BGP,
The Good,
The Bad, and
The ~~Ugly~~ Missing

Ideas to improve BGP

Thomas Mangin
NetMCR – Oct 2017

TL;DR

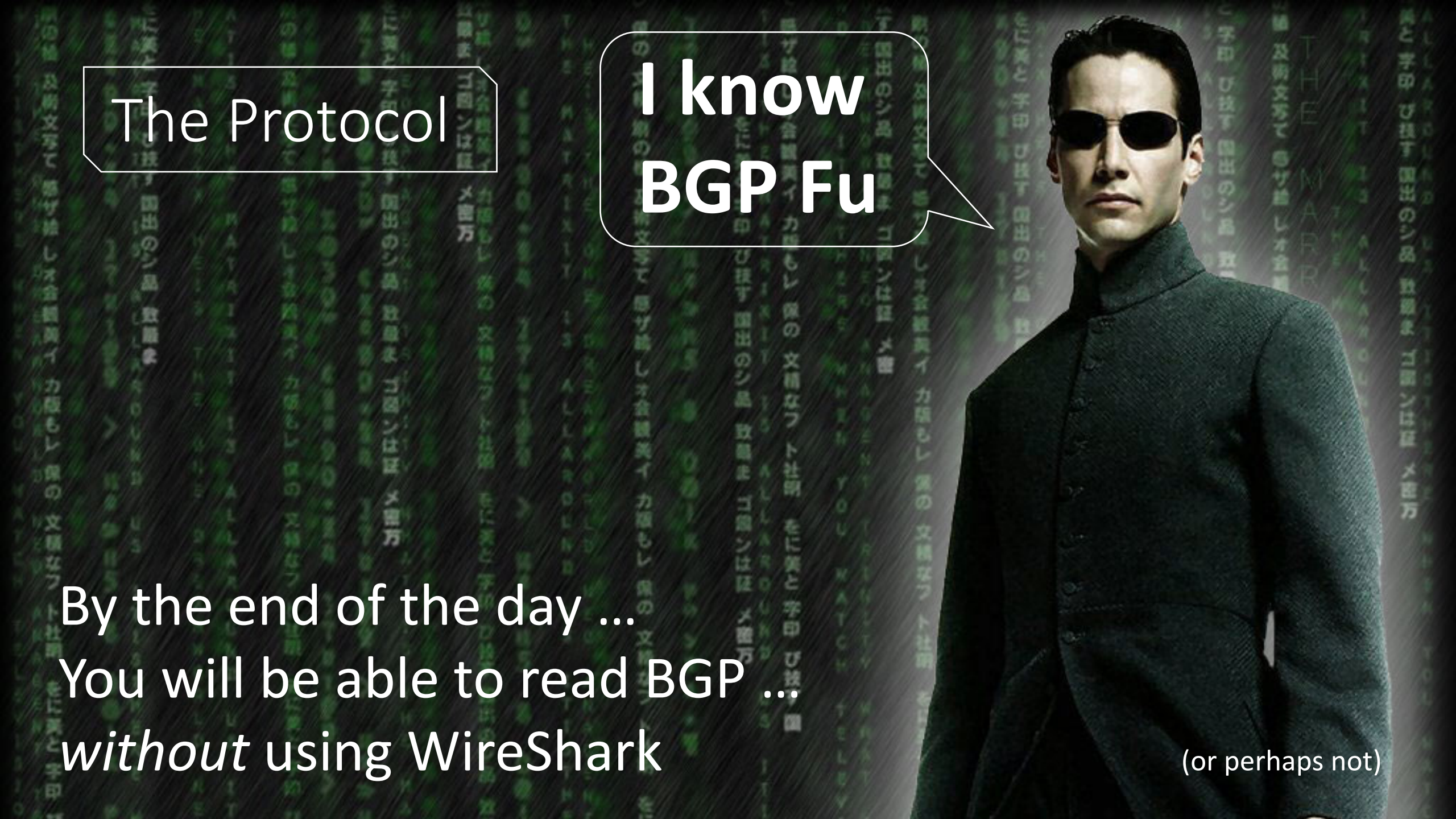
TL;DR WIKIPEDIA

Wikipedia: Condensed for your pleasure.

1. Show how BGP ***was*** compact on the wire and memory friendly
2. Point some minor weirdness / quirkiness
   Explain how successive RFC ruined BGP and/or did not improve things
3. Try to look forward at ways on how this could be fixed
4. Explain why this is very unlikely to happen at the IETF

Ultimately, argue that BGP need ”fixing” (or a new protocol is needed) by the industry in the hope someone with money, time and skills is listening somewhere and decide to help.

The Protocol



By the end of the day …
You will be able to read BGP …
without using WireShark

(or perhaps not)

“Layer 2” Connection

- TCP port 179
  - Easy to code
  - works through NAT !!

  Good
  - TCP session failure detection is very, very, LONG … RST ?
  - hence a “convoluted” protocol heartbeat mechanism

  Bad

- Tricks
  - There are quite a few “undocumented” behaviours like Alcatel using a TCP window size of zero to tell speakers that no CPU time is available and that peers should not send UPDATEs anymore.

  Bad

Framing

- Simple binary TLV ..
  - Binary, compact & OO friendly
  - Old school

  Good

- Many TLV, or LVT, or LV, or ..
  - Every draft re-invented a TLV variant
  - No chance in hell to get that fixed

  Bad

- Maximum message
  - 4k should be enough for everyone ..
  - Design for RAM contrainted systems
  - 4k is a UNIX page size (easy allocation)
  - Hardcoded in the draft, not packet

