Warren: A protocol for control of ATM hardware.

Dr. D.J. Greaves and Mr. R.J. Bradbury

ATM Forum Document Number: ATM_Forum/98-0008
ATM Ltd and Computer Laboratory
University of Cambridge

```
SOURCE:         David J Greaves, Richard Bradbury
                ATM Ltd (+ University of Cambridge, UK)
                Mount Pleasant House
                Cambridge   CB3 0BL
                Phone:  +44 1223 334636
                Fax:    +44 1223 566915
                E-Mail: djg@atml.co.uk
```



```
DATE:           February 1998

DISTRIBUTION: AF-RBB Residential and Broadband
```


Abstract

This contribution proposes that the ATM Forum accept the Warren Control Protocol (WCP) as the recommended means for low-level control for line interface chips, endstations, and dumb switches. The protocol has been designed and implemented with the home environment in mind, but it is suitable for all complexity-sensitive applications, such as a ceiling array of sensors. The system has been extensively tested and implemented in over 100 prototype devices.

This contribution was motivated by the contributions on 'A simple Device Protocol for the NT' and 'An in-band configuration Protocol' presented at the last Forum meeting [1], [2].

1 Summary

The 'Warren' ATM technology is described in this month's (Jan/Feb) issue of IEEE network magazine [3]. The Warren approach to low complexity ATM was developed jointly by ATML and researchers at the Computer Laboratory of the University of Cambridge in the UK.
See http://www.cl.cam.ac.uk/Research/SRG/HAN.

Warren allows minimal complexity end-systems, whereby the usual functionality of a network interface adaptor is trimmed down to include only those functions required. In this way, an ATM port can be added to an end-system in the form of a 3000 gate macro-cell added to one of the normal system chips. Typical end-systems would be a digital TV set, camera or DVD player. No software is required in the end-systems since devices are under control of proxy software running in a single host that can handle hundreds of Warren end-systems. The proxy controller communicates with the end-system for basic control using single-cell messages which can be handled and generated within the macro-cell. All normal ATM flows are unaffected.

Not only end-systems, but a mesh of Warren switches can be controlled by the single controller computer. The switches respond to single-cell messages for topology determination, management and circuit setup, while the controller keeps track of the overall resource allocation and call admission control for the mesh of switches. Warren can be used to control a Telco-owned fan-out switch at point of entry to a home. Alternatively, a home owner can have their own Warren controller to manage a HAN (home area network).

The Warren Control Protocol is also intended as a substitute approach to control and management of subsystems within the I/O ports of an ATM system. Example subsystems are the accounting, performance monitoring, policing and PHY functions of an ATM switch port or remote port on a DSLAM line shelf. The need for separate address, control and data lines to these subsystems is removed and hardware design can be simplified since dual-porting, clock synchronisation and arbitration for critical structures disappears.

2 Details

This written contribution presents the core of Warren Control Protocol. For further information about how we have applied the system in real homes, see IEEE Network Magazine and the WWW pages.

The Warren Control Protocol uses single-cell messages on a pair of dedicated VCIs. These VCIs are reserved for Warren Control Protocol use and all other VCIs act normally.

3 Payload Format

First we describe simple commands to devices from an adjacent master and later we explain how a Warren Subnetwork, or more simply, a 'Warren' can be constructed that includes switches, busses and loops.

As shown in figure 1, commands in Warren Control Protocol Cells use only 40 bytes of the cell. Normally commands are placed at the start of the payload with 8 dummy bytes at the end of the

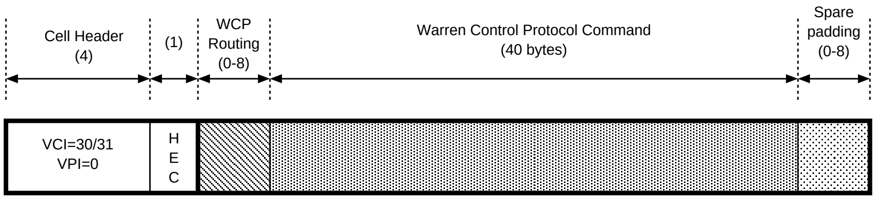

Figure 1: Structure of the ATM cell as used for the Warren Control Protocol

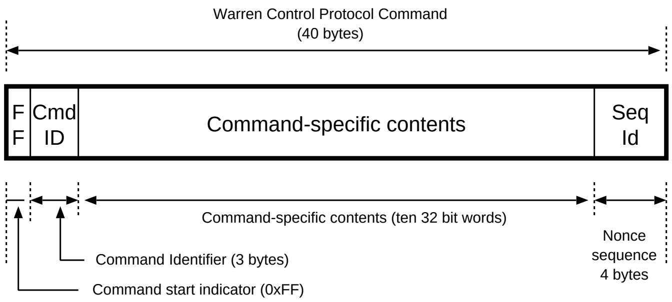

Figure 2: Warren Control Command format within payload.

