SUNET

ALL I WANT FOR CHRISTMAS IS … LARGE BGP COMMUNITIES.

Dear Santa,

All we want for Christmas, is Large BGP Communities.

There is video evidence that such requests have been granted before:

Let start from the beginning. BGP Communities have been around since the dinosaurs roamed the earth and were standardized in RFC 1997. RFC 1997 has been implemented and is used actively in essentially all BGP-based networks. You can use BGP Communities for a various of tasks, you can mitigate DDoS with a community that tells your routers to trigger a reroute into a scrubbing VRF. You can use BGP Communities to not announce a certain prefix in Hungary. You can use BGP Communities to prepend a prefix 3x times when sending it to peer X but not to peer Y and many more common choices to influence the flow of traffic.

We can take a look of how we do it in AS2603 (NORDUnet) for example. Which is a more heavy users of BGP Communities then what SUNET is so serves a better example.

This is example of origin-communities that identifies from which customer a certain route is learned from.

```
community NORDUnet-SUNET 2603:1653
community NORDUnet-FUNET 2603:1741
community NORDUnet-UNINETT 2603:224
```

We also set communities on where peering-routes is learned from, for traceability and being able to reuse them as triggers.

```
community IX-AMSIX members 2603:64805
community IX-DECIX members 2603:64807
community IX-NETNOD members 2603:64801
```

With this, customers can for example filter that they do not want routes learned via AMS-IX.

It could also work the other way around with traffic engineering communities. A customer could use these to blackhole, or selectively blackhole their prefixes from the Internet.

```
community no-commodity_no-peers members 2603:666;
community no-commodity members 2603:664;
community no-commodity_no-non-nordic-peers members 2603:665;
```

Here the customer could either tag their prefix with the 2603:666 which is a full blackhole. Using this would withdraw the customer prefix on all peers and to transit making them unreachable from the Internet. 2603:664 is a selective blackhole that only withdraws the prefix from transit and 2603:665 will still announce the prefix in Scandinavia but not to transit and not to peers outside of Scandinavia.

There is also traffic engineering communities that is very specific and is built on a per-peer basis.

```
community nordunet-noadv-hibernia members 65500:5580;
community nordunet-prependx1-hibernia 65501:5580;
community nordunet-prependx2-hibernia 65502:5580;
```

These are four examples of traffic engineering communities that either prepend or withdraws the prefix of your choice towards a specific peer. This does not follow the 2603:xxx construct since we run out of logical namespace and had to start over in the private allocation space and also to get coherent naming scheme. 65500, 65501, 65502, 65503 you can append with any peer we have and achieve the desired outcome. 65500:1273, 65500:714, 65500:20940 list goes on and on.

Remember now that RFC 1997 is using a single 32bit value displayed as two 16bit values separated by a colon (:). If you apply at your local RIR for a ASN today you will most likely get 32bit ASN since we are all almost all out of usable 16bit AS-numbers, there is a few left at the local RIRs (since IANA has nothing left) but don't count on getting one. And if you get a 32bit ASN today you will not be able to reap the same benefit as a 16bit ASN would, because of to little space in a RFC 1997 community.

Lets say that AS2603 wants to peer with AS196752 over AMS-IX. What we typically do when establishing new peers is to put up a "identification" Community on ingress policys, prepend/discard-communities on egress policys, group these communities into the regional and continental communities and a few more. None of this works with a 32bit ASN today without doing workarounds and offset the naming-scheme, what we have done so far is to do a translation into a private allocation 16bit ASN to symbolize the 32bit ASN. This scales horribly and will eventually create collision in the namespace and it needs offline documentation to make any sort of sense.

So what do we need? We just need the classic RFC 1997 communities to become somewhat larger so they can fit all information we need. There has been two previous attempts to fix this (not counting RFC 4360 since it doesent fix 32bit problems and was not originally designed to fix this problem, but rather VPN problems). First out is Flexible BGP Communities draft-lange-flexible-bgp-communities which didn't go anywhere. Decent academic idea, but no consensus was reached nor any working implementations. Next out was Wide BGP Communities Attribute draft-ietf-idr-wide-bgp-communities. Still an active draft, but suffers from feature-creep, is overly complicated (draft is 26 pages already) and that it has no stable spec or fully functioning implementations. Both of these ideas would've solved the operator problem, but also adds a lot of other features which we may or may not need. As i said, we just need the BGP Community to become bigger but keep the simplicity, so it's actually implementable and deployable.