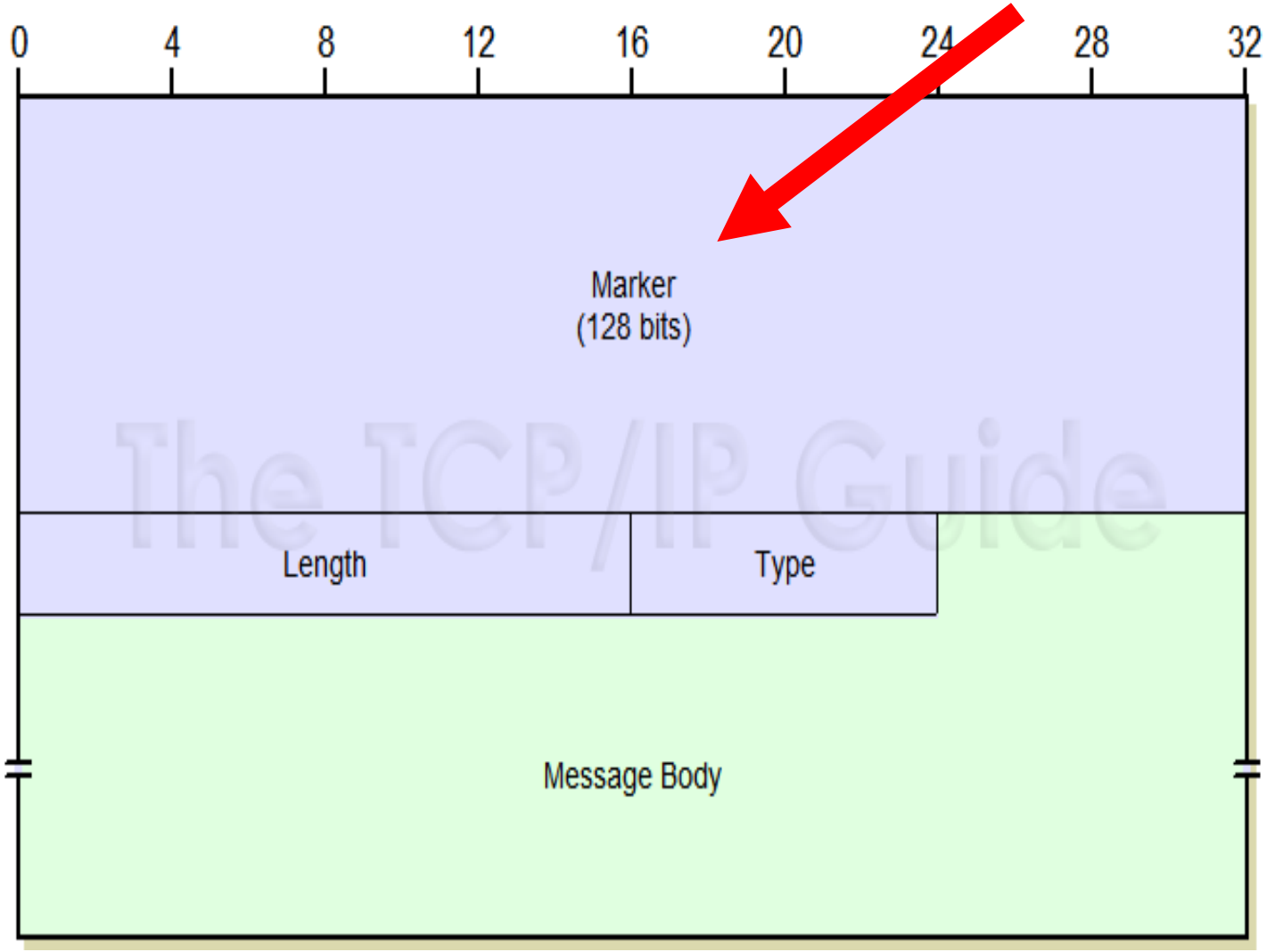
  Good then Bad now

One draft lingering for year trying to the raise the limit to 65k ..
(finally seriously considered).

Missing, but there is hope ...

Mandatory Sci-Fi reference
(A Dalek from Star Trek)

Framing

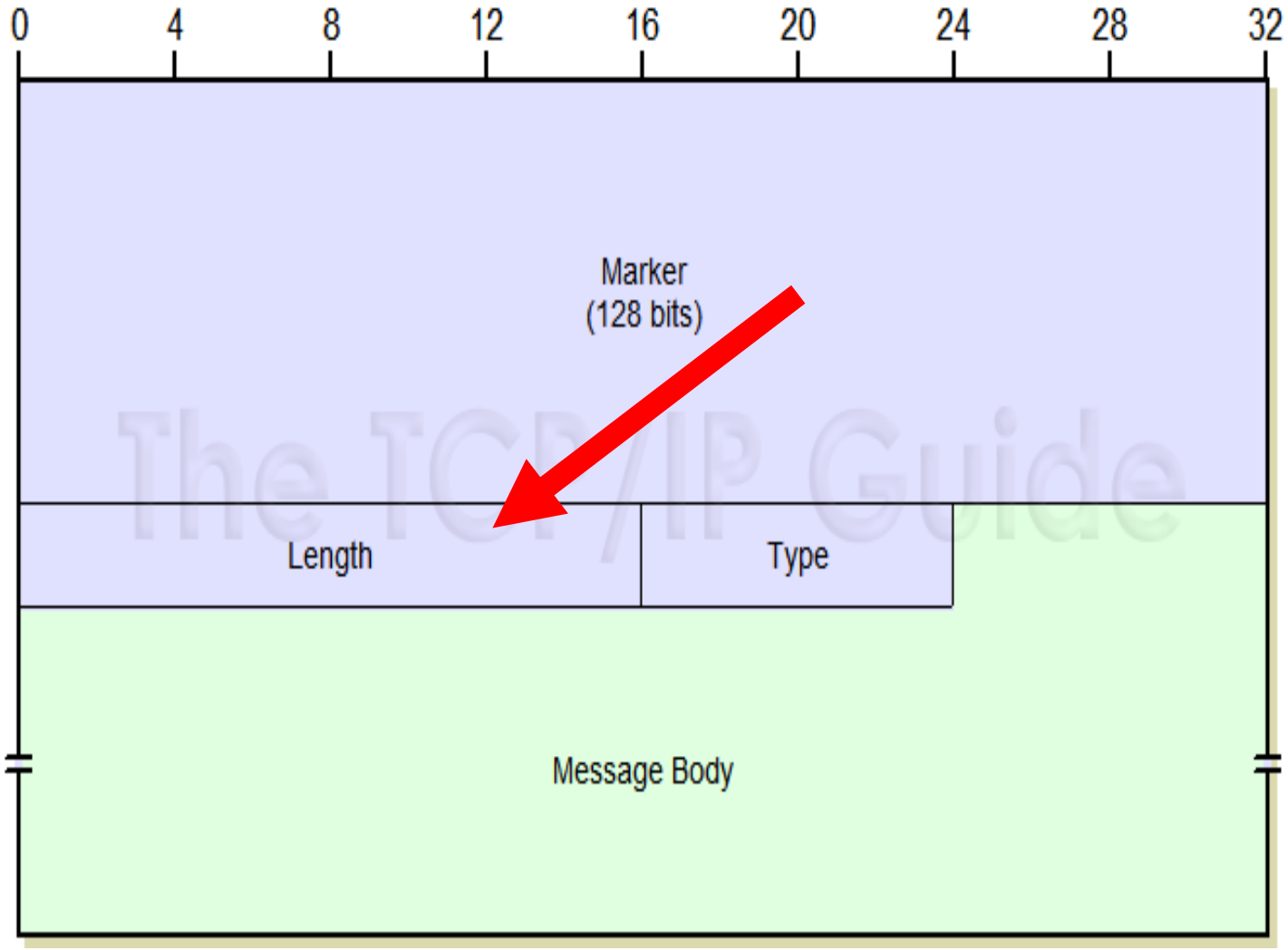


This is a BGP Header.

