Chapter - Control Network Elements & Network Management

1. Control Network Elements

In chapter one, we mentioned about the *local* control network elements (Fig.2.3), in this chapter, we'll further discuss about this but in a bigger picture. A typical global picture of control network is depicted in Figure 1 below.

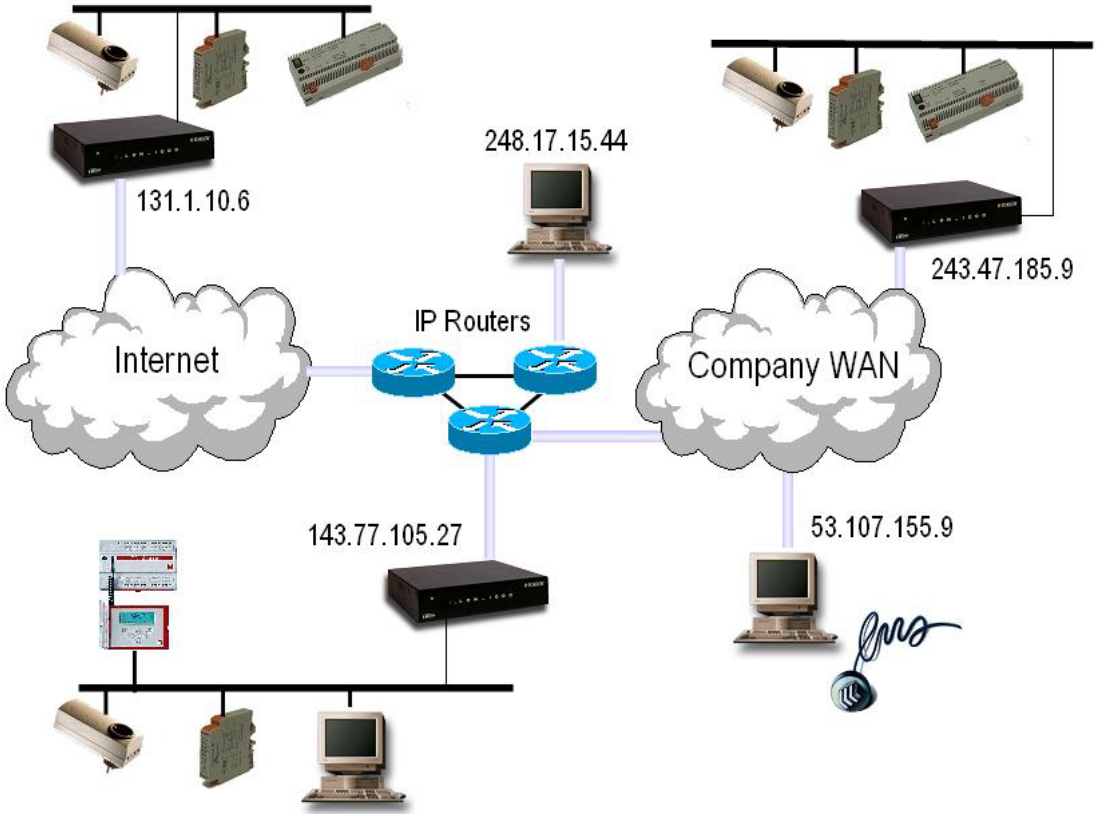

Fig.1. Control network elements – A global view

Typically, on top of those local elements including *nodes (sensors, controllers, actuators), channel(s), transceiver(s),* control networks may also include *repeater, bridge, switch, router, hubs, and gateway.* The following will explain these terms.

Repeater

Repeaters operate on OSI layer 1 and physically isolate network segments s shown in Fig.2. Repeater is always necessary if the line transmission loss getting too high or if the *fan out* of the transceiver bus driver is beyond its limit by the amount of connected nodes. Repeater does not check for valid packets, it just takes care of a signal shaping between sender and receiver side.

