SOFTNET - AN APPROACH TO HIGH LEVEL PACKET COMMUNICATION

Jens Zander and Robert Forchheimer
Dept. of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

Abstract

SOFTNET is a packet-radio concept under development in Sweden. The network is distributed and all nodes are programmable via the network during normal operation. This concept represents an unconventional approach to the protocol issue and offers elegant solutions to the higher level communication problems. This paper gives a programming model of the network, along with some illustrating examples.

I. Introduction

The SOFTNET approach was conceived in 1980 and discussed among Swedish radio amateurs. The discussion led to a proposal for an experimental network in the 432 MHz band utilizing bit rates up to 100 kbps. During 1981 this draft was presented to the Swedish Telecommunication Administration. The Administration responded in a positive way, giving the packet radio group at Linköping University virtually free hands. This group, consisting of 6 people, is currently involved in developing prototype nodes and basic software for the network.

The main concept behind SOFTNET is that all packets are considered to be programs of a network language. These programs are interpreted in the nodes as soon as they arrive. Nodes can be programmed by any number of users simultaneously without unwanted interaction. This approach makes it possible for a user to define his own high level services like datagrams, virtual calls, file transfers and mailboxes. The concept also allows changes at lower levels during operation, permitting redefinition of LINK-level/Access protocols. A detailed description of these ideas can be found in [1],[2],[3],[5].

II. Node model

In a SOFTNET node, an incoming packet that has passed the link level is given to the node computer for interpretation. Here, a standardized set of instructions are available. The kernel of this set is simply a FORTH interpreter to which has been added functions that control the node hardware. Thus, any user may execute his own FORTH program in any of the nodes that he can reach. This way he is able to instruct another node to either deliver the packet to the owner of that node or to retransmit it so that the node merely acts as a repeater. FORTH allows the creation of private directories so the user may also store programs in remote nodes. These programs may either wake up upon the arrival of a packet from the user or upon an internal signal (e.g. the real time clock) produced by the remote node itself. Describing the node thus reduces to describing a programming model. In the FORTH case, this is done by simply listing all the available functions or "words". [4]. Fig. 1 summarizes the packet format from the user's point of view.

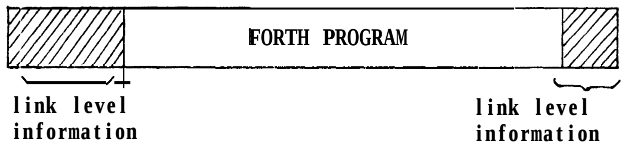

Fig. 1. A SOFTNET packet

In fact, the link level protocol has been added to the FORTH kernel so that also the link level information is handled by a FORTH interpreter. This permits on line reprogramming and extension of the link protocol such as new version of HDLC, access algorithms etc. Thus, from the first byte to the last, a SOFTNET packet is simply a set of FORTH statements. From a practical point of view it is a good idea to conceptually keep instructions at the link level apart from the higher level programs since changes at link level have to be coordinated among the users.

III. The Node - a multiuser/multitasking system

Processing at the link level requires real time performance while higher level tasks are less time constrained. On the other hand, the link processor serves one packet at a time sequentially while higher level tasks may run concurrently. Also, the programming activities of one user should not influence any other. Thus, a SOFTNET node must be able to support parallel tasks besides being able to keep apart the current users of the node. For the prototype implementation our choice was a dual processor (6809) system. One of the processors are solely devoted to link level processing. The second processor contains a multi tasking FORTH interpreter and is shared among the users. A special task - the owner process - interfaces the node to the owner's equipment which can be anything from a dumb terminal to a fullgrown computer system. In the latter case the dual processor FORTH system is simply considered a modem between the owner's system and the network.

IV. Node programming example

Consider the simple network given in fig. 2. Here four nodes are connected by two-way radio paths as indicated by the arcs between the nodes.

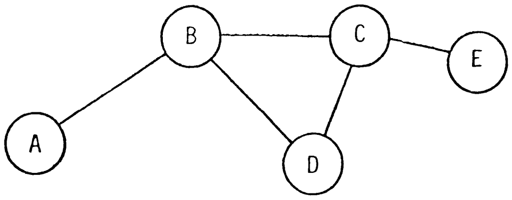

Fig. 2. Sample Network

Suppose a user is located at node A and has the specific task to deliver a large number of packets to node E, i.e. he wants to establish a "virtual" call to E. This can be done in at least two ways. The simplest thing to do is just to add a retransmit command to all packets as is shown in fig. 3a). The command % takes the next symbol as an (one-hop) address and transmits the rest of the packet to that address. This goes on, dropping one address each time, until the remaining packet reaches node E were the data portion is transferred to the OWNER of the node. This procedure may however consume valuable packet space, especially when many intermediate nodes are used. We can instead make use of the programming

```
%B   %C   %E   OWNER <data>
```
a)

```
%B   : VCE ." VCE" %C ;
```
b)

```
%B   %C   : VCE ." OWNER" %E ;

: VCE ." VCE" %B ;

VCE  <data>
```
c)

Fig. 3. Node programming

feature and instruct the intermediate nodes B and C just to pass along the packet to the next node in line. This can be done as in fig 3b). Here we define a new function, named, say, VCE in the intermediate nodes and our own node. A new definition is made FORTH-style starting with a "• " and ending with a ";". The effect of executing VCE is to pass on the rest of the packet to the next node in line. Also, the function places a copy of its name first in the packet for repeated execution in the succeeding nodes.

V. Project status

Since the advent of the project at Linköping University, a rapidly growing number of interested radio amateurs have joined the discussions. A SOFTNET user group, SUG, is being formed as a subgroup of AMSAT-SM. Up to date this group has received about 1.00 applications for membership.

Hardware development has made considerable progress. The Node-computer board is under production and a first shipment of 50 kits was delivered in February. Also the PC-layout for the LINK-computer board is completed. The packet-radio utilizes a duobinary direct FSK modulation scheme with favourable bandwidth properties. Transmission is synchronous and MFM coding is used to recover clock information. Due to problems in the design of the radio testing of the digital hardware-and software had to be done on a cable bound local network. A system with up to 4 nodes has so far been successfully demontrated and has provided useful results for further software development.

VI. Conclusions

The SOFTNET concept with its fully programmable nodes will give the user opportunity not only to communicate, but to conduct experiments in network architecture and network protocols. The concept is applicable to all kinds of communication networks. An implementation using a local network cable has been successfully tested and a UHF-radio broadcast network is under construction [2].

VII. References

[1] Persson, I., Forchheimer, R.: Design Considerations of a Distributed Packet Radio Network using the Amateur Radio bands, Internal Report, LiTH-ISY-I-0408, May 1980.

[2] Zander, J., Forchheimer, R.: Preliminary Specifications for a Distributed Packet Radio Network for Computer- and Radio Amateurs, Internal Report, LiTH-ISY-I-0424, January 1980.

[3] Zander, J., Forchheimer, R.: Softnet-Packet radio in Sweden ARRL Amateur Radio Computer Networking, Conf. Gaithesburg, MD, October 1981.

[4] Forchheimer, R., Zander, J.: Softnet - User's Manual, Linköping 1983.

[5] Qvigstad, F., Matts,, S.: Construction of a packet radio Node computer, Internal Report, LiTH-ISY-I-0491, December 1981.