Internet is technology 30 years old that need to change. In early 70’ internet as we know it took shape. Then we had only one protocol used for the routing and for transport aspect. IP was made as core layer that allowed both underlying network and overlaying applications to develop separately. In the 80’ it was adopted at universities at Berkley with its UNIX distribution. At the end of 80’ congestion problem appeared. 88 was first internet worm. ISO vs. Darpa had their war over internet architecture domination. In 90’ with IE masses started to discover internet. At the end of 90’ because of its popularity first address problems came up. Ipv6 appeared as solution to insufficient addresses. 2000’ is internet patch time with Nat. Until today we still do not have clear solution to problems of current internet. One of the reasons is that every new thing introduced here is very slowly adopted. There is around 15 year networking innovation cycle (from idea to widespread use).

**Addressing**

With 32 bits of address we could in theory have 4 billion of addresses, but addresses are allocated to subnets and not all of the bits are used and cannot be further shared. Appearance of IPv6 with 128 bits fixes this problem, but introduces its own that stops it to be adopted in use, mainly scalability in default-free-zone due to bad address aggregation.

**Naming**

DNS is additional application running over internet that is so well adopted that we can consider it as a true part of internet. In case of DNS dies, whole internet would die. This type of single point failure is not got for the whole system. One example of the panic that is caused by potential of DNS failing is shown with Kaminski bug. Another DNS problem is that it is not fast enough for support mobility. It needs hours to update address in case of changing places. DNS is host-centric, and usual use of internet is all about the content, about the data. It is inefficient in that way, because if we want to get to some data we need to go over host address even if we don’t care about it.

**Routing**

BGP that manages communications between autonomous systems behind the scene is known to be unstable. If something is changed in one corner everybody needs to know about these changes. For the conversion to happen it needs a lot time. It is called BGP flapping (some AS thinks that it would have to go here and next minute there). Nest problem with BGP is that people have contracts for some routes that are mapped on BGP connectivity graph, and that is just an abuse of the system.

We have the example of Pakistan redirecting all the traffic from the YouTube, to its sort of black hole. Here we have the problem that whole internet works only if everybody is behaving correctly.

Another problem is that end users cannot influence routing at all. There is a possibility for doing that integrated in IP called source routing, where every user could influence the traffic that passes through with table of routs. That feature is disabled because of security reasons. There is concept in theory called resilient overlay networks that enables possibility of choosing different route if current one is down. It is difficult to implement it in practice. There is also **multipath** that would be nice feature that would be nice for bandwidth. Only problem is that TCP cannot distinguish between packet taking a different route and packet loss and performance will suffer. Mobility is also not possible, and it would be nice to have functionality that you are recognized no matter where you are in the network. Each time we connect to different place new TCP session is created. Main reason is that addresses today represent name and location of the node. And you cannot take your IP address with you.

**NAT** is the solution for the address shortage in IPv4. The way it works where it allows that more addresses exist behind one address, is violating one of the basic internet concepts. With NAT there is no end to end communication, and all nodes in the internet are definitely not equal. End to end communication is useful for example in video conferences.

**P2P** as a direct communication between two nodes, opposed to communication over server is not a problem as such. Problem lies that it is implemented on application and not on network level. It should really be one of core functionalities of internet. (CNN web page needs to replicate its content to relive the load on the server).

**Sensor network**

Also known as network of things has a problem that every little sensor needs to have IP address to be able to communicate over the internet. That is now definitely not possible as we don’t have enough addresses for people. It would be nice to have automatic address assignment behind DHTP or something like that. Also IP stack potentially too heavyweight for small sensor. There is also problem with battery of the sensors. They are most of the time of, and it is not possible to communicate with them over current protocol.

With **always on** policy, nodes are not able to choose if they want to accept some packet (traffic) or not. End systems should be able to actively choose if they want to accept communication, when and from whom. It would be nice to allow only communication to some chosen node and for some time. Today we have NAT and firewalls that help hide node itself, but it is not final solution.

**QoS vs. Overprovisioning**

Internet was always best effort network, so it doesn’t really have quality of service as such. We cannot specify for example how much packets can be lost, delay of the packet etc. This is important for VoIP and video on demand. Instead we have overprovision where we have much larger bandwidth that we actually need and when we have traffic burst it can be piped trough network. It is not good approach in the long run. It is quite opposite of green computing because we use much more energy that we need most of the time.

Internet becomes a battle ground between huge economic powers for commercial interest. It is not designed for such thing, it is assumed that everybody is nice to everyone and collaboration is necessary. Idea is that new internet architecture be prepared for commercial type of dividing.

Third plane besides data and control plane is proposed, called knowledge plane. It would be kind of extension that would allow internet to self-manage to some point. Ideal would be that we stop babysitting it and allow some artificial intelligence to do most of the work, keeping the Internet running.Net Neutrality is debate between providers and the rest of the world. Providers would like to have some preferred services and sell them as such. P2P would be easily banned, QoS would be provided for services like http, VoIP etc. The other side says that reason for the success of the internet is in allowing different functionalities to develop freely. Google and Verizon proposed second pipe for providers to play with, but that could possibly mean that providers can neglect old internet pipe and force people to buy services on new pipe.

**Active Networking**

Network evolution and network customization is very slow. Inventing and deploying new protocols takes very long time because we are not allowed to change the core of the internet. With active networking we could have mobile code all over the network that can provide late binding. It could deﬁne internet functionality at run time, instead of compile-time. This idea is developing very slowly. It has been initiated in early 90’ and just this year we have first baby steps towards active networking. Open Flow and the Open Network Foundation initiative from 2011 makes switches programmable, so the flows of data can be manipulated.

**De-Verticalization, Virtualization**

Cisco sells its routers as a packet that includes hardware, software and usually controls management. They integrate everything in one product where you don’t have option to buy just hardware or some other piece. You can only buy whole package. It would be nice to buy only the hardware and in time all the other software parts can be changed implementing new functionality or maybe customize it to special needs of some user. Actually Cisco has a project of network virtualization, where they plans to offer resource slices through switches, and user could do whatever he wants with it.

**New Internet Architectures**

There are lot research going on, but there is no clear plan for the future. Instead we are focusing more on new additions for the current internet like the cloud.

**Content-centric Networking**

This is proposal of new networking architecture that is supposed to solve majority if not all the problems that we have with internet toady. At almost first statement it offers optimal content distribution, painless mobility, wireless, virtualization, same scalability & efﬁciency as TCP/IP, simple, secure, robust conﬁguration, an easy, incremental, evolutionary path, much better security. It is still in research phase so many of these promises are currently only just that, promises.

Core idea revolves of CSN around data. Internet experience of end users is all about content. Everything else is more-less just tool to get some content. There are two packet types: interest (a question) and data (an answer). Main difference in relation to IP is that packets do not have source and destination information. Packet is supposed to find data only by its name. Nodes that today represent only connections, in CCN have storage for data on every node. With this we have option to get data from closest source, and not always form original one. In case that node is not on the node request will be forwarded to some other node. There is also no need for nodes to be always on with this approach. This is also good for integration with sensor networks, where sensors do not need to be up all the time. They can just collect or store date when they are up, and then go to sleep again. Another thing is security based on container that is abandoned here. Data-based security is introduced. Data is paired with its name, its description and then signed. Source of the data has private key for signing, and everybody else can get public keys for using data. CCN has no key distribution problem since keys themselves are content. There is still need for name lookup over some third party application, like Google.