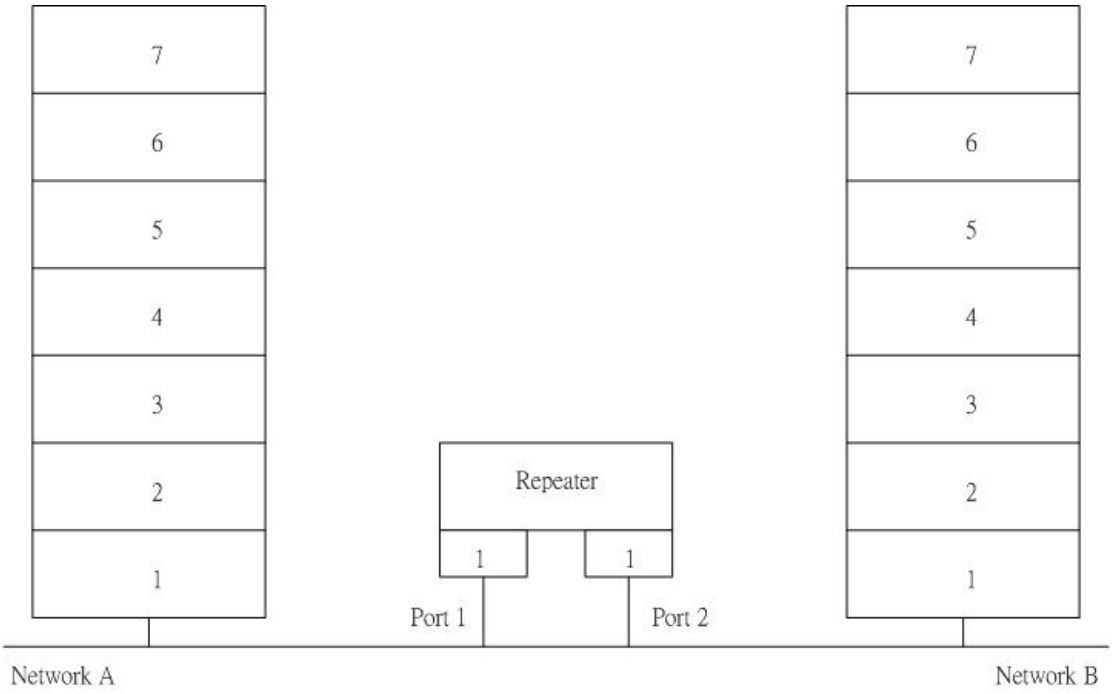

Fig.2 Repeater and OSI-layers

Bridges

Bridges as shown in Fig.3 operate on OSI layer 2 and perform in generally a decoupling of network segments for the purpose of

- load sharing.
- error limiting.
- take care of security aspects.
- connect network segments using different layer 2 protocols w.r.t. their media access procedure.
- Perform repeater functions like separating network segments physically.

Each node in the network has its own unique hardware address (MAC-address). In order to forward packets from one side of a bridge to the other side, bridges hold forwarding tables that have forwarding flags assigned for the MAC-address of the destination nodes. Since the forwarding tables need to hold forwarding flags for all destination nodes on each side of the bridge, the size of these tables is directly proportional to the amount of nodes in the logical network.