Introduced with BGP v3
(like v6 comes after v4)
in October 1991

To erase BGP “v1” headers,
not changed/fixed since
Bad

Framing



Length first,

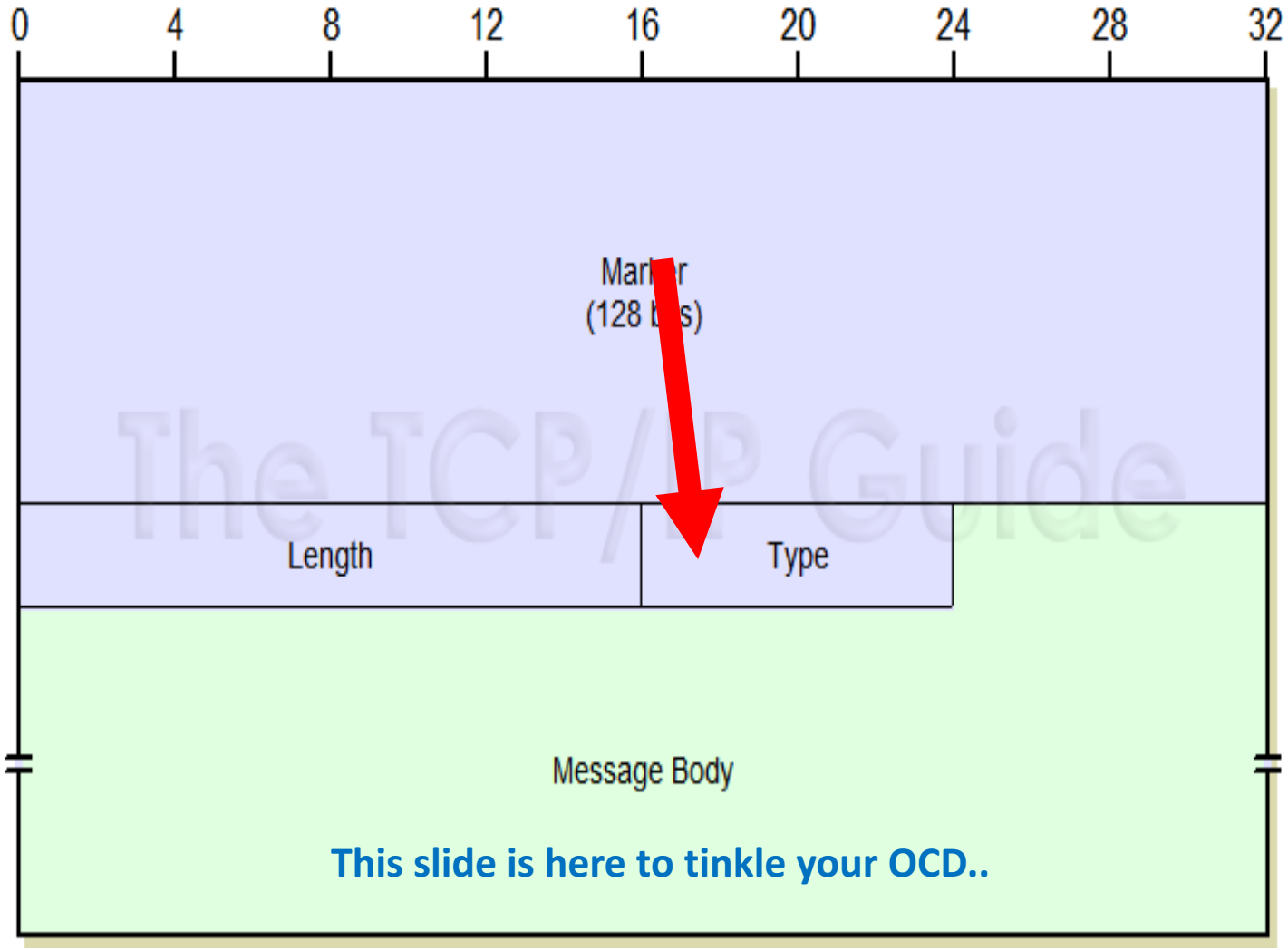
It allows to put the packet content in memory with one read

Good

No simple way to upgrade it to 32 bits by changing the MESSAGE Type

Bad

Can not see any logic in the numbering …
It does not matter unless you have very acute OCD



