



---

# AI, NETWORKING PROTOCOLS AND VIBE CODING IN THE 21ST CENTURY

---

Some Recent Observations and Thoughts



Jeffrey Sicuranza



MARCH 1, 2026  
APPLIED METHODOLOGIES, INC.

Contents

1 INTRODUCTION..... 2

2 AI AND NETWORKING PROTOCOLS ..... 2

3 THE NETWORKING PROTOCOL TECHNOLOGY SANDWICH ..... 4

4 AI AND VIBE CODING ..... 10

5 IN SUMMARY ..... 13

## 1 INTRODUCTION

---

I have been reading some interesting posts that come through my LinkedIn notification feed regarding the topics of AI, vibe coding, network automation and creating AI agents so one can converse with their network devices for various uses. I just wanted to share some thoughts with the community about this recent activity in our industry with a historical backdrop and what has been in the back of my mind regarding the state of our networking protocols to date. I applied some imagination and questions about the current state of the union in networking and its future with what we see coming in terms of AI progress and adoption. My goal here is to provide some questions, imagine some possible outcomes and have you ponder as well. I hope you find this read interesting, imaginative, enjoyable with some humor, and thought provoking.

## 2 AI AND NETWORKING PROTOCOLS

---

I believe we are still very early in a Scientific Method and business philosophical stage of AI and computer networking. When I mention AI in this document, I am referring to all its subsets, Machine Learning, Decision Tree, generative, purpose built et al. In addition, when I mention L2/L3 and up I am referring to all the current networking protocols we use today in the TCP/UDP/IP stack and their upper layer utility and application protocols, DNS, SIP, HTTP et al. in the context of Data Center AI networking but this can still apply somewhat to the enterprise and WAN as well. Of course the security aspect including risk, security solutions, network and crypto protocols, governance, compliance is presumed, thus no need to cover in detail.

At the time of this writing in our industry, the initial Vibe coding networking projects posted on LinkedIn by some talented members are fascinating and very cool to experiment with and are great way to learn some AI tools while reinforcing your network protocol Finite State Machine(FSM) knowledge. They appear to be fun to tinker with and the open community folks are experimenting with this nascent technology but be wary, you need that assurance component.

We are still in this initial democratization of data phase, and I had seen this current trend to converse with your data coming a couple of years back. I gave a presentation to a client on AI in IT to consider its use for improved NOC visibility and one message in my presentation was **“we will be conversing with our devices not sending queries to them”**. You can find that presentation at <https://tinyurl.com/3vty6n2e>

I believe we are inching towards “some” goal of democratizing networking via AI, but still a ways off. Also, I believe the network original equipment manufacturers(OEM)s will be including “enterprise” grade protocol FSM specific Small Language Models(SLMs), AI chips and agents with potential Retrieval-Augmented Generated(RAG)to retrieve data from their other products in their product ecosystem soon for that “complete picture of awareness” across their products when in use.

Of course, if you are going to greenfield this landscape change in the enterprise you may want to do it with a vendor's solution for recourse. Open Source POCs are a great first step and are fun, but risky. Plus, we may end up creating additional layers or middleware bloat that compound the abstraction to a darker, black box, and possible shadow AI use, or worse *Artificial Insanity*, so learn from the past(no machine learning pun intended).

However, this is where I was impelled to draft and post some of these thoughts and questions. Due to our current economics of business and technology ROI convictions we are still running everything on aging protocols over 30-40+ years old.

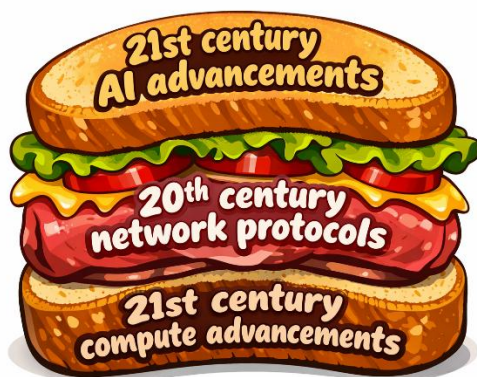
### 3 THE NETWORKING PROTOCOL TECHNOLOGY SANDWICH

---

Our current networking protocol FSMs are solid and mature in operation so yes please for the love of the RFC's don't reinvent those wheels by trying to emulate its FSM using AI, its abstraction tools or bolt on to the protocol's code thread in production with AI integration. The management plane(MP) is a potential fit for AI integration with a natural language interface(NLI) that I will discuss later. However, we must be very careful with its use for the Control Plane(CP) and Data Plane(DP) for obvious reasons I don't need to go into here. Who needs misleading forwarding/routing bias or hallucinations.