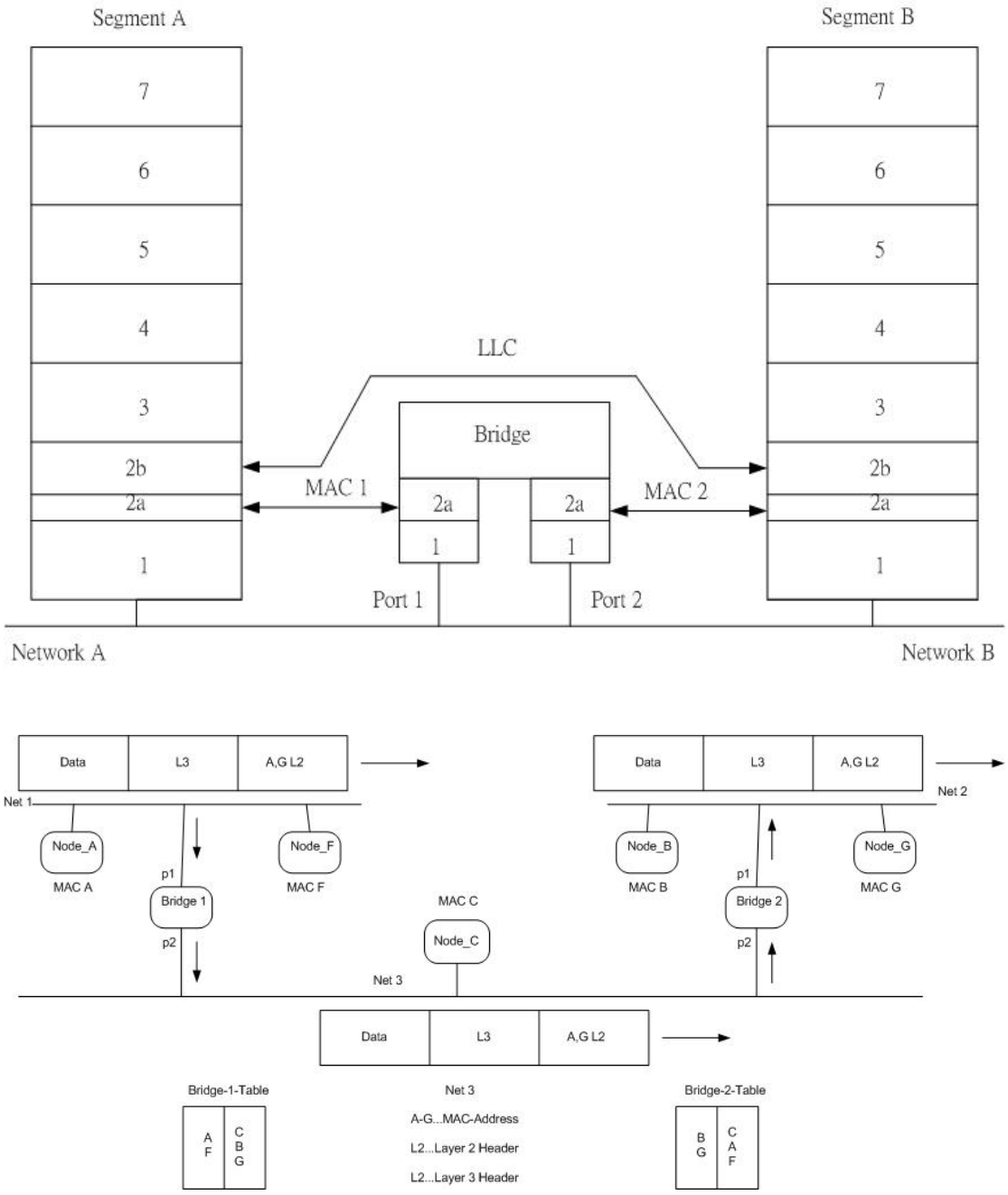

Fig.3 Bridges using MAC-address with forwarding tables

Router

Router use addresses and protocols based on layer 3 to perform forwarding of packets as shown in Fig. 4.

Layer 3 addresses are unlike MAC-addresses logical addresses and thus hardware independent. Router support structure a network into subnets and to assign nodes to a subnet. In order to use a router in a network, layer 3 addresses need to be assigned in addition to the MAC-address on each node. A router connects adjacent subnets and knows the shortest path to other subnets.