payload. However, up to 8 routing bytes can be inserted before the command, thereby moving the command to the rear of the payload. The first byte of all commands is 0xFF and no routing byte has this value so that the presence of routing bytes is easily detected. Routing is discussed further below.

4 VCIs in use

Two VCIs are used by the Warren Control Protocol: one to the device under control (the 'down' direction) and one from the device under control (the 'up' direction).

Where the Warren Control Protocol is the being used to control subsystems within a piece of ATM equipment, the numeric values of the VCIs used is a proprietary issue since the Warren Control Protocol cells never leave the piece of ATM equipment as a whole. The choice of VCIs (or whether to use VCIs at all) for this situation is discussed later.

Where the Warren Control Protocol is being used between standard items of ATM equipment where one (or both of them) are being controlled by the Protocol in the intended fashion of the ATM Warren, then the controlled device (or both of them) expect to send and receive the Warren Control Protocol cells on well-known, standardised, hardwired VCI values. All of our prototypes currently use VCI 30 for down and VCI 31 for up. Clearly we would be happy to change to other values if discussion in standards Forae advises us to. In this configuration, control cells should be sent and received on VP zero. Clearly these VCCs can be tunnelled and switched over a regular

ATM network and only need have these numeric values at the equipment implementing and being controlled by the Warren Control Protocol.

5 Ping Pong Nature of Commands

Each Warren Control Cell that is sent on the down channel has an expected response cell on the up channel. A 'nonce' field in the command is preserved by the device so that a controller can correlate commands and acknowledges as desired. In keeping with a protocol aimed for minimal complexity when implemented in hardware, the difference between a command cell and its reply may often only be a single bit.

6 Peek and Poke Commands

The most important commands are peek and poke. These commands contain a 32 bit address field that addresses a 'virtual' space inside the controlled device. The commands contain a size, giving the number of bytes to be peeked or poked from the virtual address space. They also contain a data field containing the data to be written into the device (poke) or read from the device (peek). The reply to the poke is normally only different from the command in that the response bit is set. For the peek, the reply will have the data field updated as well.

7 Device Type Identifier

All devices have a type identifier which describes their make, model and function, together with as much additional information as desired. This type information is mapped into the device internal address space at virtual address zero and is in a self-describing form. A primary function envisaged for this information, although not part of the Warren Control Protocol itself, is to contain URL's or BBS telephone numbers and addresses etc., so that in a home environment, a dumb device can cause the download of software into a home controller to look after it. A short form of the device identifier info is included in the beacon cells described in section 10.

8 Use of Virtual Address Space

Apart from the device identifier information, the virtual address space can be used freely by any particular device. Software controlling the device requires something akin to an SNMP MIB so that it knows exactly what is in the address space, and as mentioned, these MIBs can potentially be fetched over the WWW based on device identifier information.

For a line interface chip (PHY device) the virtual address space can include a map of the normal management registers. For a policer or simple ATM switch, the tables of per-VCC statistics and control information can be mapped in. For very simple home devices, the address space can include various control and status information.

However, for most first-class devices, it is just as easy, in hardware, to implement control and management functions on separate VCIs. This is the approach we have mostly taken in our implementations of devices. The advantage of separate VCIs is that the ATM system can route the control and management VCCs directly to the computer which is controlling the device. In contrast, as mentioned below under routing (section 9), the Warren Control channels are always routed to a controller based on physical topology. This controller may not be the same machine (or VCC service access point) as the Warren Controller.

9 Routing

8 bytes within each Warren Control Protocol cell are reserved for routing. The number of routing bytes present may be between 0 and 8 and these fall at the start of the cell, pushing the command itself towards the rear of the cell.