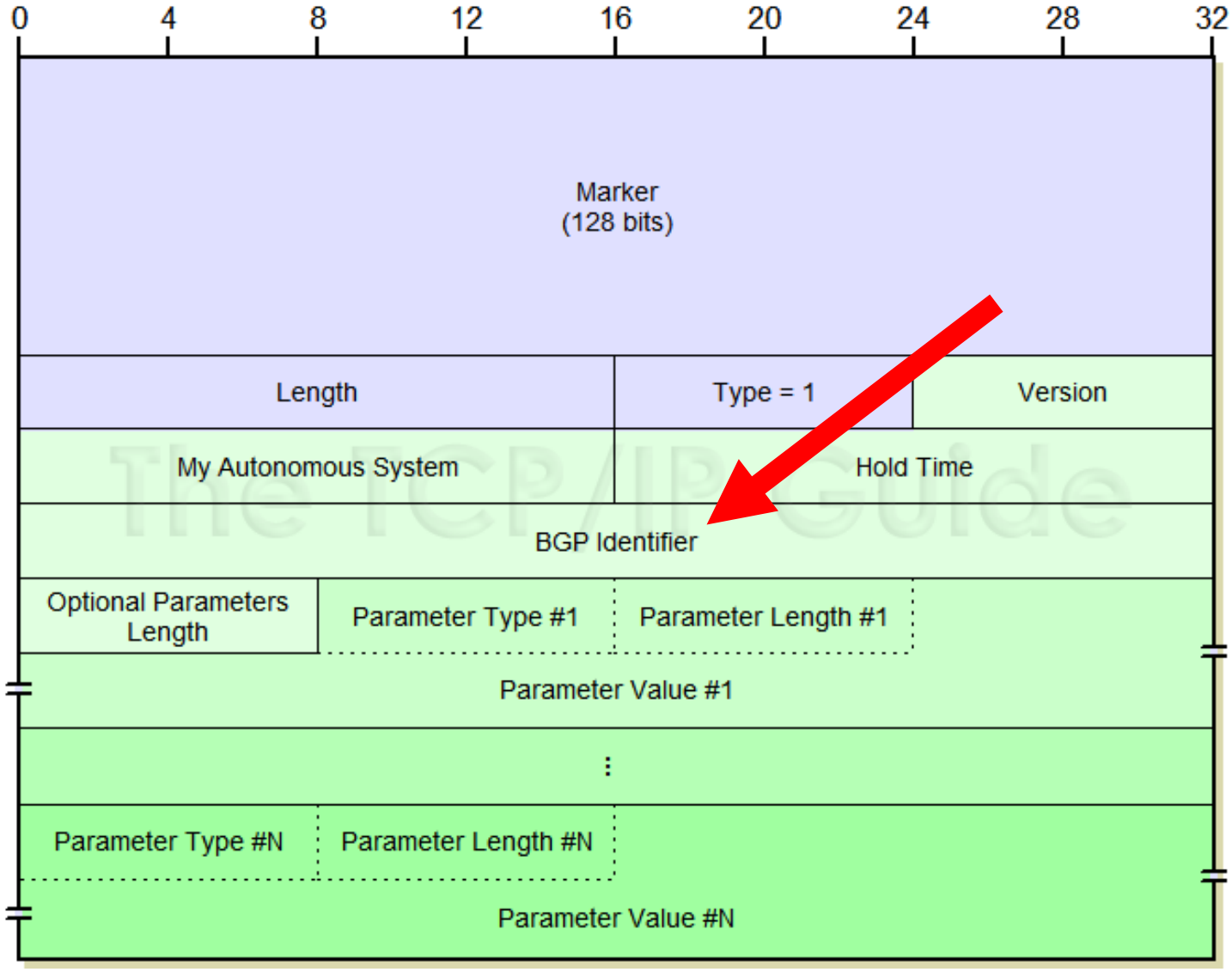
Message Type Code

1. OPEN
2. UPDATE
3. NOTIFICATION
4. KEEPALIVE

- Message Type Order

1. OPEN
2. KEEPALIVE(s)
3. UPDATE
4. NOTIFICATION

OPEN

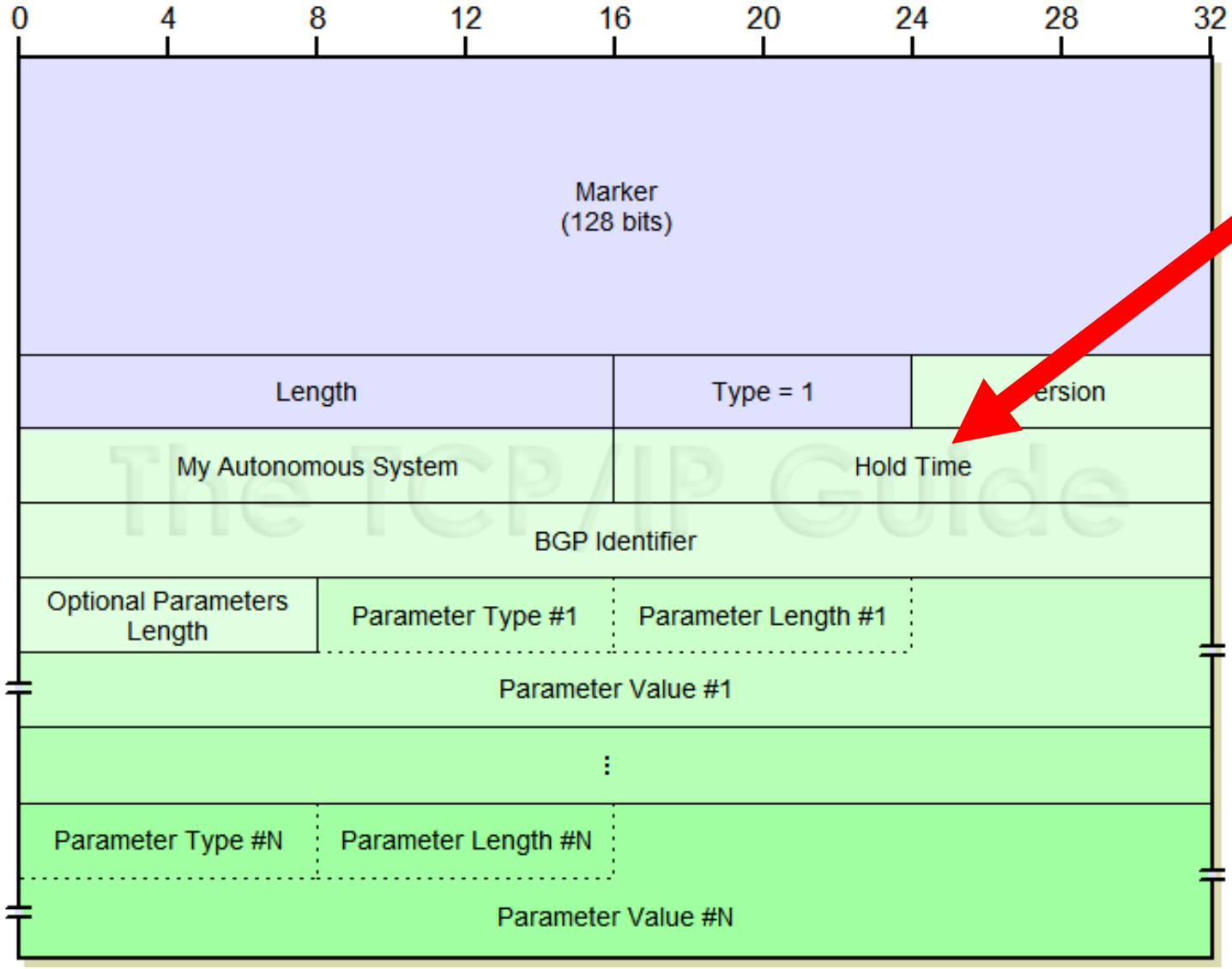


BGP Identifier aka “Router ID”
Not an IPv4: an ASN unique ID (“linked” to the OSPF Router ID)

Not IPv6 only network friendly
Hard to foresee 20 years ago
But a pain for v6 only networks

Huawei tried to change this and failed.