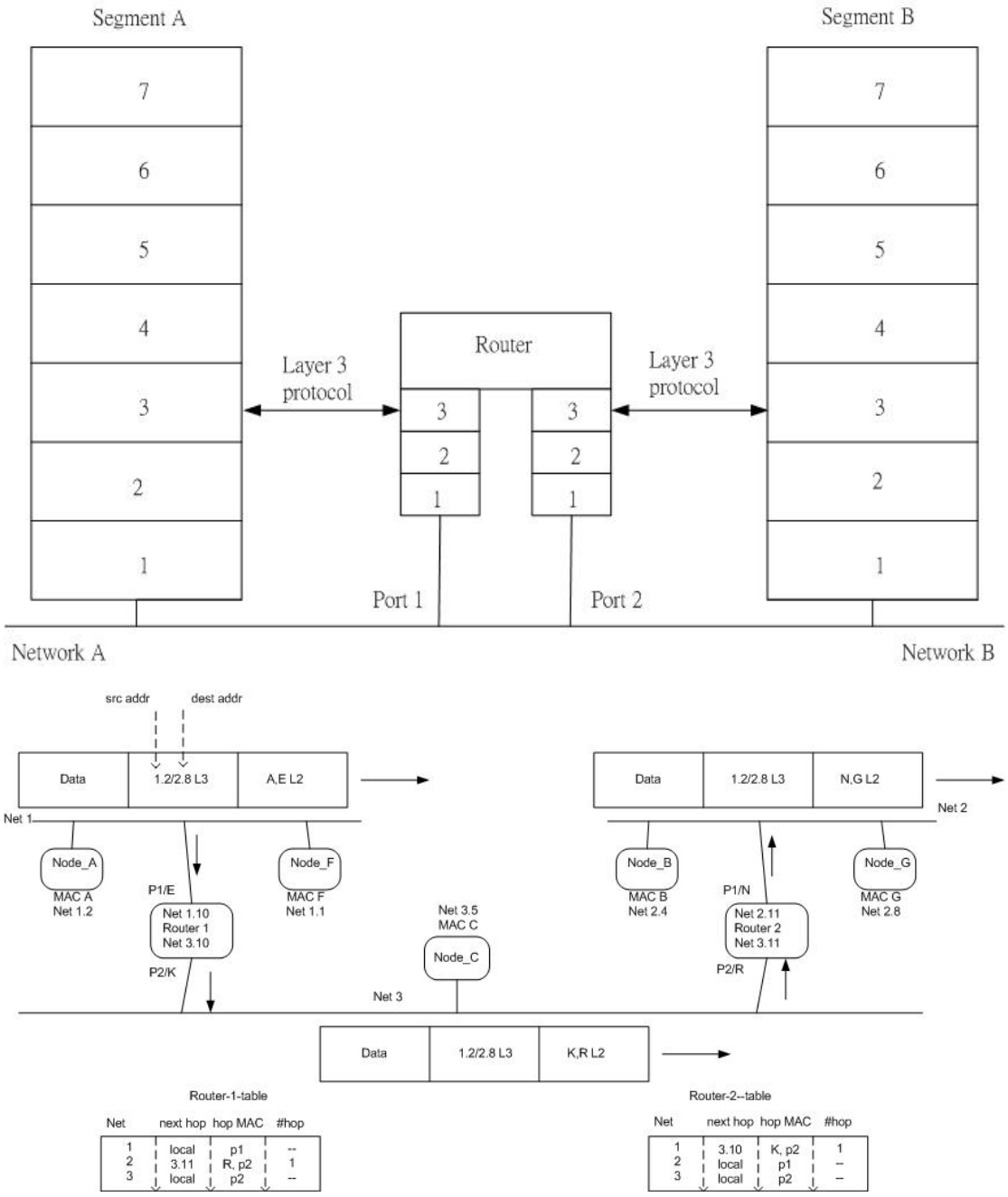

Fig.4 Routers with routing tables

In contrast to bridges, routers manage only forwarding tables for layer 3 addresses containing forwarding flags for each noticed subnet. In that case, the forwarding tables size is proportional to the number of subnets and not the number of nodes. Data packets are addressed by the transmitting node to the router's MAC-address, containing the layer 3 address of the destination node. The router processes only packets addressed to the router. In case that the router receives a valid packet, it withdraws the contained layer 3 address and look it up in its forwarding table to decide what to do with this packet. If there is a forwarding-bit set for the destination subnet, the router forwards the packet, otherwise the packet will be discarded. Routers

need to have the following knowledge about:

- the MAC-address of all nodes within the subnet where also one router port belongs.
- the routes to several destination nodes by knowing the necessary routers.

The requirements for routers are:

- The transport system needs to have layer 3 functionality (logical layer 3 addresses, protocols support routing, etc..) implemented between two end systems.
- End systems need to be informed about the location of their belonging router.
- End systems need to change their layer 3 addresses if they change their locations (subnets).
- Routers need to exchange information about the network topology by routing protocols in order to keep their routing tables consistent.

One example router, which can be configured as repeater, bridge or router is a router sold by Echelon called Lonworks router, this router is a 2-port router, as shown in Fig.5. Each port of a Lonworks router is using a Neuron chip.

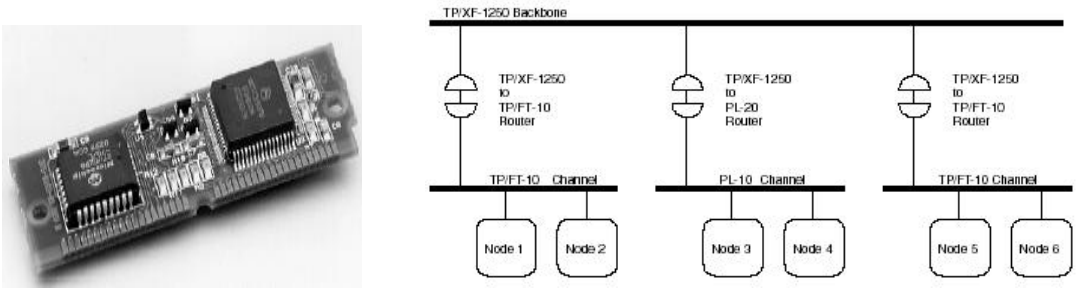

Fig.5 Lonworks router with example application

Gateway

In general, gateways operate on top of OSI layer 7 and serve to connect different communication systems. If the communication systems to be connected provide a similar application layer interface, the expense of the application in the gateway can often be hold low. On the other hand, if communication systems need to be connected providing very different application layer interfaces (for example, if there are no similar services provided), the expense in the gateway's application can be very high. Interconnecting protocols that do not have similar characteristics result in an application intensive gateway, called *application gateway*. According to the enhanced efforts in the application, the reaction time of such a solution increases inevitably. In order to achieve short reaction time, the hardware efforts increase accordingly

because of the need of a fast application CPU to handle the gateway application. The essential disadvantage of an application gateway is the creation of the gateway software itself. For every project to be realized, a specific application needs to be created or at least needs to be adapted. Thus interconnecting different protocols always result in higher system costs.

