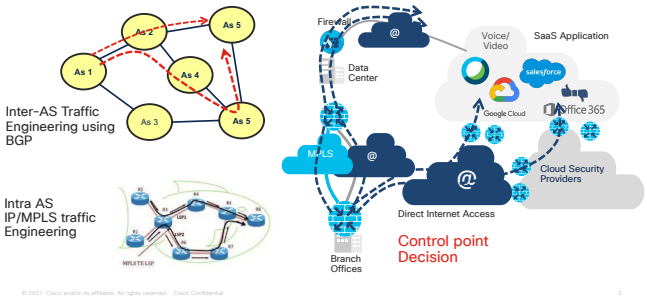

Figure 2 Edge Path Selection in a Predictive Internet

The notion of Path selection in a Predictive Internet refers to the ability to predict (dark/grey) failures for each of the path from edge to destination, on a per-application basis.

Lack of Learning in the Internet?

It is an unescapable fact: most of the control plane networking technologies do not incorporate learning from the past but rather focus on the ability to react as quickly as possible. Imagine a human brain incapable of learning and rather just reacting. The human brain is without a doubt the most advanced learning engine and the Hebbian theory related to synaptic plasticity has been a key principle in neuroscience "What fire together wire together"; thanks to synaptic plasticity neural networks are formed (wire together) dynamically thus allowing us to *learn* and also unlearn (for example thanks to synaptic downscaling during sleep).

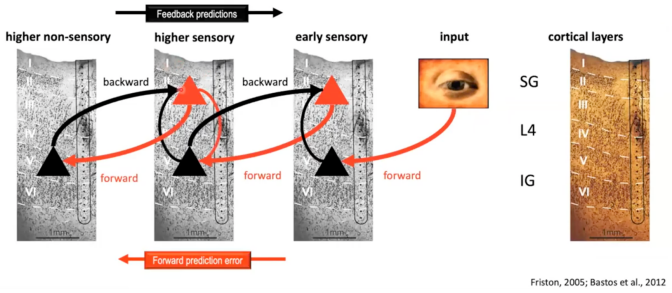

Figure 3 Brain Predictive Coding across cortical layers

The brain is an impressive predictive engine: the simple action of grabbing an object involves a complex series of predictive actions (e.g., just anticipating the shape of the object). As shown in Figure 3 *Brain Predictive Coding across cortical layers*, some areas of the brain get inputs from other regions, provide predictions that get adjusted according to sensing, with a number of theories of the critical topic of "Predictive Coding" such as the Free Energy Principle first explored using a Bayesian approach by Friston in 2005. Similarities with a predictive Internet will be shown later in the document. Other forms of predictions involving hierarchical structures (Hierarchical coding) with interaction between sensing and prediction (in different brain areas communicating via different layers of the neocortex) are also very well-known in vision, auditory pathways and Natural language processing. Other forms of higher level predictions are also known to be performed in the PreFrontal Cortext (PFC).

How about learning in the Internet?

As a matter of fact, data has not really been used for designing protocols that are capable of learning. There are some minor exceptions such as packet retransmissions (backoff) at lower layer of transport layers, ability to learn the instantaneous bandwidth along a given path or the use of hysteresis, … Various protocols have adaptive behaviors according to a very recent past, without true learning/modelling. Most control plane operations mostly focus on the ability to react.

Towards a Predictive Internet

Quoting Niels Bohr: *"It is hard to make predictions, especially about the future"*. Cisco has invested considerable research to investigate the ability for networks to combine a proactive approach with a reactive approach. To that end, an unprecedented analysis has been made on millions of paths across the Internet, using different networking technologies (MPLS, Internet), access types (DSL, fiber, satellite, 4G), in different regions and Service Provider networks across the world. The objective was first to determine the dynamicity of a vast number of paths, along with application experiences. [internet-dynamics] provides an overview of such analysis. *Figure 4 Models of Path dynamics in the Internet ([Internet-dynamics])* shows a few approaches for data path statistical models.