The Warren Control Protocol uses self-routing, also known as source-routing.

A device implementing the Warren Control Protocol only executes commands sent to itself. These commands are identified by being on the down channel and having no routing bytes at the start (i.e. the first byte is 0xFF indicating a command start). Bus devices, described in section 13, deviate from this rule slightly, in that the 0xFF is in the second byte.

A device implementing the Warren Control Protocol that receives a command on its down channel that has routing bytes at the start of the command should route the cell if it can, removing the routing information used as it does. Removal is performed by shifting the contents of the payload forwards and adding dummy bytes at the end. This shifting process is very simple to implement in hardware.

Similarly, command responses received on the up channel should have routing bytes added at the start. These routing bytes indicate the input port (or more generally, input identifier) that the command arrived on. Responses are routed 'upwards' based on the spanning tree for mesh networks described in section 12. For non-mesh networks, such as bus and loop, see that section of this contribution.

10 Beacons and Hello Keep Alive.

Warren Devices indicate their presence by beaconing. Beaconing is the process of sending a Beacon cell on the up channel at a frequency of about 1 Hz. Beacon cells contain a short description of the device. Beacon cells accumulate the route to the device as they are propagated upwards towards the controller. Therefore the controller knows how to contact them.

Once discovered and under control of a controller, devices should stop beaconing. To let the device know that it has been discovered, the controller should periodically poll the device with a Hello Keep Alive cell. At the device, a timeout on Hellos of about 4 seconds should be used before resorting to beaconing. An LED indicator on a device indicating whether it is beaconing or not is an important maintenance feature for home use (see Section 19).

11 Warren Subnetwork and Addresses

A Warren Subnetwork, or more simply, a 'Warren', is a set of devices implementing the Warren Control Protocol on the standard VCI values under control of a single controller. Each device on the network can be addressed from the controller using the source addressing scheme and therefore has an 'address' dependent on topology. Therefore addresses do not have to be built in to a device. This is the standard approach to home equipment: consumer manufacturers are not used to including unique identifiers within their products. In particular, everybody is used to the fact that a telephone takes on the telephone number of the physical socket it is plugged into.

12 Routing on a mesh part of a Warren network

An arbitrary mesh of switches and devices is the generic topology of a Warren network (figure 3). ATM switches implementing the Warren protocol need to know which port is their up port in order to forward Warren Control Protocol cells to the controller. When each switch has an up port, a spanning tree is thereby formed, rooted at the controller.

The spanning tree is formed anew if a whole Warren is switched on at once and it is updated as parts of the Warren are connected or disconnected. A newly connected switch sends beacon cells on each port. If there is a route to the controller, the beacon cells will arrive at the controller with this route prefixed in the payload. If there are multiple routes, several copies of the beacon cells will arrive, each with different route information. The controller selects a route that it desires to use to communicate with that switch and sends a beacon acknowledge back to the switch using that route. On receipt, the switch stops beaconing and accepts the port that the acknowledge message arrived on as the up port for that switch. In this way, a new branch is added to the spanning tree. The route which a controller selects to use is typically chosen to balance the tree as much as possible and perhaps chosen so that spanning tree addresses of devices do not change. In our higher layer software, we have developed the concept of 'Soft MACs' which are addresses for devices maintained in a database in the controller that are preserved as far as possible over spanning tree modifications. This eases applications.

Non-spanning tree links in the mesh may be freely used for data, but do not normally carry Warren Control Protocol messages. Their presence must first be detected by the controller. It does this by sending a 'cross link probe' Warren Control Cell out of each port on a newly connected switch. When these are received back at the controller, their routing information reveals the destination port address of the cross link and the controller knows the source link address by correlating with the nonce value it used.

The structure of addressing information used in mesh switch ports need not be the same from switch to switch, since the controller knows the type identifier of each switch and therefore how that type of switch chooses to use its addressing information. We have used a simple structure of one byte per switch with the byte simply being a port number. Port number 255 is prohibited since this is the value to indicate a command start, but we do not have switches with that large a number of ports.