As we pass the first quarter of the 21<sup>st</sup> century we basically have a 21<sup>st</sup> century technology sandwich problem that we may be inadvertently trying to mask with some upcoming tools or approaches under the recent AI hype and transition era. The old axiom of "building a solution to a nonexistence problem" or well... just making a mess may start to resonate, especially since we don't have xAI(Explainable AI) yet.

To illustrate the proverbial technology layer sandwich the bottom bread in the figure below represents 21<sup>st</sup> century compute advancements, the middle meat represents (may end up being baloney, no pun) our aging L2/3 and up networking protocols, and the top layer bread represents 21<sup>st</sup> century AI advancements.



We can make this sandwich and eat it begrudgingly depending on how the meat tastes (industry status quo) or make a new sandwich with a new middle layer(major industry shift). Economics and ROI will dictate this debate over the technology's progression for a while.

The networking industry may head towards or attempt the “lets marry AI and the protocol FSM CP/DP” route so there is an on device AI model that has learned all the rules of a protocol’s FSM but can inference and learn to improve its operation in real time while adhering to its RFC rules for potential improved protocol use and easier expansion. Possibly marketed as “AI routing” or “AI networking” where the administrator now converses via a NLI with the network on a per device or distributed manner depending on size. However, to achieve that scenario too would be a major industry wide, and operational endeavor let alone cultural shift in how we work on such systems compared to today. And yes, that while my scenario is high in the clouds for potential, the risks and ROI that it poses are well.... Cloudy.

Take networking protocols that have built in expandability via Type/Length/Value fields in their header structure for example. For your private network can you imagine a time where there may be the ability for RFC user allowed spaces for private use in the TLVs for a personal private extension use need?

You converse with your networking device or dashboard AI agent and ask it to add a TLV to ISIS so I can do x with it. Or add a SAFI to MP BGP so I can use it for y, then provide the rest of the deployment steps all because the protocol’s FSM is in an open to be trained model now. For example will AI influence Segment Routing operation using inference and a form of generative adversarial network (GAN) on the device?

Yes, we do this manually today via a device’s command line interface’s(CLI) language with some of the protocol’s optional bits to flip or repurpose certain headers fields. For example, you can use the IPv6 flow label bits if you wish for a specific tagging and filtering route or security policy purpose. And, yes we may ending up conversing with AI using our current CLI language(discussed later) as our dialect so you can imagine what may come.

---

*Now the big question, with what was imagined above, though fanciful, if trending towards reality would that reality be better served with existing network protocols or a newer generation of protocols that fit in with AI’s ongoing development for even improved efficiency, scale, ROI and security for local and distributed computing?*

---

AI can be a great tool for the management plane with observability only(today) and eventually automation with a phased adoption for future products. For example in the context of our technology sandwich metaphor:

**Phase 1).** AI to be applied to current products (sandwich) or newer future products.

Based on what you want to accomplish (per device or entire network) this is where that intent networking comes into play some vendors have been pushing for about a decade. You know, that whole implied vs explicit network build deploy and manage approach. You converse with a central or per device basis via your NLI for the change you want to make and AI executes locally, regionally or globally, with the assurance roll back component of course. Now, for the Small Office/Business and Home Office scale, this ability to converse with your network for simple changes may be a boon with the less technical savvy operator that buys a retail product with the accompany security included of course.

For the enterprise and industrial sectors, it has its uses too but with a tremendous amount of risk. Natural language your network it seems. There is an amount(TBD size) of compliance and restrictive regulation on the enterprise/industrial that would need to be identified and then baked into networking products to prevent the proverbial “misconstrued NLI conversation” or “Artificial Insanity”. Maybe by the time this happens I should be in, the old meaning of, Ether by then.

**Phase 2).** This phase may be risky to try to “sandwich” in with current and slightly older networking hardware and operating systems(OS) versus future AI engineered network hardware, operating system(OS) and protocols.