This is where Large BGP Communities comes along and saves the day. draft-ietf-idr-large-community describes what we really need. Just **larger** BGP communities. Instead of the `16BIT:16BIT` in RFC 1997, we'll now have a new scheme of `32BIT:32BIT:32BIT`, no more and no less. It is a very elegant solution to a long standing problem. Since Large BGP Communities is nothing else then just bigger buckets to put data in it fits very well into already established routing policies. You'll just need to make it
compatible with the newer recommended notion to use ASN:FUNCTION:PARAMETER, where as the old scheme was usually ASN:FUNCTION. Our community 2603:64806 we use to mark a prefix learnt over LINX, in Large BGP Communities this could for example be expressed as 2603:0:64806 if just doing a straight up translation (where i advocate to take a quick read on draft-ietf-grow-large-communities-usage to see best-practice implementations and get new tricks).

Making route-policys for 32bit ASN without Large BGP Communities

It is also easy to implement, which was not the case for either *-wide-* or *-flexible-* and hence why we do not see these efforts taking off. Large BGP Communities already has real functioning implementations out there, take a look at INEX for example that has upgraded their route-servers running BIRD to support Large BGP Communities and also has a new policy construct for this.

POLICY ACTION	COMMUNITY
Announce a prefix to a certain peer	`43760:1:peer-as`
Prevent announcement of a prefix to a certain peer	`43760:0:peer-as`
Announce a prefix to all peers	`43760:1:0`
Prevent announcement of a prefix to all peers	`43760:0:0`

This looks awesome! easy and recognizable scheme that anyone can understand, and most importantly, can also serve a 32bit ASN just as good as a regular 16bit. Woho go **INEX**!