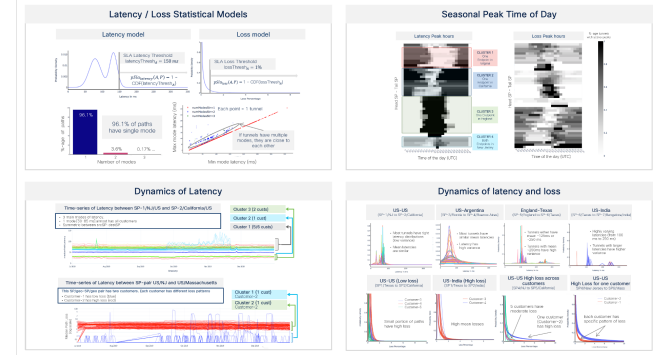

Figure 4 Models of Path dynamics in the Internet ([Internet-dynamics])

More advanced models relying on a variety of features and statistical variables have been performed (e.g., spectral entropy, welch spectral density, MACOS, …) along with their impact on application experience.

The fact that path "quality" across the Internet is diverse and varies over time is not new. But the main key take-away lies in the ability to quantify across space and time using a broad range of statistical and mathematical analysis. As pointed out earlier, in a cloud, highly virtualized world where the network keeps changing and applications constantly move, it has never been so important to equip the Internet with learning capability.

What is Prediction/Forecasting?

Simply put, predicting/forecasting refers to the ability to anticipate events of interest such as failures; thanks to the use of a model trained with historical data.

The most common question is "*Which algorithm are you* using?". Unfortunately, there is no one-size-fits-all, and Predictive Internet undoubtedly relies on a collection of algorithms and technologies used with specific objectives in mind. The forecasting horizon is one of the most decisive criteria. Forecasting with very high time granularity (e.g. months) is extremely smooth and easily captured by simple algorithms. Such an approach is referred to as "*trending*": simple and robust but of relatively minimal value for forecasting. On the other side of the spectrum, a system capable of forecasting a specific event (e.g., failure) impacting the user experience is far more interesting and challenging. Even for such a well-defined problem, various approaches may be used:

- The time series may first be categorized using hand-crafted criteria applied to various characteristics such as the entropy, welch spectral density, … using various rules-based and ML-based clustering algorithms,
- Then ML algorithms such as Recurrent Neural Network (RNN) like Long Short-Term memory (LSTM) may be used for forecasting, using local, per-cluster or even global models.

We have also developed other approaches such as State Transition Learning for forecasting failure events by observing the prominent subsequence of network state trajectories that may lead to failure.

Mid- and Long-term prediction approaches ought to be considered whereby the system performs training on historical data and models the network to determine where/when actions should be taken to adapt routing policies and configuration changes in the network in light of the observed performance and state of the network (i.e., Internet behavior). Such predictions then allow for making recommendations (e.g., change of configuration or routing policies) that will improve the overall network SLO and application experience. Mid- and Long-term predictions have proven to be highly beneficial, although less efficient than short term recommendations that deal with short term predictions and remediations. Such systems must take into account a series of risk factors including the stability of the network and traffic pattern in order to minimize the risk of predictions that would be quickly outdated. This contrasts with a short-term predictive engine used for "quick fixes" and avoidance of temporary failures that would be enabled with full automation (a topic that will be discussed in a companion document).

Forecasting accuracy is a recuring topic. Any forecasting system will make prediction errors. However, such a system can be designed so as to make trade-offs between False Positive (FP) and False Negative (FN). FP means that a failure predicted does not occur whereas a FN refers to the opposite situation. For example, a Machine Learning (ML) classification algorithm may be tuned to deal with the well-known Precision/Recall tension where Precision=TP/(TP+FP) and Recall=TP/(TP+FN). In other words, the algorithm must be tuned to favor either Precision or Recall. Cisco's predictive engine favours Precision over Recall, a safe approach for highly minimizing the risk of FP, while with current reactive approach there is no prediction, so the rate of FN=100%. In other words, most prediction of the event should be correct even if all events cannot be predicted.