OPEN



Minimum HoldTime is 3
(or 0 for disabling)

"KEEPALIVE" Heartbeat messages every HoldTime/3
(should be the timer value here)

Best time for failure detection is 3 seconds. … a **bit** slow nowdays
Bad

Open –> Negotiation –>

BGP LITIGATION

- “Capabilities” negotiation
  - It is what allowed BGP to evolve
  - And have partial feature implementation

  Good
  - Size constraint slowly showing

  Bad

- Anything recent is “negotiated”
  - 32 bits ASN
  - Family (IPv6, VPN, FlowSpec, EVPN, …)
  - Add-Path

- ASN are not 16 bits anymore
  - Caused “transitive sessions drop”

  Bad
  - All fixed so “ok” ..

  Good

- Explicit version in header
  - Every implementation checks it
  - Wonder why, we have the marker

  Good

UPDATE

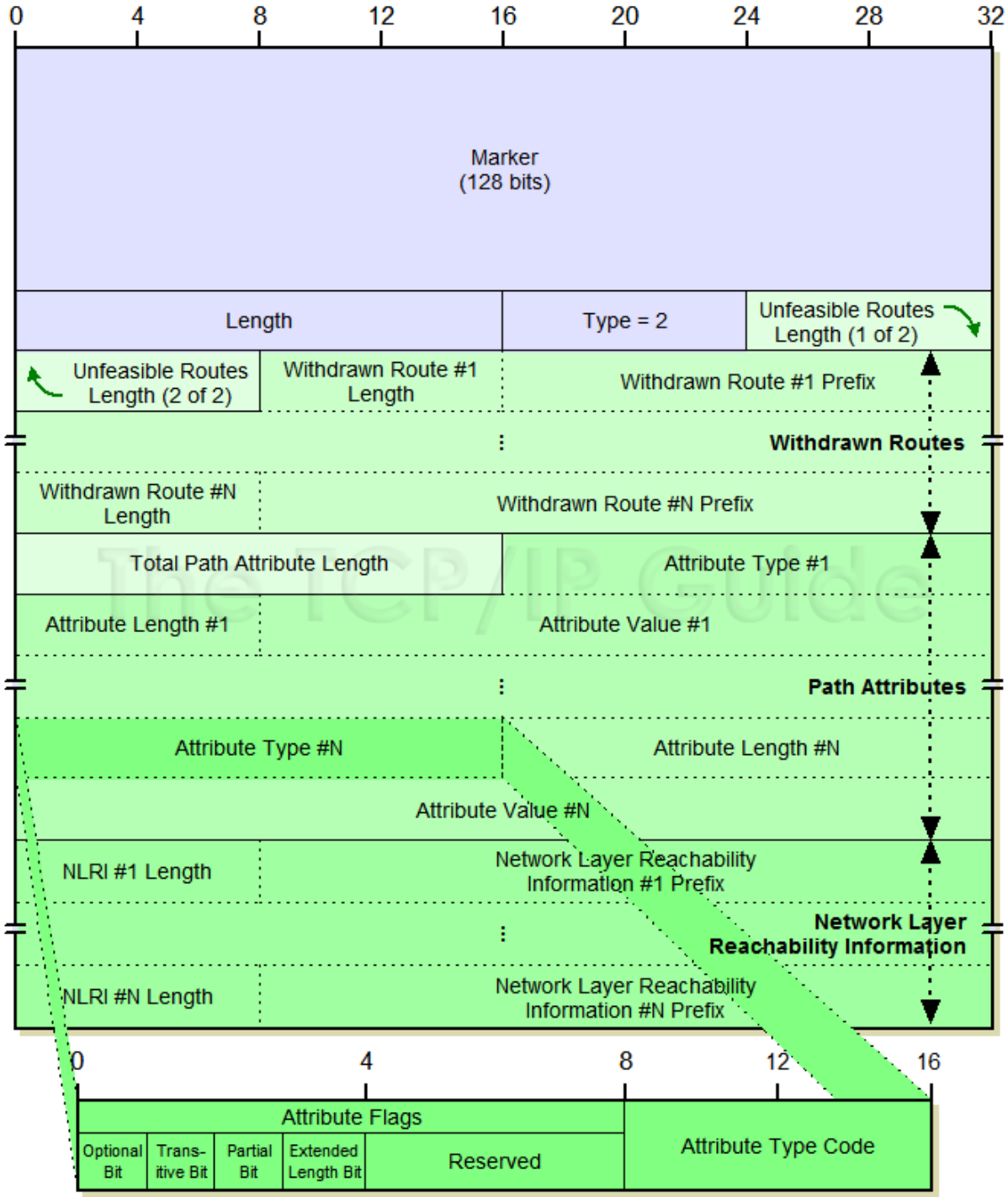
BROADGREEN PICTURES

- NLRI encoding
  - IPv4 is **VERY** space efficient

  Good

- Multiprotocol after thought (ie: IPv6)
  - A IPv6 NLRI is an attribute ! What !
  - ONE announcement & withdraw

  Bad
  - The packing is now **VERY** wasteful !