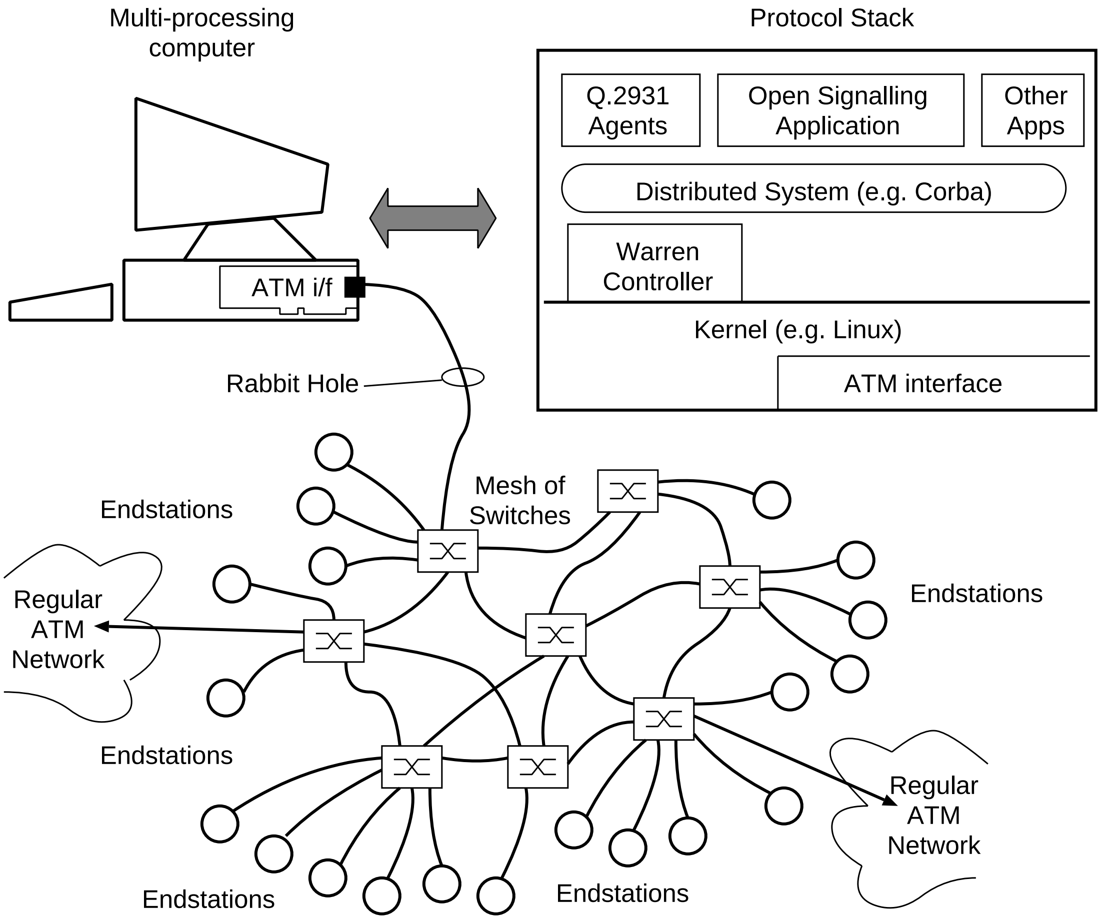

Figure 3: Physical Configuration of an ATM Warren and Protocol Stack within the C ontrol Computer.

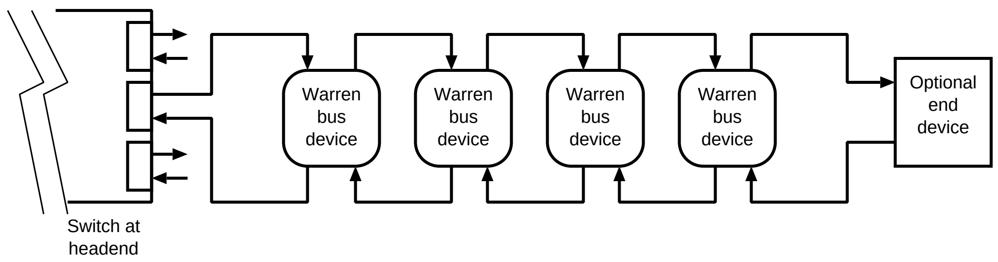

Figure 4: Structure of an ATM bus with an optional regular device on the end. The two directions of flow are normally within the same cable.