Hubs & Switches

So, how do we build up a bigger network with many nodes? When we studied network topology, we came across the star, and the bus topologies, if we combine them, a star-bus topology is created as shown in Fig.6.

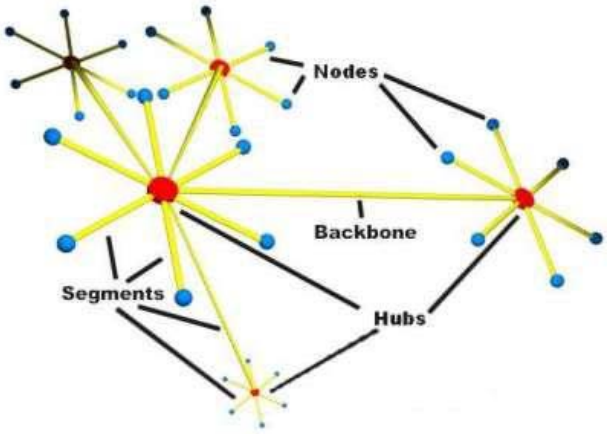

Fig.6 Star-bus topology

Star-bus is probably the most common network topology in use today, it combines elements of the star and bus topologies to create a versatile network environment. Nodes in particular areas are connected to hubs (creating stars), and the hubs are connected together along the network backbone (like a bus network). Quite often, stars are nested within stars.

The hub takes the signal from each node and sends it to all of the other nodes connected. Hubs come in several sizes, noted by the number of ports available -- a four-port hub can connect four nodes, an 8-port hub can connect up to eight nodes and so on. Most hubs are stackable. A stackable hub has a special port that can connect it to another hub to increase the capacity of the network. So if you start with a four-port hub but eventually have five nodes, you can buy another four-port hub and connect it to the one you already have, increasing the potential number of nodes on your network. A cable/DSL router usually has a four-port Ethernet hub built in.

In the most basic type of network found today, nodes are simply connected together using hubs. As a network grows, there are some potential problems with this configuration, including:

- **Scalability** - In a hub network, limited shared bandwidth makes it difficult to

accommodate significant growth without sacrificing performance. Applications today need more bandwidth than ever before. Quite often, the entire network must be redesigned periodically to accommodate growth.

- **Latency** - This is the amount of time that it takes a packet to get to its destination. Since each node in a hub-based network has to wait for an opportunity to transmit in order to avoid collisions, the latency can increase significantly as you add more nodes. Or, if someone is transmitting a large file across the network, then all of the other nodes have to wait for an opportunity to send their own packets. You have probably seen this before at work -- you try to access a server or the Internet and suddenly everything slows down to a crawl. *In control network, this form of data transfer is not suitable for alarming systems due to the extended latency which may cause the dropping of data packets.*
- **Network failure** - In a typical network, one device on a hub can cause problems for other devices attached to the hub due to incorrect speed settings (100 Mbps on a 10-Mbps hub) or excessive broadcasts. Switches can be configured to limit broadcast levels.
- **Collisions** - Ethernet uses a process called CSMA/CD (Carrier Sense Multiple Access with Collision Detection) to communicate across the network. Under CSMA/CD, a node will not send out a packet unless the network is clear of traffic. If two nodes send out packets at the same time, a collision occurs and the packets are lost. Then both nodes wait a random amount of time and retransmit the packets. Any part of the network where there is a possibility that packets from two or more nodes will interfere with each other is considered to be part of the same collision domain. A network with a large number of nodes on the same segment will often have a lot of collisions and therefore a large collision domain.

While hubs provide an easy way to scale up and shorten the distance that the packets must travel to get from one node to another, they do not break up the actual network into discrete segments. That is where switches come in.

Switches

Think of a hub as a four-way intersection where everyone has to stop. If more than one car reaches the intersection at the same time, they have to wait for their turn to proceed (refer to such analogy as shown in Fig.7). Now imagine what this would be like with a dozen or even a hundred roads intersecting at a single point. The amount of waiting and the potential for a collision increases significantly. But wouldn't it be amazing if you could take an exit ramp from any one of those roads to the road of your choosing? That is exactly what a switch does for network traffic. A switch is like a cloverleaf intersection -- each car can take an exit ramp to get to its destination

without having to stop and wait for other traffic to go by (refer to such analogy as shown in Fig.8).