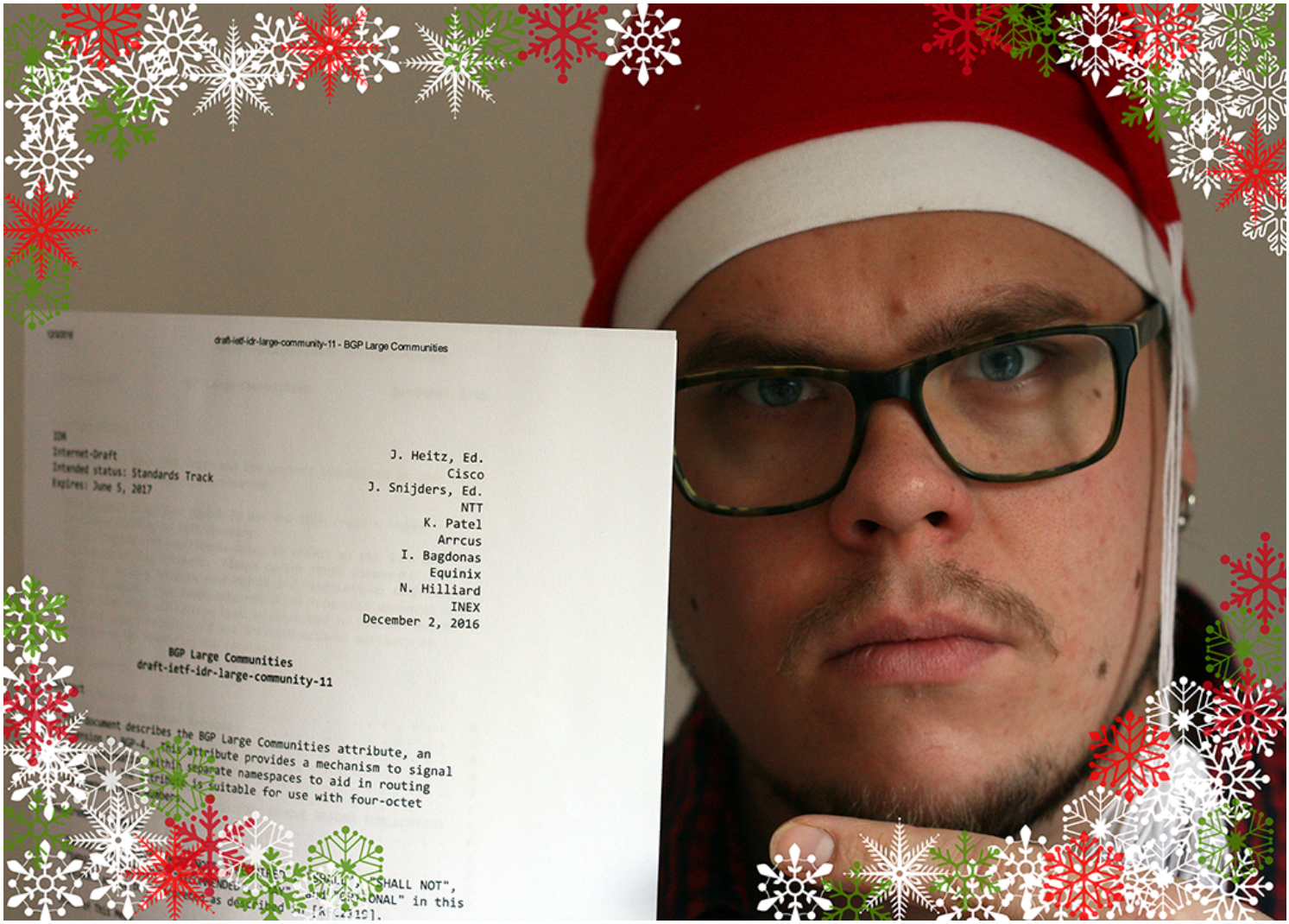
The current progress of Large BGP Communities inside the IETF is extraordinary. Its already almost in IESG review and currently IETF last call is going on, the support for the draft is very good. There is already plenty of fully working implementations out there that fully support Large BGP Communities which wildly help with the progress, but it does not end there. The community as a whole needs to continue working on this, and to make this a universal and globally accepted standard we need to tell our vendors about this. The open-source community is already reeled in to the boat and the big BGP daemons out there such as OpenBGPD, BIRD, ExaBGP, GoBGP, Quagga etc is already done. Now we just need the vendors that does boxes to also understand that this is what we need.

The Service Provider vendors such as Cisco, Juniper, Huawei, Arista, Nokia, Brocade etc need to be reminded about this and motivated to provide us with the code, but not only them, we can't forget about the datacenter either, edge-networking and