13 Routing on a bus part of a Warren network.

It will often be useful to daisychain ATM devices in a bus, connected to a switch (Warren or otherwise) port at one end. This configuration is shown in Figure 4. An important part of achieving low-complexity in such a bus is that cells are passed on with fixed delay, thus avoiding any queuing or FIFO structures that are typically found in an ATM switch port. Example applications are loudspeakers in a multi-speaker system and single-chip video cameras in a composite camera array. Each device therefore has connections to a pair of neighbours. A device acts as a drop-mux switch for its own data cells. These data cells are selected by VCI value as usual. We suggest that devices use VCI groups that are spaced by 256, thereby allowing 256 devices in a chain and 256 actual data VCIs per device.

So that devices on a bus do not need to contain unique identifiers and instead always transmit and receive on a fixed, hardwired set of VCIs, it is possible that each device subtracts 256 from all data VCIs before passing on the cell. Therefore devices are addressed for data according to how far down the bus they are. Data transmitted by devices is passed in the opposite direction with the opposite manipulations of the VCIs.

As an alternative, bus devices can have the VCIs that they use for data set up using the Warren Control Protocol. In any case, the Warren Control Protocol will be using its standard, unmodified VCI values and it too needs a method of addressing individual devices. The method chosen is that devices on a bus respond to Warren Control Cells even when there is routing information at the start of the cell. The routing information allowed is one byte, which has the value 0x01 to indicate that the command is for the current device. The value 0x00 indicates that the cell should be ignored. Cells received (value=0x01) should either be deleted by nulling their VCI value, or nulling their routing byte (or total deletion in the case of PHYs that have explicit SOC tokens such as ATM25). If the byte is higher (in the range 2 to 253), the value is decremented and the cell passed on without other action. Responses and Warren Control Protocol cells generated for the up channel on a bus device should be created with an initial routing byte of 0x01 and this is incremented as the messages is passed back along the bus.

The value 254 is reserved as a multicast for Warren Control Cells to all devices on a bus. This value should not be modified as passed on. Such multicasts should not be replied to.

The value 255, if encountered by a node on the bus, should be passed on unchanged. This value means that the command starts immediately. This mechanism allows control of any non-bus Warren end device on the end of the bus. Switches, or other devices which use the mesh routing mechanism, clearly cannot be placed in the middle of a bus and they also cannot be placed directly on the end of the bus, but they can be teed-off if they are built with a special bus port. The routing mechanism would then continue beyond the bus as normal.

14 Routing on a loop part of a Warren subnetwork

A loop, or ring network is desirable in that it requires half the number of PHY ports than a bus. In a loop, or ring, the input port of one station is connected to the output port of the previous and the loop is closed by connection to a switch port. Details of routing on the loop can be found in [3].

15 VCI values to use in subsystem components.

Assuming that the Forum or others agree on a standard pair of VCI values to use for Warren style subnetworks (such as 30 and 31 that we are currently using), the question arises as to what VCI values should be used when the Warren Control Protocol is being used to control subsystem parts of a proprietary ATM system, as mentioned above. This ATM system may be connected, at one or more of its ports, to a Warren style subnetwork per se. Therefore the standard VCI values must be supported by the equipment at its external ports and these values cannot be directly used inside the equipment for internal control and denied at the external ports. In any case, such denial (i.e. holes in the space of VCIs available) would be a negative aspect of the product specification.

The most obvious header values to use inside a proprietary system are physical layer cells on VCI=0 or cells with non-zero GFC field. Alternatively translation can be used, or a system with movable values can be used where if the value in use becomes required at a public port, then the system quickly reconfigures the values its uses internally, without disruption.

Although we have not considered this in detail, a movable translation system seems the best solution. This requires one register writable by the Warren Control Protocol and a handful of gates. It works as follows. For internal use, the device exchanges Warren Control Protocol cells on the standardised VCI values. The device also contains a 15 bit register that can be programmed by the Warren Control Protocol. When the register is zero, no special action is taken and further devices can be controlled beyond the current device (presumably still within the equipment) using the Warren Control Protocol.

When the register is non-zero, the device terminates the Warren Control Protocol on the standard VCIs thus defining the edge of the Warren within the equipment. The value in the register defines a pair of adjacent VCIs that are then mapped to the standard Warren Control Protocol VCIs, in both directions, by the device. This consumes two VCIs from the total of 65536, but without specific restriction to values. This technique means the Warren ASICs can implement just one pair of VCI values and are the same regardless of the style of Warren intended.