This is a BGP UPDATE

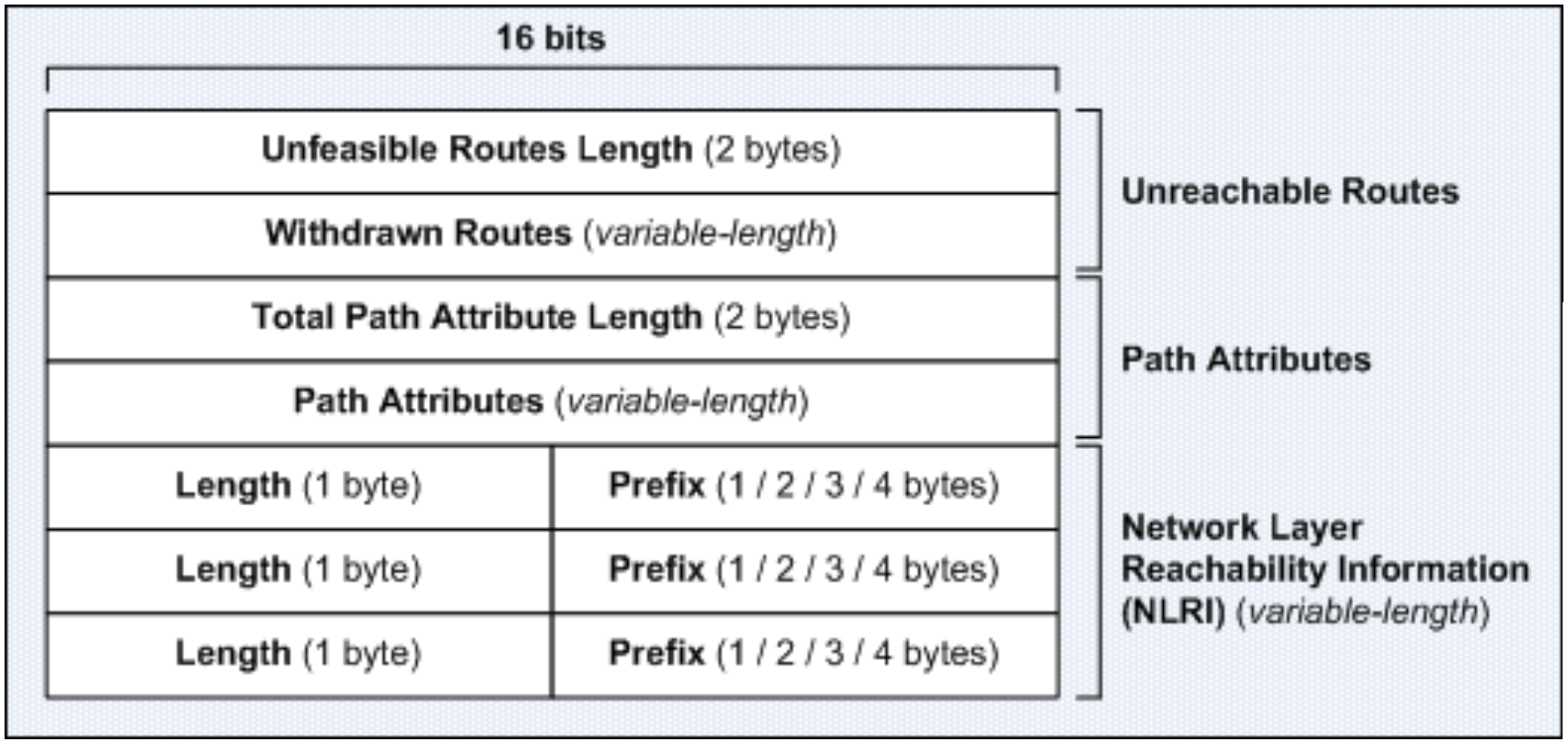
We could speak at length about UPDATE "attributes", but they are **"ok"**

Let's skip their weird encoding (7 or 15 bits)
AS **transitivity** still **scare** some people.

They are hard to explain in quick talk.
But fundamental to BGP design

UPDATE

Lovely packing, now feeling nostalgic about other “good old” binary format such as IFF, later PNG



Nice, Simple, Compact !
Just simplified a “bit” here for clarity !
(no Path Attribute)

UPDATE

BGP
Business Growth Programme

- Attributes are a kitchen sink
  - Every BGP new feature is an attribute

  Very very bad
  - Easier code to change by vendors

- UPDATE generation code is COMPLEX
  - Have to break every 4k

  Bad

- Many issues fixed in recent RFC
  - ordering, reliability, …

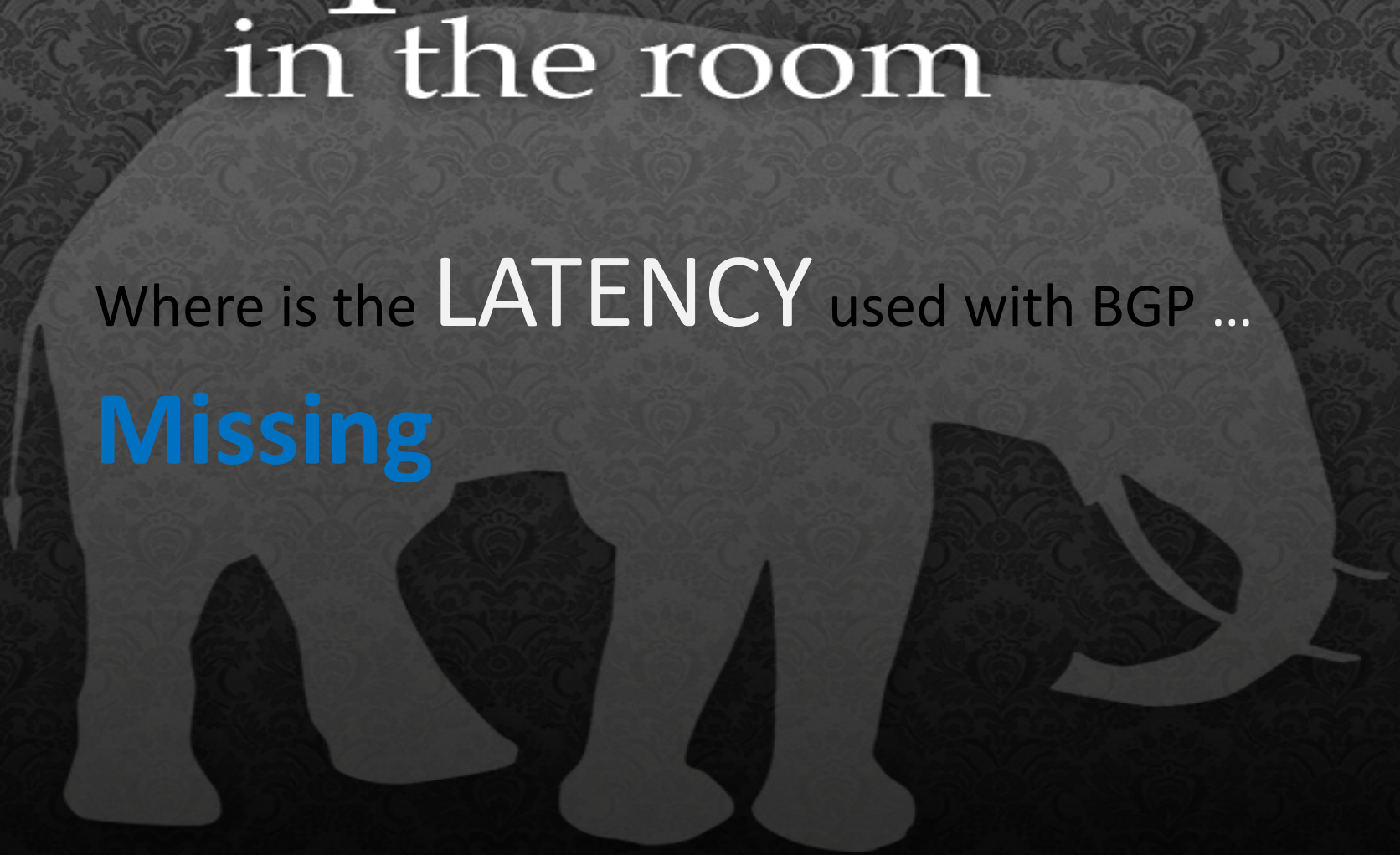
  Good

Mandatory “cute” kitten

Forward-Looking Statements

All of the statements in this presentation that are not statements of historical facts constitute forward-looking statements within the meaning of the Private Securities Litigation Reform Act of 1995. Examples of such statements include future product development and regulatory events and goals, product collaborations, our business intentions and financial estimates and results. These statements are based upon management's current plans and expectations and are subject to a number of risks and uncertainties which could cause actual results to differ materially from such statements. A discussion of the risks and uncertainties that can affect these statements is set forth in the Company's annual and quarterly reports filed from time to time with the Securities and Exchange Commission under the heading "Risk Factors." The Company disclaims any intention or obligation to revise or update any forward-looking statements, whether as a result of new information, future events, or otherwise.

