**Tau** differs from AnA and Omnetpp because it focuses on transport layer. Reason for that is that all services happen on the transfer layer. Function composition is done here. Sender is the one that decides which components provide the service. Receiver is informed with look-up table at a transport message meta-header. Header descriptors show the protocol function identifier with type, size and location protocol header. Then receiver knows what function to apply on what part of the package. He only needs to have functions that are specified in the message. Server is that one that dictates everything.

Messages issue interrupt every time they arrive, and then enter the queue. Scheduler then takes the packet and adds it to one of waiting functions and by that makes it to running function. When running-function finishes it goes to sleep again.

Problems can happen with show-stopper functions. Some functions X depends on Y. If Y needs another packet to finish, it blocks entire system. Solution for this is to detect show-stopping functions and execute them first. If functions are not logically independent we cannot do that, so we need to is call another function within another that solves the problem, we call Tau within the Tau.

**Active networking**

Task of programming the network is hard by itself but, it is completely different to implement it. It wouldn’t be possible to say all the users to start using new thing at once. Better approach is to try changing it flow by flow in a way that users can choose when and if they want to use new functionality. New implementations need to be compatible with current network system.

Idea is to provide instructions instead only data. That way we can say to every node which is on the path to process our data as we like it. Only thing that is needed is compiler or interpreter of code we are sending on receivers end. We send imperative code that gives instructions what to do. In some way we send the message with the messenger.

**Fraglets**

Nodes are reactors that have collection of fraglets. When two fraglets meet they react with each other. When they react something new is produced. First word in fraglet i the command and it triggers the reaction. We have two types of reactions of fraglets: Transformations and Catalytic Reactions.

There are a lot of possibilities to combine fraglet functionality. And they are still research project.

/////////////////////////////////

Re-write packet headers on-the-fly changes the operation on the next hop. I.e. address, flags for receiving quality of service, after 3 hops returns…

Source routing: First node dictates an explicit path to be followed across the network.

Emulate network conditions: Failures, lousy communications, congestion…

Move code and collect data lying in the network: For some sensor network we can send message to collect data and to return it to us. For example: temperature.

Simple Confirmed Delivery Protocol (CDP): sends back acknowledge.

Everything taking into consideration fraglets are just tool to implement chemical network and its ideals.

/////////////////////////////////////

**SelNet**

Idea of SelNet is based on adding functionality in the packet. Problem is that packets have limited size, so we put function call in the packet. We basically send function pointer via message. If the receiving node has library of functions, message will trigger the function. Forwarding function today uses the same principle, but it is limited only to forward function, so in some way it is generalizing functionality that is already here.

SelNet has the tree parts. Selector is kind of function holder. It is the entry point of the function. Simple Active Packet Format is type of packet used in SelNet that has some kind of agreed form, with selector and way to interpret selector. eXtensible Resolution Protocol is thing what gives user access to the selectors to be able to use them.

Every node has a table that contains descriptor and function that correspond to it. Selectors are not globally known. Every node can add its own selectors to functions in the table. To discover selectors of the nodes we use Selector Resolution protocol. First request is broadcasted, and every node that gets it sends back its location together with its selectors.

**LUNAR protocol** is a protocol that maps name of some node to path, without knowing intermediate nodes. In SelNet, node A asks around to find node B via some link layer (Wi-Fi, TCP…). If node doesn’t know where B is it just forwards request. When request gets to B he sends its selector back via same route. All the intermediate nodes do the same, until a gets single selector of its neighbor that leads to B. It is some kind of chain setting the selectors on the path to B.

**Highway routing**:With SelNet we could influence routing in many ways. We could have many selectors on each node to choose the way of routing we are going to use. By that we can take many different paths to the same destination considering some criterion. We could choose regular routing ore take some predefined quick route via MPLS - Multi Protocol Label Switching. Today we can’t do something like that because providers are forwarding traffic for us.