Why? Well this phase would be what we have been trying to get to for decades that unicorn holy grail(or hell) of the “*self-healing and maybe self-managing network*”. So, using forthcoming networking infrastructure products with AI capabilities and automation baked in each networking device with its own AI chips, SLMs(Small Language Models) all trained on our current set of protocols operation and practices for a solid secure recoverable connectivity no matter how it is connected. It then figures out the topology and configures the devices accordingly. Almost to the point where you can take your detailed network diagram feed it to AI and it does the rest on devices staged or out in the field already base configured and connected together. Or, you discuss your topology with it and it happens.....A bit out there yes.

Each device’s local AI knows the FSM rules and learns the entire network when it joins, local flows, route tables, potential impacts, local hardware issues, temp. power, memory integrity et al. all the way to the point it knows the whole network not just in flows but in detecting potential failures, bottlenecks, all forms of bursts and security. Each device is basically its own network engineer’s learning brain with enough autonomous capability. It may include RAG to check for code updates at the vendor or user repository on its own and executes automatically after notifications or instructions you “discussed” with it. The bandwidth of such a network would probably double, one set of natural production traffic and the other is AI learning and conversing about the same natural traffic with you.

What used to be networking protocols exchanging per hop behavior information about state between networking devices from basically a distributed 20<sup>th</sup> century decision tree approach, as long as all involved play by the RFC rules and other sound practices, may eventually become an AI network with ML protocol models instead that will still take the connectivity operation information/knowledge and use it locally but also communicate between their brethren to address networking matters directly long before it gets to a NOC screen or log journal for preventative adjustment. And this may be accomplished with a central or distributed device AI chips/agent approach now that we have forthcoming 100g plus speed and eventually Terabit Ethernet.

Granted a bit of a large or quite large pie in the sky scenario to say the least but it is possible. And if this were even close to some AI based networking nirvana our OEMs may have nothing left to sell. Right?

Okay, back to the debate aspect with the sandwich metaphor, do we get to a point to attempt a Phase 2 type of world using the protocols we have today but with newer purpose built AI networking devices. Or do we stay at Phase 1 longer while the industry adjusts, wedges in, and squeezes the existing protocol's use until a new set of protocols that has Phase two's ability baked in for AI and the slow adoption/transitions begins to occur over a decade or two?

I believe it is an interesting scenario we may face in the industry and it reminds me of a different time(seems so ancient now) when the networking industry for a decade plus was in its multiple protocol era and companies, regardless of size, did migrations and forklift changes from their IPX, LanManager, NetBios, AppleTalk, Vines, DEC, SNA, worlds with various protocol gateways until we all landed with Ethernet and TCP/IP. A decade later we thrashed about at the L2 again with our fabric wars, tried SDN's Open Flow and today with EVPN and Segment Routing.

There was a period when compute, networking and applications appeared to advance together but then we kind of slowed with our L2/L3 and up networking protocols. From a business standpoint the market just settled with Ethernet and TCP/IP. Ethernet in this context will include WIFI though not as old but from a frame construct aging standpoint.

Even though Ethernet continued to have their relevant PHY/PMD and media advancements occur over the same period as compute has in the 21<sup>st</sup> century with its hardware, virtualization and containerization it is still as old as our TCP/IP friend.

So, do we also consider changing to something beyond Ethernet as well for AI especially in the AI Factory space, including all the clusters, pods and nodal networks that are being built or planning to be built today with aging Ethernet, but tweaked with efforts as Nvidia's NVlink, Ultra Ethernet working group's specifications, Broadcom's Scale-Up Ethernet framework?

We had our protocol wars in the 80s early 90s, that featured some interesting and compelling protocols of their time yet economics steered us towards the Ethernet and IP model based on our compute abilities and uses. Ethernet has advanced at the PHY/PMD with some L2 Ether-type additions and shims included. However, those L2 shims are aging as well as our DNS, ARP, Multicast and the best we can do are some enhancements like QUIC and TLV options in some protocols.

I mean consider our DNS structure today and IPv6 is also 28 years old for BGP's sake!

Granted the originating 19<sup>th</sup> century copper POTS system lasted over a century while it supported 20<sup>th</sup> century computer/networking advancements so why not our aging current L2/L3 and up protocols?