Many other dimensions must be taken into account in such a predictive system. For example, will the proactive action of rerouting traffic impact the traffic already in place along the alternate path?

In a live system, such as the one Cisco has developed, other criteria must be taken into account and time granularity is of the utmost importance both for telemetry gathering and time to react (triggering close loop control) with tight implications on the architecture.

Are all failures predictable?

From a pure theoretical standpoint, yes, since random events are extremely rare in nature, but of course the reality is different. In most cases, events indicative of some upcoming failures usually exist but they are not always monitored by existing systems. Moreover, the timeframe is not always compatible with the ability to trigger some actions (even a fiber cut may be predicted by monitoring the signal in real-time but some nanoseconds before the damage leading note enough time to trigger recovery action, even if a predictive signal exists). In reality, the ability to predict events is driven by the following factors:

- Finding "Signal" in telemetry with high SNR
- Computing a reliable ML model with sufficient Precision/Recall
- Designing an architecture at scale supporting a Predictive approach (this last aspect should not be overlooked; there is a considerable gap between an experiment in a lab and the scale of the Internet).

Predicting consists in finding signals used to build a model and producing a given outcome (e.g., component X will fail within x ms, or probability of failing of component Y is Pb) using classification and/or regression approaches. The ability to predict raises a number of challenges Cisco managed to overcome; thanks to a decade of deep expertise in Machine Learning and analytics platforms.

Is the Predictive Internet a promising avenue?

Yes, it is. Such a system has been in production in 100 networks around the world, doing real-time predictions for several months and has proven to perform predictions highly improving the overall network SLO and application experience. Although the details of the exact architecture, telemetry (and technique for noise reduction), algorithms, training strategies are out of the scope of this document, it is worth providing several examples of overall benefits that a predictive system can bring. Figure 5 *number of minutes of application SLA violation (red) and number of minutes of SLA violation a predictive system would have avoided (real values in an existing network)* shows the overall number of minutes with SLA violation observed in a 30-day period on a network (in Red). Next to it are the number of minutes of SLA violation that would have been avoided ("saved") using the predictive engine, which managed to

accurately predict such (grey) failures (application SLA violation) but also finding an alternate free of SLA violation in the same network (without adding any additional capacity).

Figure 5 number of minutes of application SLA violation (red) and number of minutes of SLA violation a predictive system would have avoided (real values in an existing network)

For the sake of illustration, let's explore examples of such predictions (i.e., SLA violation) that were correctly predicted. The next set of figures show when a failure was predicted (see green dots on the time line). Various time series show after the predictions the loss, delay and jitters. In ocean blue is the default path programmed on the network, in the dark blue color is the path recommended by the predictive engine, thus validating that the prediction was indeed correct.

In the first example Figure 6 *Prediction of SLA violation because of packet loss on a path between Costa Rica and Malesia: 90% of SLA violation avoided* (a path between Malaysia and Costa Rica), many minutes of traffic (11,000 minutes of voice traffic) with SLA violation could have been avoided (green) for traffic sent along Business Internet path (ocean blue) by proactively rerouting traffic onto an existing bronze internet path (a priori with less strict SLA) (dark blue).

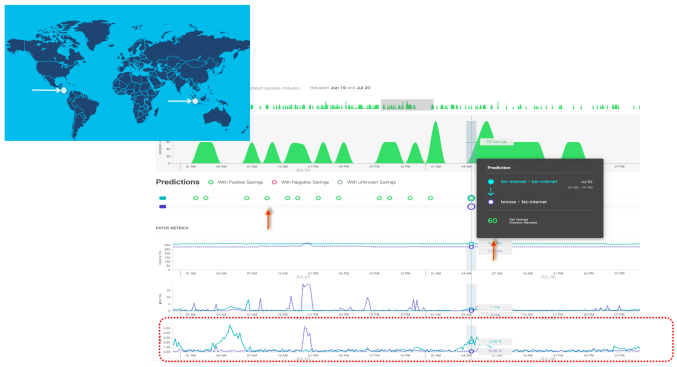

Figure 6 Prediction of SLA violation because of packet loss on a path between Costa Rica and Malesia: 90% of SLA violation avoided

*Figure 7 : Prediction of high packet loss spikes on an MPLS path*shows a prediction of packet loss spike (way before they take place) along a short-distance MPLS paths resulting in 82% of SLA violation over a 30-day period.