The Elephant in the room



Where is the LATENCY used with BGP …

Missing

Attribute MESSAGE, ideas !

- Separation of Attributes and NRLI parsing
  - Dissociate Attributes and Updates
  - Same attributes are parsed and parsed again
  - Most of the BGP parsing is attributes

  Terribly Bad for IPv6 – **Just very Bad** for IPv4

- New MESSAGE for attributes information ?
  - CPU + bandwidth vs Memory / Caching
  - Memory not the weakest link to achieve good convergence
  - Remove the definition from the UPDATE, Create a new MESSAGE
  - Reference “Attributes” MESSAGE in UPDATE (save LOT of parsing)

Attribute MESSAGE, ideas !

- Also allow attribute composition ?
  - This is how router configurations are build on modern CLI
  - Many communities are used :
    To set high/low local-pref
    To remove RFC 1918,
    To drop traffic,
    To slice bread, …

- Around 95% of routes in the DMZ have unique AS_PATH
  - Having the AS_PATH part of the grouping is sub-optimal
  - It ***may*** make sense to move AS_PATH with NLRI
  - No real personal research on attribute grouping

UPDATE MESSAGE, ideas

- A “route” is really a NLRI & a next-hop
  - Attributes are for route selection
  - Grouping next-hop with other attribute data is sub-efficient

  Bad

  - It does make sense to group by next-hop
  - But next-hop not really an “attribute”

  Split next-hop from the other attributes and group NLRI per next-hops

- None of the ideas presented change the route selection process

UPDATE(2) MESSAGE, more ideas

- Why not create a new MESSAGE type for Multiprotocol
  - Keeping the same format for attributes (improved or not)
  - Just different NLRI encoding (not considering AS_PATH)
    - AFI/SAFI
    - MP withdraw
    - Attributes (current format with proposed idea)
    - Next-hop + set of MP announce,
    - Next-hop + set of MP announce, …

(Or have an attribute and/or capability to signal a change of NLRI parsing)

Disclaimer: The chance of seeing these ideas happen is (near) zero

But please feel free to show me wrong !

Finally, an agreement was reached on a standard change

This is/was an opinionated talk … I am right and everyone else is wrong

BGP, means IETF

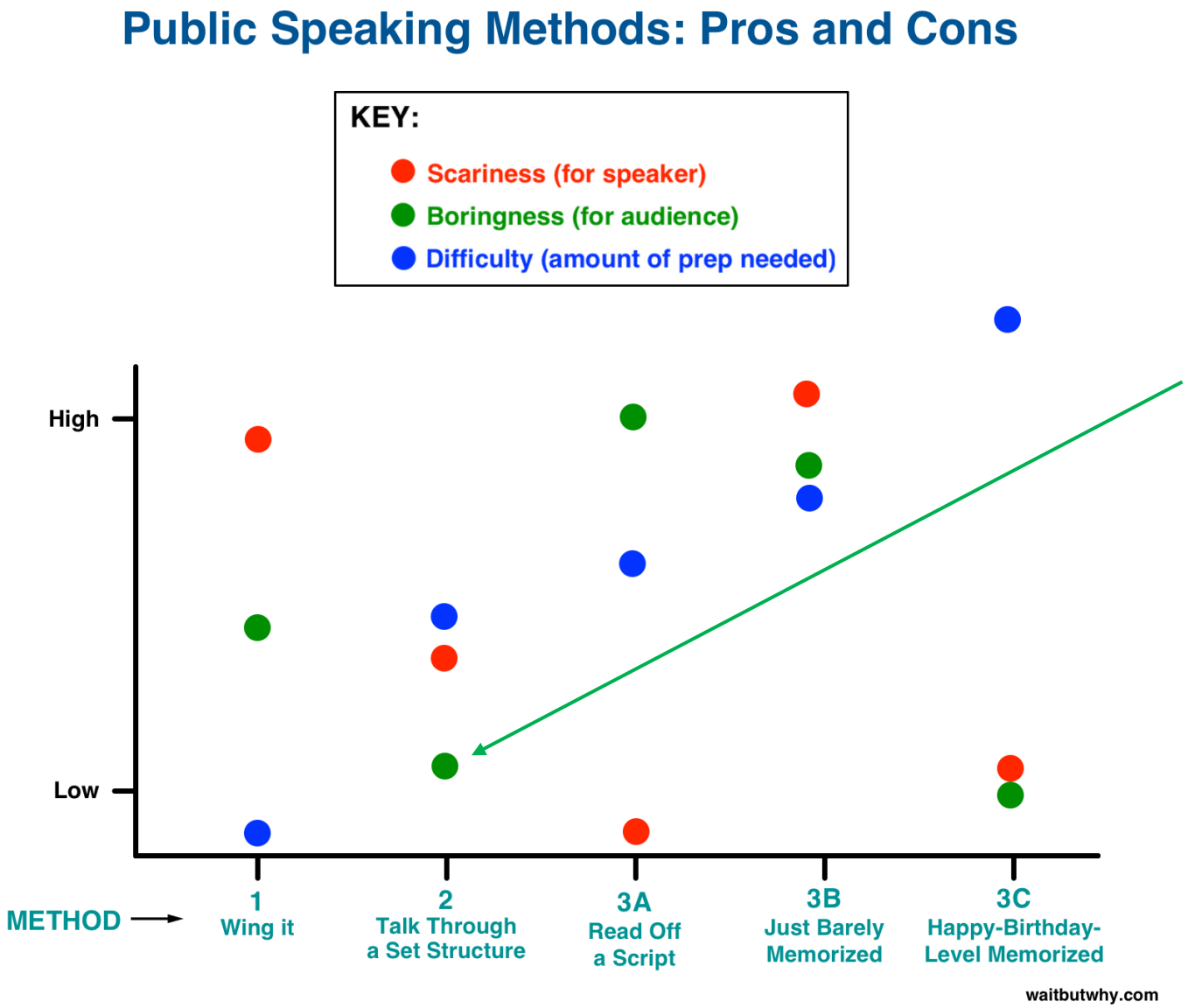
- Vendors are very influent
  - They pay people who code the thing
  - They listen to $$$ clients
  - BGP is made by 10s and 10s of RFCs
  - Useful drafts in limbo for years
  - Lots of politics (like everywhere)
  - by “spec writers” not “programmers”
    can lead to some weird stuff

- Very few operators
  - Mostly only large networks
  - Not enough operator feedback
  - Not enough operational feedback

  Bad

No one interested in fixing BGP,
Like HTTP/Bis fixed HTTP

Dev or Ops
IDR or Grow



https://waitbutwhy.com/2016/03/doing-a-ted-talk-the-full-story.html

Extra slides ??