So now we have these AI "enhancements" as re-introduction of various 802s for congestion control, priority signals et al. to achieve lossless. You see DCQCM, PFC, WRED, re-using EC bits in IP for signaling ECN in TCP and all these older L2 802.3x, 802.1qbb and .1p/q, RoCE, to provide some form of lossless capability for intra and inter xPU node communication. Yes, there may be some xPU to xPU proprietary but we are still just using the same aging constructs.

The fact that AI is demanding true lossless for inference and high level learning and the best we can provide are still queuing, weighting congestion control approaches may never mean true lossless capability. You look at this and it's as if the industry is trying to apply VoIP QoS to AI loads.

Ethernet and BUM may or not for now lend itself to inter/intra nodal communication for machine learning(ML) or inference flows though it appears we are trying other options with SUP and Ultra Ethernet but it is still Ethernet even at TeraBit.

Consider this, AI clusters today in 2026 should not have to rely on any form of priority control, RED, congestion avoidance/control buffering et al. to try to provide a lossless environment for an inference workload. Yet decades ago we had the concept of a Virtual Circuit and circuit switching to guarantee a workload's bandwidth. The big question outside of the sandwich metaphor is how much further can we squeeze the round peg of computing progress into the square opening of our current networking protocols?

We tried with ATM or other virtual circuit solutions in the past but AI workloads especially inference types will need that guaranteed true non-blocking true-lossless paths between nodes. Trying to achieve this with tweaks to aging Ethernet and L3 congestion mechanics may buy us some time until a wall is hit.

Will internal nodal InterServ RSVP type be offered as well as a retrofit?

Or maybe intra/inter nodal AI flows will be serviced via a recast Terabit Ethernet at the ASIC level using locally significant virtual simple PtP Ethernet fixed circuits between endpoints in silicon. That single locally significant Terabit ethernet circuit is called up between two xPUs and there is no blocking or switching involved. A block of hardwired PtP mesh links instead of a switch on a chip that has inherit arbitration. They can do it since we currently build these large parallel processing GPUs. Maybe new UADP types, SiliconOne or Tomahawk sets. Similar concept. Instead of cells it is Ethernet. Again pie in the sky but interesting if vendors are looking at this as well.

But even if my magic chip scenario is somewhat plausible, again do we hit our Jumbo and MTU walls? Do we just go proprietary inside and translate upwards?

Some additional questions to ponder:

Will we achieve or reach a non-Fragment and non MTU constraining networking future by the halfway point of the 21<sup>st</sup> century?

Is what we have today at the L2/L3 enough to get us by for a while or will this protocol meat in the sandwich stale out while the breads continue to improve?

Could we see something like this in our lifetime again going from TCP/IP and the whole stack, possibly the MAC too, and that networking technology will lend itself to AI(and Quantum) far better than what we have today? We have Moore's Law for compute mind you.

I mean we appear to have our computing advancing with Quantum, and parallel processing, xPUs, models, memory, storage(even glass) and our network medium as well with moving photons closer to our ASICs internally, Fiber, Coherent and Hollow Core optics advances in play yet we seem a bit stagnant with the networking protocol aspect.

I guess the analogy I am attempting to make here with the sandwich metaphor is we have top shelf bread today but we may end up making our technology sandwich using aging, warm Oscar Meyer, still wrapped its unopened plastic, meat in the middle that is getting, uh for a lack of a better term, "gooey".

Let's daydream a bit, it's mid or late 21<sup>st</sup> century and we may have Quantum Entanglement Distribution Network links in space and terrestrial yet still using TCP with its Van Jacobson, Nagel, Reno, delayed ACKs or QUICs, IP with fragment and offset ID bits running over it still. Great ROI, but also something to ponder.

Now don't get me wrong here, this is just an observation to make us question and keep our eyes wide open. I am not bashing TCP/IP and our current networking stack. The ROI has been tremendous for its age in the technology field.

I enjoy all protocols, and learning new ones while I relic in the nostalgia of all my old friends like Frame-Relay, SS7, Token-Ring/SRB, IPX, AppleTalk, NetBios, LLC and many others. I also enjoy working with, tinkering and expanding my knowledge of our current protocols in the TCP/IP stack and all those new RFCs and 802s.

So, to question further will we either advance only with the sandwiches' bread to the point we don't have to worry about the middle meat, for the compute and AI software layer bread will compensate for any aging network protocol limitations, and we don't care due to the ROI, or we will have no choice to change the sandwich's middle meat? I wonder if the networking industry OEMs are pondering the same to change or have the meat catch up. Do our OEMs have Skunk Works type projects active? Maybe, OpenAI launched stateful AI on AWS, signaling a control plane power shift in capability.

