I have been scouring the internet for articles about internet traffic hacking or data interception with FPGA, but instead I ended up with this article : ***FPGA systems for preventing TCP SYN flood attacks***. This is not really what I was looking for, but it will have to do.

Introduction
When dealing with internet applications or webpages and the like, they are usually hosted on servers and to gain access to these applications your local computer has to request a connection to the server. For each request the server sets aside resources, fx RAM, and since it is a finite resource, only a finite amount of connections can be established.

Problem.
So what is the problem with this:
The server allocates the resources as soon as a computer tries to establish a connection, and it keeps these resources allocated until it receives a connection termination signal.
The problem arises when we open up a lot of connections without terminating them. Eventually the server will run out of resources, which prompts a denial-of service to any further connection attempt.

This is basic idea behind the ***Distributed Denial-of-service*** attacks, where a botnet repeatedly sends request to the server and the ***Slow lorries*** attacks where the botnet deliberately is slow to respond, which means that the connection is kept open even though a timer on the server is used to prevent the DDOS attacks.

Related work
The paper introduces several methods for detecting and mitigating these DDOS attacks and I would recommend looking into the references if this is interesting, but to mention a couple we have the ***SYN-COOKIE, SYN-CACHE*** methods***.***

These rely on allocating the least amount of resources possible, until an ACK is received.

Specification of the problem.
The paper deals specifically with the TCP-SYN flood attack that exploits the 3-way handshake of the TCP protocol.

To understand what they are doing, we need to know how a TCP connection is set up.
Usually a client sends a TCP-SYN package to the server, which is the request of connection.
The server has to respond with a TCP-SYN/ACK package, to indicate that resources have been allocated and that it is ready to establish a connection.
To this the client has to respond with a TCP-ACK package, and the connection is now setup.

If the last ACK package is never send, then we have the problem.

Proposed technique.

The proposed technique applies an FPGA between the internet and the server and the FPGA will then run the SYN attack/prevention algorithms.

Not to go into too much details, the FPGA is basically setup to have 2 separate registers, a GOOD and a BAD, with 1 counter per register.

Both registers store the client IP and PORT addresses and the destination IP and PORT addresses, plus the counter value for entry in the register.

There are 2 algorithms running on the FPGA.
The first one intercepts a package and determines whether it is a SYN package.
If it is, then the GOOD register is checked to see whether the addresses already exists there and if not, the addresses are stored in the GOOD register, the counter are incremented and the package forwarded.
If the counter is more than 1, meaning that 2 or more SYN packages has arrived from the same client, then the addresses are removed from the GOOD register and put into the BAD register.

If there is not a hit in the GOOD register the second algorithm checks the BAD register.
If a set of addresses are in the BAD register and a SYN package arrives, the algorithm, the FPGA, sends the server SYN/ACK package. If the next package is then an ACK from the client, the algorithm establishes the connection with the server.
If the next package is again a SYN package, then the client is banned until the FPGA receives the amount of ACK packages corresponding to the counter value in the entry for the client.

Advantages and disadvantages of proposes technique.
The paper explains that the advantage of the proposed technique is that the FPGA will handle all the extra needed computations which will save CPU utilization on the server.
The next advantage is that the delay in packages will be greatly reduced for an already established connection, because of the FPGA's parallel execution.
This is due to the fact that both algorithms can be executes simultaneously while the FPGA forwards packages between clients and server.

The main disadvantage of the technique is that for a frequently visited server the FPGA's memory capacity can set an upper limit to how many connections can be made at a time. This is because the FPGA has much less memory when compared to the server.

Results & Conclusion
The paper doesn't go into details about how the test of the system is done and the achieved results. But it explains that to test the system they used a dataset, of which they are linking to, and that they successfully identified all TCP-SYN attacker addresses from the dataset.

The paper concludes that the system is a definite improvement over traditional TCP-SYN identification and mitigation methods, but that the solution is not perfect.

The paper refers that future work has to include efficient memory management algorithms and faster methods for matching addresses in the GOOD and BAD registers.

And with this I will end my presentation of the paper.