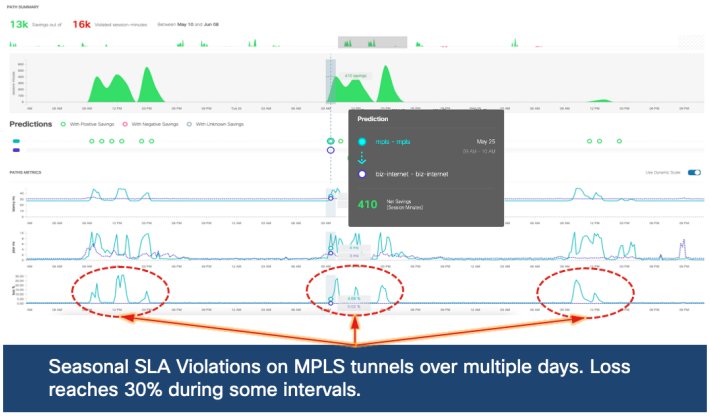

Figure 7 : Prediction of high packet loss spikes on an MPLS path

Figure 8 Prediction of sporadic packet loss shows another example of prediction of a sporadic packet loss (17%) in Australia.

Figure 8 Prediction of sporadic packet loss

The number of such successful predictions is endless and a few examples are provided for illustration. Even though all failures cannot be predicted, any single prediction means that an issue is being proactively avoided, while all other (unpredicted) failures are being handled by the reactive system, making the combination of both approaches a huge step forward for the Internet.

Are predictions' accuracy and efficiency consistent across networks and time? No, as expected. First of all, each network has its peculiarities in terms of topology (and built-in redundancy), traffic profiles, provisioning and access types to mention a few. A predictive engine may then exhibit different level of efficiency, as expected. Furthermore, the objective is not just to Predict but also to find some alternate path fee of SLA violation. Consequently, the built-in redundancy will be a key factor. A very interesting fact that has been observed is that prediction do vary significantly over times as the Internet evolves (e.g., failures, capacity upgrades, new peering, …) requiring constant learning and adjustment.

Figure 9 *Number of minutes of traffic with SLA violation avoided thanks to accurate predictions on six worldwide networks* shows for multiple

regions of the world and across multiple networks, the number of accurate predictions and ability to find alternate path (with measures such as the number of minutes of traffic saved from SLA failures) and their variation other times.

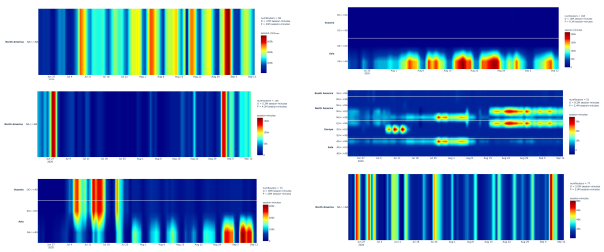

Figure 9 Number of minutes of traffic with SLA violation avoided thanks to accurate predictions on six worldwide networks

The Y-axis shows the regions of the world and X-axis how it varies over time. It can be noticed that the number of traffic saved varies significantly across networks and over time. On the top left corner, a network where the number of failures accurately predicted and avoided varies from low the high across all regions. On the top right another network where most of the failures avoided are located in a given region and tends to exhibit some form of seasonality. One can observe yet another pattern on the bottom left network where at some point most of the avoided failures were in a given region and later in another region. It is also worth noting that predictions are constantly adjusted thanks to continuous learning: indeed, the Internet and other SP networks are highly dynamics and experience failures, new peering agreements take place, SaaS applications do move across the Internet and traffic loads evolve.