16 Warren Security

When the Warren Control Protocol is used to control an ATM subsystem within equipment, one of the ports of the subsystem will typically be wired towards the controller. For security and integrity of the equipment, the subsystem device should not respond to Warren Control Commands on the other ports. This configuration should be hardwired into the masks of ASICs or selected with logic level strapping on device pins.

With a free-standing Warren Switch, for maximum flexibility, the up port is soft and allocated as the spanning tree is established. However, it is possible that a rouge controller can grab control before the up port is determined. Our switches also implement a ‘clear up port setting’ command which can be received on any port. A rouge controller could use this to gain control of a whole Warren subnetwork.

A simple physical means of security is to designate one port on the switch as the only one that has the capability to be the up port. All Warren Control Protocol down cells received on other ports are simply ignored. An exception is the 'cross link probe' cell which must be supported for correct discovery of cross links in the topology.

17 Location and Form of the Controller

The Warren Controller may be in the home or may actually be situated several hops away over a further mesh of conventional (or otherwise) ATM switches, perhaps in the central office of a town. In this latter case, the controller would be a large computer covering many homes.

The view of a home controller that we have is a black box with an ATM port and either a PCMCIA slot or a modem (ADSL or otherwise). The controller contains stable storage (i.e. non-volatile) for downloaded proxy software for each module and user customerisation information. It also contains execution resources in the form of a processor and RAM.

18 Interworking and Unauthorised Traffic

The ports exposed (i.e. unused) on the outside of a mesh of Warren switches may be fully ATM Forum compliant UNI or NNI switch ports even though the switch has no software or processor. To do this, instances of protocol stacks such as Q.2931, P-NNI and ILMI are provided on a central computer (the Warren Controller or otherwise) and the VCIs required are established through the mesh of Warren switches to the controller.

It is common in ATM switches that cells received on unestablished VCI values are counted and logged. In our implementation of a small switch, unused VCIs are routed to an internal dummy port that detects traffic. When traffic arrives at this port, at whatever rate, the port generates Warren Control Protocol cells once per second. These are routed upwards to the controller and contain the input port number and VCI of the unexpected traffic.

When standard ATM equipment is connected, the standard equipment will attempt to use ILMI on VCI 5 or metasignalling on VCI 1 to communicate. This traffic will be notified to the controller through the mechanism just described. The incoming VCI is a good indication of what sort of traffic is arriving, but for further confirmation, the controller can route the VCC temporarily to an agent process that determines what sort of traffic it is. This done, the controller instantiates a protocol handler for the new traffic. In the case of Q.2931 or P-NNI, the protocol handler will wish to establish calls over the Warren subnetwork. An implementation of these protocols for the Warren is achieved without modification to the bulk of the protocol handler code by simply providing a new, specific set of signalling leaf libraries, as must be done when this software is ported to any new switch hardware.

Some of our devices are so simple that they generate cell steams all of the time. This is so that they can be plugged into a reciprocal device and communication established without any signalling stage. For instance, our microphones and loudspeakers can be plugged into each other with a simple crossover cable, as can our displays and cameras. They then just work without any software.

For these devices, which generate cells as soon as they are switched on, the agent can determine the device type from the contents of the cell stream. The switch port is then programmed to truly, fully discard the cell stream until needed. In this way, the switch routing table provides an on/off functionality for devices.

19 Maintenance in the Home Environment

I.610 operations and maintenance flows can be implemented and carried over the Warren devices in their fully standardised form. The gate count to implement I.610, excluding performance monitoring, is roughly the same as required to implement the Warren Control Protocol, and if both are implemented, many paths and structures can be shared. However, it may not be necessary or desirable to have both systems in a low-complexity home system.

As mentioned above, the best form of maintenance system for the home is based around LED indicators which show whether the device is in full contact with the controller. For a device with its LED off, the wires can be followed backwards towards the controller until a device with its LED on is found. The fault location has then been discovered and a small amount of replugging or swapping of devices will isolate and fix it.

It is also possible for the full topology of the network to be displayed by the controller and we have implemented an html output from the controller so that the topology and device details can be viewed graphically. Contact DJG for the URL of a live example.

SNMP and other management protocols can be implemented for Warren devices using further proxies on the controller. The basic information needed to be accessed by the SNMP stack is already present on the controller. No new MIBs would be needed, except that we already have a larger class of devices than MIBs have currently been written for.