Fig.7 Four-way intersection

Fig.8 Cloverleaf intersection

A vital difference between a hub and a switch is that all the nodes connected to a hub share the bandwidth among themselves, while a device connected to a switch port has the full bandwidth all to itself. For example, if 10 nodes are communicating using a hub on a 10-Mbps network, then each node may only get a portion of the 10 Mbps if other nodes on the hub want to communicate as well. But with a switch, each node could possibly communicate at the full 10 Mbps. Think about our road analogy. If all of the traffic is coming to a common intersection, then each car it has to share that intersection with every other car. But a cloverleaf allows all of the traffic to continue at full speed from one road to the next.

In a fully switched network, switches replace all the hubs of an Ethernet network with a dedicated segment for every node. These segments connect to a switch, which

supports multiple dedicated segments (sometimes in the hundreds). Since the only devices on each segment are the switch and the node, the switch picks up every transmission before it reaches another node. The switch then forwards the frame over the appropriate segment. Since any segment contains only a single node, the frame only reaches the intended recipient. This allows many conversations to occur simultaneously on a switched network.

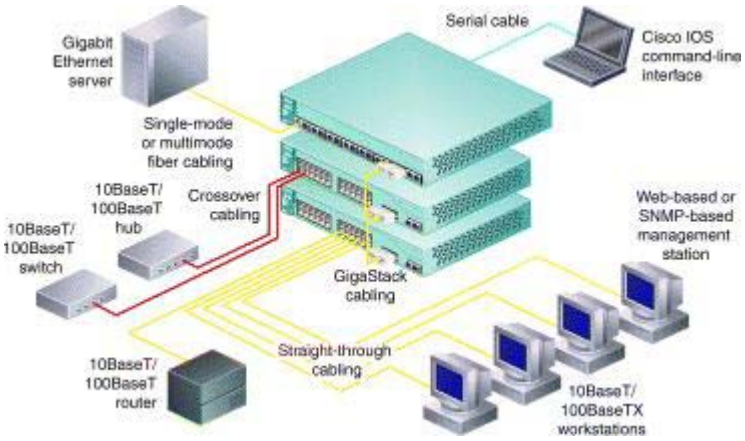

Fig.9 Example of Network using Switch (Image courtesy Cisco Networks)

Switching allows a network to maintain full-duplex Ethernet. Before switching, Ethernet was half-duplex, which means that data could be transmitted in only one direction at a time. In a fully switched network, each node communicates only with the switch, not directly with other nodes. Information can travel from node to switch and from switch to node simultaneously.

Fully switched networks employ either twisted-pair or fiber-optic cabling, both of which use separate conductors for sending and receiving data. In this type of environment, Ethernet nodes can forgo the collision detection process and transmit at will, since they are the only potential devices that can access the medium. In other words, traffic flowing in each direction has a lane to itself.

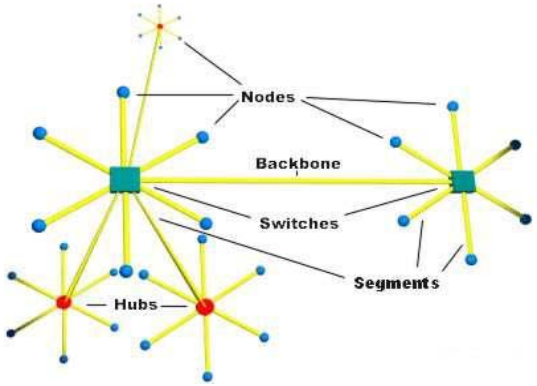

Fig. 10 Network mixed with switches and Hubs

This allows nodes to transmit to the switch as the switch transmits to them -- it's a collision-free environment. Transmitting in both directions can effectively double the

apparent speed of the network when two nodes are exchanging information. If the speed of the network is 10 Mbps, then each node can transmit simultaneously at 10 Mbps.

Most networks are not fully switched because of the costs incurred in replacing all of the hubs with switches. Instead, a combination of switches and hubs are used to create an efficient yet cost-effective network. For example, as shown in Fig. 10, a company may have hubs connecting the computers in each department and then a switch connecting all of the department-level hubs.

An application example using Lonworks switch in a control network is shown in Fig.11. With its 5 network ports it can directly connect 4 FT-10 channels and one port connect to a high-speed backbone channel. In larger networks this high-speed TP-1250 backbone channel can be used to connect multiple switches. The communication between the different network segments happens completely transparent to the nodes on the different segments and the operator of the network.