## 4 AI AND VIBE CODING

---

As for the vibe coding aspect of my observations, I just have a few additional thoughts. I know I know I can be a bit verbatim and redundant.

For the network engineers reading this(I hope). Think about it for a moment. The networking device's CLI(command line interface) has been the network engineer's programming language for decades. Those commands are our language's variables, verbs and literals. Future engineers may never even see it again by just conversing with their devices at a high level. But will they converse with such devices using the language of our current protocols or a newer 21<sup>st</sup> century communication constructs?

Vendors have been trying to achieve this pre NLI management environment via "click ops" solution using protocols (SNMP, ASN and MIBs for example) and "programmed decision tree based" expert assistant systems for decades with mixed results.

The previous "trend", prior today's AI Agents, was "click ops" via off the shelf OEM products or open source Python libraries, and our infrastructure as code mantra with various data set structures and REST APIs including our Ansible, Chef and NSO tools. This activity also included the Splunk's and pre 2022 open AI period with classical analytics business intelligence type tools used as the next "Network Intelligence" approach for operational prediction.

But now we have our little "vibing" AI agents that may still use everything I mentioned above using the same CLI calls and protocols that all the previous generation's tools used.

We may be in another middleware app/agent craze with Model Context Protocols(MCP), programming constructs like P4, intent theory and all this automation in networking which the vendors have been trying to achieve a holy grail since the 80s but missed the mark with SNMP's MIB and ASNs. And guess what you had your fully automated SDN types with SNA/APPN "back in the day" but hey old is new or new is old who knows in a Tik Tok world.

One scenario that may occur in our networking industry future is that the network device's OS's CLI command structure may become a hidden API used only tool, maybe just a "rommon" extreme break glass use, or not be used any longer to the point where it evolves into a somewhat headless networking product from the vendors with no CLI functionality.

This potentially future OS will have all its operating networking daemon code base still intact including its variables, VLAN definition space, addressing, protocol literals, TCAMs, FIB, RIBs counters all as data structures as they have always been, but with no CLI to adjust them. It will have only a GUI at boot and the rest is controlled/managed via AI conversation instructions. Protocol structures/data are still present via, buried into the code, protocol FSMs, yet the CLI is no longer needed or something like a P4 is pumped in there as the bridge to the SLM. The device starts, loads the protocol and AI daemons and you are presented with a NLI GUI and converse via locally or centrally what you want it to achieve, maybe feed it a playbook and it discusses your goals with you. Interesting if the Cisco, SONiC, Arista, HP to name a few, network operating system developers are looking at this as well.

But again a bit pie in the sky, however, my point is for this scenario of a “modern network OS” to be deployed in the future it may be better served with newer protocols which I discussed earlier, possibly in conjunction with that modern network operating system. All which are designed to embrace the newer protocol’s AI attributes, tokenization natively and thus a more efficient and leaner machine learning solution is imbedded per device with far less overhead than what we are doing today. However, that could be a decade or more away. Again, Skunk Works?

Now for the Vibe coders or “Vibers” out there please don’t get me wrong as well. Low/No Code and all these new tools are fabulous to use to help us save time in providing a new tool to solve a problem or help us do whatever we do in our world more efficiently. For example, I used KIRO and asked it to create a simple Pinger application in python and provide me the code (basically a pinger type of “app’ is the Hello World of Networking). It produced the code, I copied the code directly into a Python instance, not in my IDE, and it copied flawlessly. I hit enter and it worked perfectly.

Another example are using tools like Mistral where you can upload a network diagram and it can provide a base configuration or give you details about your network when you converse with your documentation(there is that democratization of your data).

So eventually all those Claws, Codex, Base44, MCPs, Claudes, Cursors, Replits, CoPilots, Gemini, Zencoders and so on will mature and cannibalize themselves out to the remaining best(we hope). Remember the browser wars? Then they will be integrated into your management systems, cloud tools and possibly locally on the networking devices as a utility to create a bot applet – similar to Embedded Event Manager scripts in our old friend the CLI based network OS.