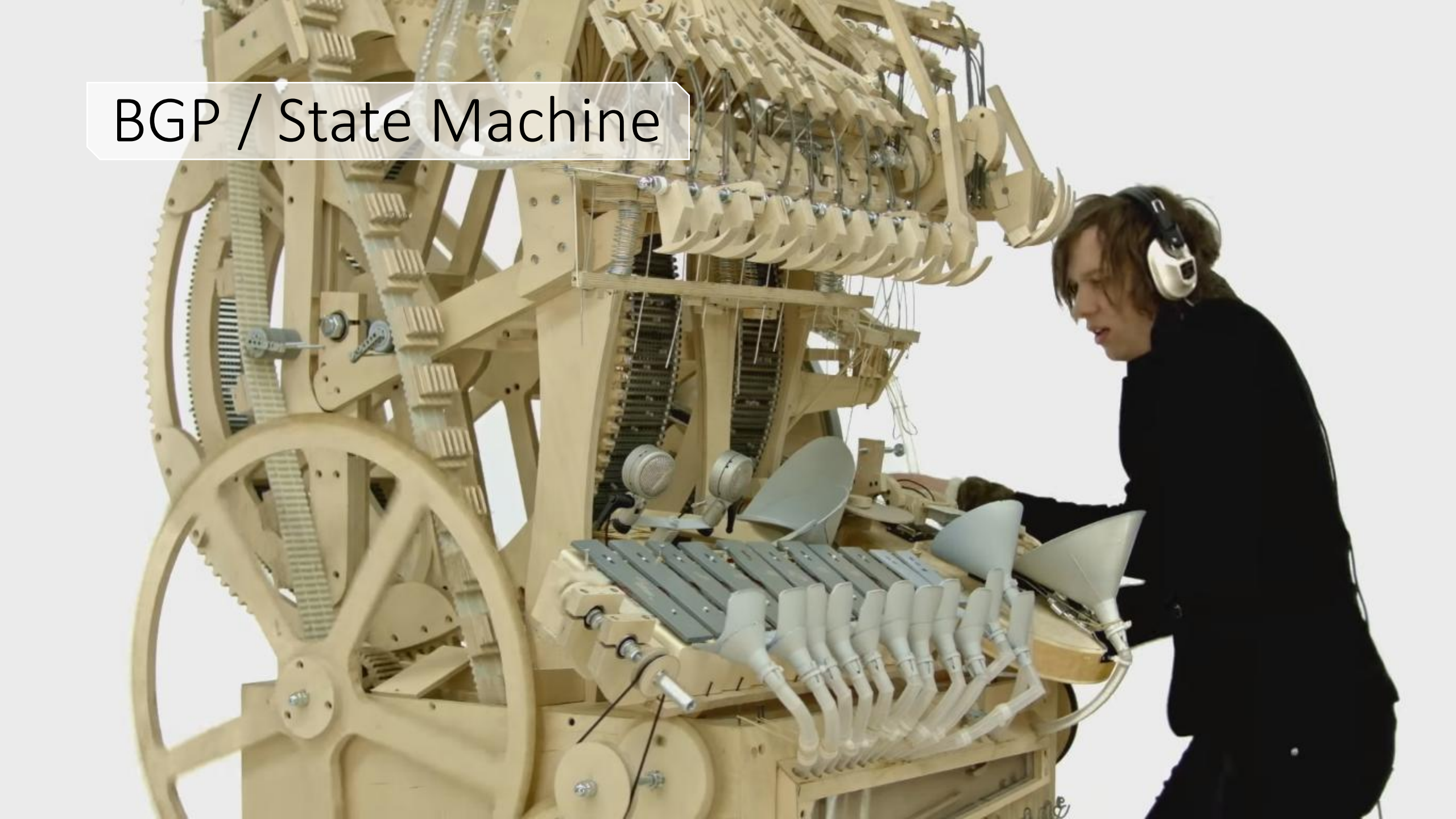
You are here .. YES YOU ARE.

And I am looking forward to seat down ..
But I **may** have spoken too fast

25 slides for 15 minutes should be around good

Emergency extra slides ?
Want more ?
Questions ?

BGP / State Machine

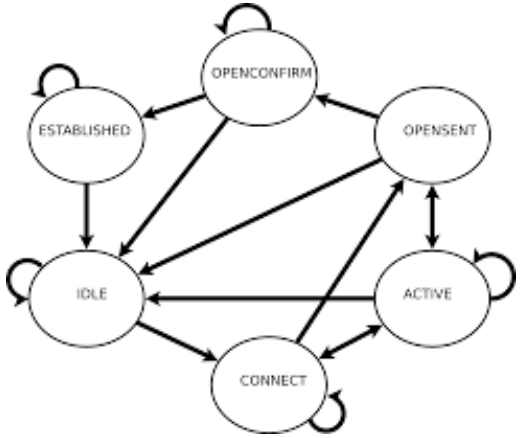


State Machine

BGP Holdings

- Should makes things clear in RFC 4271 .. Should ..
- Very hard to “get” (putting code ideas in words is hard)
- Most diagrams of it are wrong, in a way or another
- No other RFC does really update the state machine (when they sometimes should)
- Most implementations do not implement it fully/correctly
- Try to “suggest” an implementation(s) of the BGP reactor (try/except can achieve the same without it)

Good / **Bad** … Pick one !



BGP Other

- KEEPALIVE
  - 3 need to go missing to consider the peer dead
- NOTIFICATION
  - Notification of issues / session going away
  - Job worked on this :-)
- Empty UPDATE
  - Known as EOR (ie: you can now sync the RIB to the FIB)
  - MultiProtocol IPv4 vs IPv4 “native” – interop issues in the past (resolved)
- 2x KeepAlive
  - Same but it is a trick .. Not documented anywhere

Be careful, Googling BGP can be surprising …