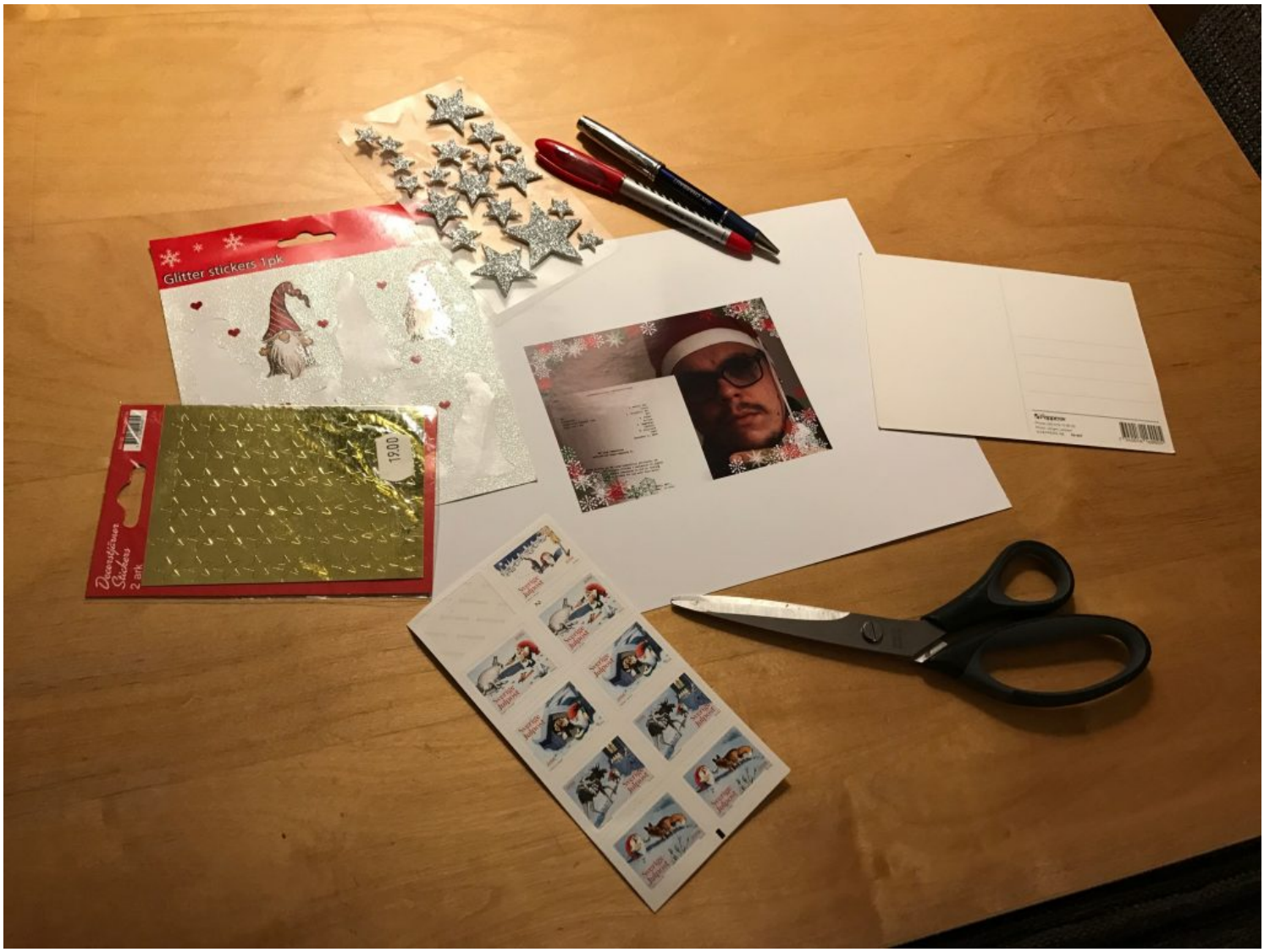
datacenter grows into becoming the same market so we need this on their products as well. If you are a customer of Cumulus, IP-infusion, Pica8, Openswitch and others, remind them to. We also need support in our tools that uses communities to base information upon, pmacct, deepfield, kentik, arbor toolsuite, tcpdump, wireshark, etc. If all were to start at the same time on getting this code in, instead of waiting each other out, we can get this quickly out in the market and into our networks for us to use.

And to remind these vendors on what to do, i suggest we do it in a festive way with a christmas card. Feel free to use this as you wish.

Step 1. Construct a christmas-card in a festive, but serious way. Print it out on adhesive paper.



Step 2. Apply some scrapbook logic on your newly printed christmas card.



Step 3: Adress the vendor(s) of your choice and make sure that they understand you mean business.


Fix Large BGP Communites!
XXX
AS1653 /AS2603
Sverige Julpost 2016
Sverige Julpost 2016
Att: JUNOS BGP team
Juniper Networks
1133 Innovation Way
Sunnyvale, California
94089 USA
Phone (46) 018-10 96 00
Photo: Jörgen Larsson

Step 4: Send it!


postnord
PostNord Kundservice
0771-33 33 10
17oo
13oo

Step 6: Profit?

Not everyone can travel the world or find the time to participate in IETF and be active on the IDR mailing list. While it would be good, its not for everyone. What everyone can do though is the above: **ask your vendor to support Large Communities**. IETF has neglected the importance of fixing BGP Communities for a long time and now we finally have **the** initiative to solve this. Since we don't need another fiasco like -*flexible*- or -*wide*-, It is very important that we see ***-large-*** all the way through!

Call your vendor today!

Skriven av

FREDRIK “HUGGE” KORSBÄCK

Network architect and chaosmonkey for AS1653 and AS2603. Fluent in BGP hugge@nordu.net