For a bit of perspective I have been in IT for a while, I am a young ancient. In my youth it was termed “MIS’ or “DP” or just “High Tech” and I coded in Basic, COBOL, C, x86 ASAM, Visual languages, TCL, Perl, and a host of other scripting types.

And yes even in 1994 we had CASE (Computer Aided Software Engineering). I was Oracle DBA Master and Oracle CASE Master certified and used these tools.

Guess what? Us experienced folks had similar tools using Decision Tree based “AI” as also our protocol analyzer’s (Network General and HP/Aglient Internet Advisor)expert systems utilized in that era.

I picked up Python a decade ago and a few years back wrote a BGP monitoring application for a Bank and a BGP EVPN Looking Glass NOC application to check the state of underlay ISIS activity and overlay BGP EVPN NLRI type 2 activity for VM workloads. My terminal application of choice ZOC has a great macro recoding for CLI using REXX which I have used for some tactical pre “DevOps” automation “applets” in the past. The tools today can take idea to proof of concept almost instantly and with good quality.

As for the “anti” Vibers out there recall we had early version of the low code/no code approach throughout our computing technology history. COBOL originally was meant to be used for the “nonprofessional programmer” to build whatever business application they needed in a natural English language syntax, “natural language” sound familiar? In its early days it was expected that a manager of a business can just write his own COBOL application and use it, no need for the DP or MIS department to code it for them. It obviously didn’t quite turn out that way but you understand my comparison. Then we had BASIC and PASCAL and the turbos and the IDEs, visual and the cloud infrastructure as coding tools and assistants that made things easier as well, so VIBE coding may be just another “re-branded” iteration of this approach. Heck, I even wrote a PC based COBOL parts inventory application for a ComputerLand franchise in the mid-80s when I was starting out as a young hardware/software technician and would have loved to have the tools we have today then.

---

*So, with that bit of history in mind I am all for these modern AI coding tools but we are treading on some risky ground when it comes to using them with our networking protocols.*

---

So, programmers today may be experiencing their art(KNUTH “The Art of Computer Programming”) being taken over by AI with massive models learning to code itself, AI agents, Vibe and NO/LOW coding for the masses and so forth, **thus** the network engineer’s CLI configuration programming art may also experience this as well. Yes, the cliché applies here, time will tell.

## 5 IN SUMMARY

---

In retrospect there are many questions, some conjecture and a little history as outlined throughout this paper that hopefully you and our OEM friends are asking themselves. Do we invent new protocols that lend themselves better in the upcoming AI and possible Quantum eras? Or, do we just progress around what we currently have and continue to make small tweaks until we hit a wall?

Do we use AI to program (converse to configure) our networking devices today with what we have in terms of our CLI language or do we invent new network device operating systems native and or protocols to AI to converse with to manage our networks?

We may be on some collision course with AI and networking for the possible worse while vibe coded agent applets help or muddy things until then. So, again, how much longer can we stay with this same sandwich?

And yes the old axiom of “if it isn’t broken then don’t fix it” applies too but for how long?

Will harmful economic competition possibly occur? What I mean is today our aging protocols are built on international consortium, agreements and treaties, what if one country figures it out, the whole sandwich, better than the others for an advantage (imagine what that could be?). Do they share, export or extort and do well economically or do they hold onto it and revolutionize their sovereign society above others first? Who knows, but the middle meat is getting old still.

But to pivot the discussion for a bit of context we kind of blew it with social media and society and we are currently waffling on AI regulation for the better – read Anthropic and the DOD, throw in all the smart glasses with everybody capturing everyone else’s life directly or indirectly and the personal agents for all sorts of things which may be accelerating some societal and mental health issue, yet we didn’t get it right with social media but hopefully we regain some balance and AI is more use to society than detrimental.

For the new generation of Network Engineers what is old is new again in terms of possible transition and it has always been that way for decades, industry and market oscillations come and go so with AI and its integration. The new network engineer may be privy to a whole new middle part of the sandwich that us experienced folks experienced ourselves as “new” in the 70/80/90s. Embrace it and enjoy the ride..

While we work this whole AI and networking affinity thing out together through our imagination, creativity, scientific method steps and continue to debate its use for the good the bad or ugly we are living in interesting times, exciting and scary indeed, but hey, weren’t and aren’t they always....

So, Learn, study, revisit and know your history to manage your present and future while embracing all things new and keep that mind open.

Hack the Planet....