As often, there is no one-size-fits-all algorithms capable of predicting (grey) failures. Each algorithm has specific properties related to the type of telemetry used to train the model, the forecasting horizon (itself coupled with the availability of telemetry) and of course the overall efficacy. The topic of forecasting accuracy will be covered in a separate document as there are many tradeoffs to be evaluated and certainly no single efficacy metric.

For the sake of illustration Figure 10 Variability of prediction accuracy per region per path for a given algorithm shows a performance accuracy metric for multiple paths in the world for paths with different characteristics. The metric for measuring the accuracy is outside of the scope of this document but the point is to show that accuracy greatly varies (in this particular example, a lo value is indicative of higher performance).

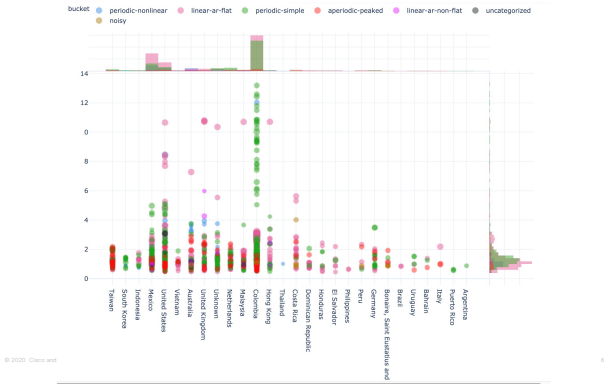

Figure 10 Variability of prediction accuracy per region per path for a given algorithm

Figure 11 Correlation between forecasting accuracy and path characteristics for a given algorithm provides another view of such variability of predictive accuracy; one can observe the relationship between forecasting accuracy and some properties of the path such as the entropy or level of periodicity.

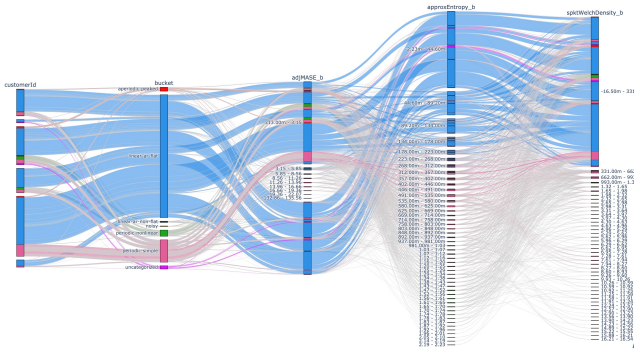

Figure 11 Correlation between forecasting accuracy and path characteristics for a given algorithm

Conclusion and Next Steps

Without a doubt, adding learning capabilities to the Internet will increase the overall network SLO and application experience and may arguably be overdue. Real data driven experiments have shown that such an approach will complement the reactive approach that has governed the Internet for the past four decades. Although there is no one-size-fit-all approach, various statistical and ML driven models have shown the possibility to predict future events and take proactive actions for short-term and long-term predictions with high accuracy, avoiding a high number of failures that would have significantly impacted the user experience. The path towards a Predictive Internet will take place over several years and is far from being over but coupled with Autonomous networks (Self-Learning/Healing networks), this approach could be one of the most impactful technologies for the Internet. Many more innovations are in the works.

Acknowledgement

I would like to express my real gratitude to several key contributors I have been working with for a number of years: Gregory Mermoud, Vinay Kolar and PA Savalle with whom many ML/AI innovations gave birth to novel innovations for networking (Wireless Anomaly Detection with root causing, Self Learning Networks, detection of spoofing attacks. …). I would like to acknowledge the work of several highly talented engineers: Mukund Raghuprasad and Jurg Diemand to mention a few who had a major contribution to this work. Needless to say, that a close collaboration with a number of customers in the world has allowed for such an innovative work.

References

[grey-failures] JP Vasseur, “From Dark to Grey Failures in the Internet”

[internet-dynamics] J. Vasseur and V. Kolar, *Large-scale Internet Path modelling and applications,* 2021.