20 Fixed VCI values and Complexity

The complexity of an ATM endsystem is greatly simplified by removing programmability. The following are very bulky items found in most ATM PHYs: host management port logic, control, status and cell counter registers. The other great contributor to complexity is the ability to perform SAR on multiple concurrent VCCs. Programmable transmit and receive VCI values are part of this. Although not part of the Warren Control Protocol itself, a typical deployment of the protocol for the home will also use fixed VCI values on device ports for the data VCCs. The advantages of this are

- simplification of logic
- direct plugability of devices without configuration or signalling
- the required programmability exists in switch tables anyway.

Name	Command	Remainder of Cell Description
Switch Beacon	0x85	Device maker and model number
Station Beacon	0x87	Device maker and model number
Peek1	0x92	Virtual address and one byte of data
Peek4	0x80	Virtual address and one 32-bit word of data
PeekMulti	0x92	Virtual address, byte count and up to six 32-bit words of data
Poke1	0x94	Virtual address and one byte of data
Poke4	0x82	Virtual address and one 32-bit word of data
PokeMulti	0x96	Virtual address, byte count and up to six 32-bit words of data
Hello	0x88	No further fields at the moment
Reset	0x8A	No further fields at the moment
Cross link probe	0x8C	No further fields at the moment
Unauthorised traffic	0x91	Port and VCI information

Table 1: Defined Warren Control Protocol Values. For each value, the response has the least significant bit set.

This approach enables us to implement segmentation and reassembly operations in gate counts that are equivalent to the complexity of the PHY line coder and the functions can even be combined if the logic is synthesised together. We then find that the complexity of an ATM version of a digital interconnect between devices is exactly the same as non-ATM versions, such as SP-DIF or IRDA. As a guide, in an endstation, approximately 500 gates are needed.

The complexity of the implementation of the Warren Control Protocol within our 6 port ATM switch is 193 D-types and 950 gates (figure 5). The implmentation uses a Xilinx XC4013E FPGA device. The remainder of the FPGA implements the ATM switch functions of queuing and header translation.

21 Current Situation and Future Work

About 100 Warren devices have been built using FPGAs. These include switches, phones, CD players, controllers, microphones, loudspeakers, cameras, IR-basestations, LCD display tiles etc.. A working Warren system is installed at the home of DJ Greaves and is in use every day for multi-room HiFi.

We are working on a new system, called Autohan, which defines the standard operation of home devices and also allows advanced programmability of the home network. This work is physical layer independent and encompasses lower-speed networks as well as Firewire and ATM with or without the Warren.

22 Patents

ATM Ltd has filed for patent protection of some aspects of the Warren Control Protocol.

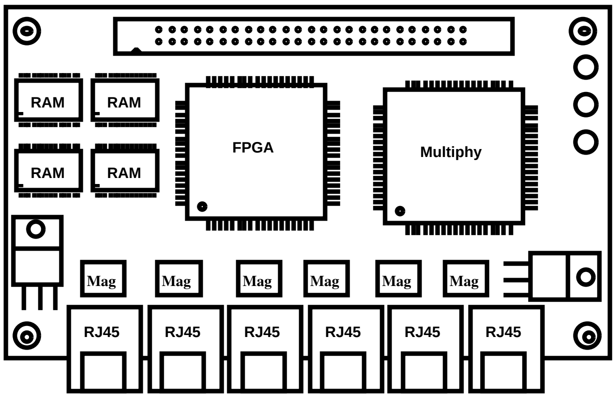

Figure 5: Structure of Warren Switch Version 1 The Version 1 Warren Switch is a six port design with ATM25 on UTP using RJ45 connectors (figure 5). A handful have been built and are in use in Cambridge. This switch is a fully working, standards-compliant ATM switch, but which uses the Warren Control Protocol on a pair of dedicated VCIs, rather than having a processor inside it.

Acknowledgements

The authors gratefully acknowledge the help of Mr Daniel Gordon in building all of the prototype Warren Devices.