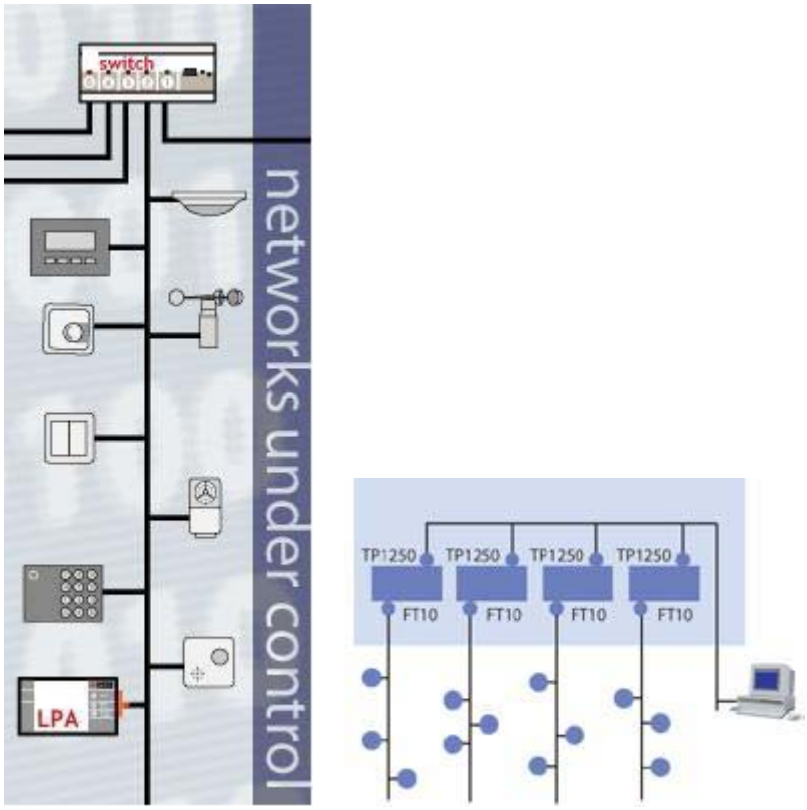

Fig.11 Control network switch

You see that a switch has the potential to radically change the way nodes communicate with each other. But you may be wondering what makes it different from a router? Switches usually work at Layer 2 (Datalink) of the OSI Reference Model, using MAC addresses, while routers work at Layer 3 (Network) with Layer 3 addresses (e.g. IP, IPX or Appletalk, depending on which Layer 3 protocols are being

used). The algorithms that switches use to decide how to forward packets is different from the algorithms used by routers to forward packets.
One of these differences in the algorithms between switches and routers is how broadcasts are handled. On any network, the concept of a broadcast packet is vital to the operability of a network. Whenever a device needs to send out information but doesn't know who it should send it to, it sends out a broadcast. For example, every time a new computer or other device comes on to the network, it sends out a broadcast packet to announce its presence. The other nodes (such as a domain server) can add the computer to their browser list (kind of like an address directory) and communicate directly with that computer from that point on. Broadcasts are used any time a device needs to make an announcement to the rest of the network or is unsure of who the recipient of the information should be.
A hub or a switch will pass along any broadcast packets they receive to all the other segments in the broadcast domain, but a router will not. Think about our four-way intersection again: All of the traffic passed through the intersection no matter where it was going. Now imagine that this intersection is at an international border. To pass through the intersection, you must provide the border guard with the specific address that you are going to. If you don't have a specific destination, then the guard will not let you pass. A router works like this. Without the specific address of another device, it will not let the data packet through. This is a good thing for keeping networks separate from each other, but not so good when you want to talk between different parts of the same network. This is where switches come in.

2. Control Network System Requirement

In order to understand and design a better control network, we have to identify the network requirement. These requirements are one way or the other connected and they have impacts on the protocol, network design and products.
Network requirements in general can be broken down as follows:

- Scalability & flexibility
- Reliability
- Architecture
- Performance
- Network Management
- Interoperability

Scalability & flexibility

The size of a network according to their applications is usually designed to be scalable. A building or a factory may need expansions over its life. From a single

sensor/actuator to a building-wide control network, the number of nodes increase from several to more than ten of thousands. As the sensors/actuators become more complex, not only the number of devices but also the distance and topology need to be modified. The size of the network dictates the networks address space. The packet data field sizes for various types of sensors, actuators are different depends on the applications. A network often contains different type of devices that carry different application messages, it may need to have some type of segmentation for a better efficiency and throughput (switches and routers usually implemented for this purpose).

Special considerations for a control network:

1. How many nodes in total can be supported or what is the address space?
2. Does it support logical segmentations?
3. Which network topology is the best for a scalable network?
4. How does it easily extend the distance?
5. How does it easily achieve the network scalability?
6. How does it easily add or remove devices?

The network protocol often has limitations of the total number of nodes on the network and the number of nodes on each channel or each subsystem. A decent protocol needs to support sufficient and extensible address space, easy distance extension and easy wiring (one example is *free topology*). The expansion of a network is mainly through the segmentation.