References

1 A Berenbaum, R McLellan 'An in-band configuration protocol' ATM Forum 97-1061

2 D Wilhelm, J Wiebe 'A Simple Device Protocol for the NT' ATM Forum 97-1080

3 D. Greaves, R. Bradbury 'Warren: a Low-Cost ATM Home-Area Network.' IEEE Network Magazine, Jan/Feb 1998. http://www.cl.cam.ac.uk/Research/SRG/HAN/Warren

Motion

This contribution proposes that the ATM Forum accept the Warren Control Protocol (WCP) as the recommended means for low-level control for line interface chips, endstations, and dumb switches.

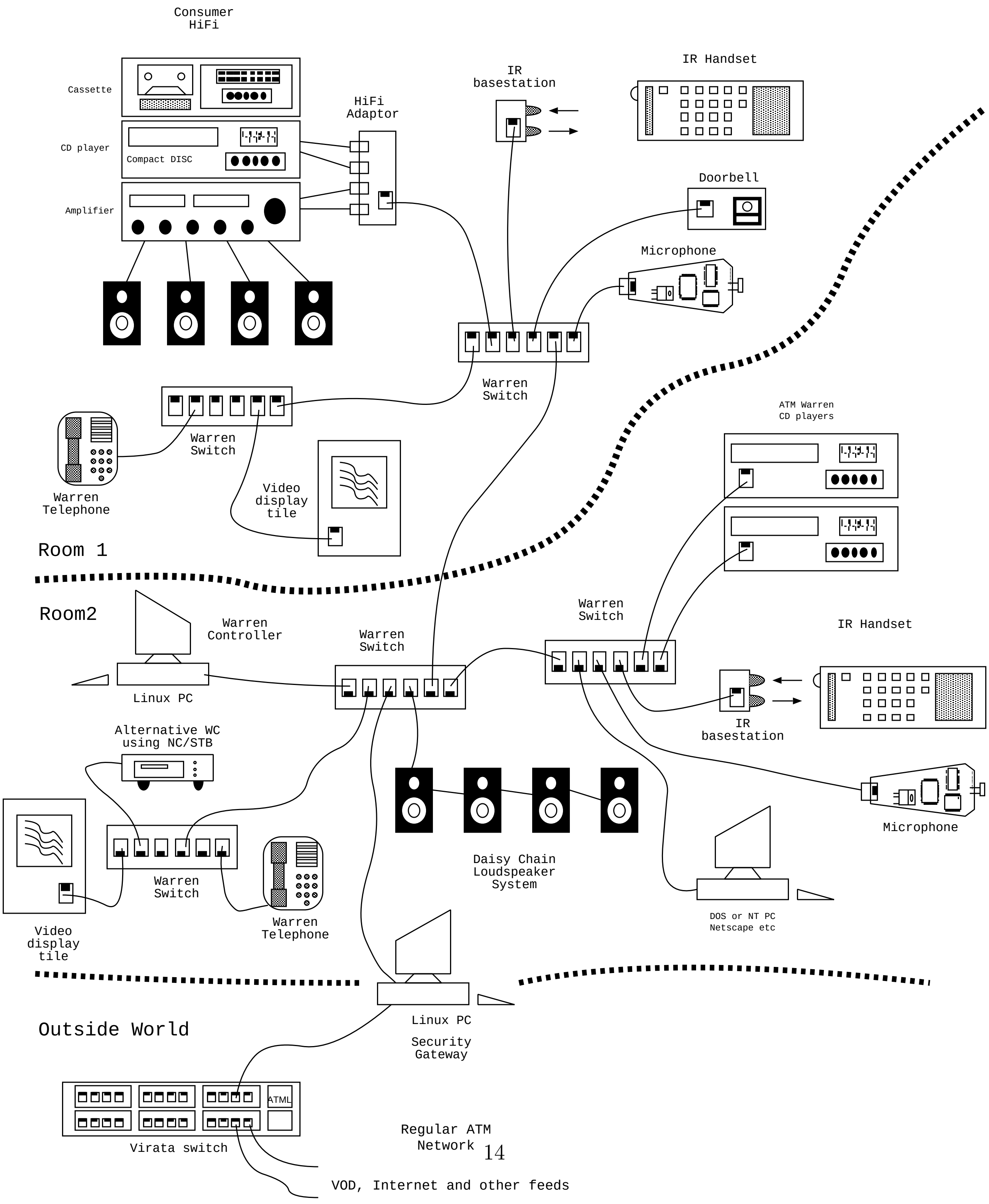

Figure 6: Typical Demo Configuration