Take Lonworks technology as an example, the Lonworks control network is supported by LonTalk protocol, Neuron chip and Lonworks transceivers, Lonworks technology provides a scalable free topology network using the free topology transceiver. It provides a flat architecture that supports the address requirements of the entire network but allows logical segmentation. It support domain, subnet, allowing the logical segmentation through network level routers. Here is the summary of boundaries:

Number of domains: 2^{48}
Number of subnets per domain: 255
Number of nodes per subnet: 127
i.e. a total of 32,385 nodes per domain, and a total of 32K x 2^{48} devices in system.

Lonworks technology supports both network variable (NV) and explicit message for network communication. The NV supports an average of 31 bytes of data packet and the max. explicit message size is 228 bytes of data.

Adding and removing devices are achieved by using a Lonworks Network Services (LNS) based network management tools.

Reliability

The network reliability mainly focuses on the following issues:

1. Reliable message transfer and control loop closure.
2. fault tolerance, fault isolation and recovery.

Due to the vast applications, a network protocol needs to support a number of message services. For example, unacknowledged, acknowledged, request/response messages. For each type of message service, it has its mechanism to support the reliability. The reliability also needs to be guaranteed at the physical level. For example, the protocol needs to implement a CRC checking.

Using Lonworks technology again, it provides the following services to enhance the protocol services:

1. It supports unacknowledged message service with repeat.
2. It supports unicast and multicast message services, with acknowledgement from each addressed node.
3. Request/response service to ensure that loop has been closed
4. 16-bit CRC checking for error detection
5. sender authentication to ensure authorized messages

Fault tolerance can be achieved through redundancy by using duplicated nodes, duplicated connections/paths, or duplicated networks. Loops are sometimes used to allow communications despite a break at one location. They prevent accidents ranging from a loosening wire in the sprinkler system to an accidental snip by electrician servicing a building. The implementation of loops need to have considerable sophistication in transceiver design as well as the network architecture and protocol support.

Architecture

The network architecture is also an important topic that has a strong bearing on the other aspect of network features.
Many historical models show multiple levels featuring mainframes, large and small minicomputers sprinkled with a reserve of personal computers at the lowest level. The difference between this vision and today's reality is too obvious. Instead, real computer network today comprises of a flat network distributed client-server system, connected with routers, bridges, switches, and gateways. Many clients and servers have equal computing power, for example, both use Windows XP operating system.

Decentralized computing put computing power in places otherwise would not have been available. This migration is inevitable due to the benefits that it provides to the end users. A distributed peer-to-peer architecture uses much less bandwidth, minimizes the network traffic, eliminates the host bottleneck and increases the network reliability. Distributed control is likewise extending the reach of control systems. In reality, the peer-to-peer fully distributed intelligent control networks are installed and operational in a commercial buildings, homes, and other industrial applications such as gas tank farms, wastewater treatment plants, metals processing systems, automotive manufacturing floors, etc.
A distributed peer-to-peer network works better than a traditional hierarchical network as long as:

1. the address space allows it
2. there are provisions for logical portioning of the network through addressing and traffic filtering
3. The computing power, communication speed and data field sizes for each node are scalable to handle a variety of control tasks.

Lonworks network that we use as an example, is a highly distributed and intelligent control network. Each intelligent sensors or actuators are made of

1. Neuron chip (or another uP/uC implements with LonTalk protocol)
2. Transceiver
3. supporting circuitry that connects Neuron chip or equivalent with I/O devices

The intelligent sensors and actuators can be connected logically by using a network management tool. The actuators and sensors are then communicated with each other to share control information. It is because of the distributed intelligent, the network reliability and throughput are improved.

Performance

A network's performance depends on the various aspect of the network design. The following is a list of these considerations.

1. Network architecture: A peer-to-peer network architecture allows communication directly across the control loop. This eliminates the central controller as a bottleneck. The centralized controller is merely an artifact of how computer power was made historically. With the increase of computing power over the low cost VLSIs, the centralized architecture becomes completely unnecessary. A peer-to-peer network requires the sensors actuators having enough intelligence to directly execute the control algorithm and take suitable actions, instead of obtaining commands from a central controller.

2. Maximum packet size: The maximum packet size determines the number of packets and time to complete a transaction. In general, all systems, for example lighting systems, or discrete manufacturing systems with a large proportion of binary sensors – contain analog/complex sensors and actuators, with data field requirements ranging from 4-25 bytes for standard data, and considerably more for configuration and calibration data. Furthermore, as these devices become smarter, these numbers will climb. It is estimated that roughly 30 bytes. This requirement may climb to 50 bytes and higher over the next several years. The average size of Lonworks NV is 31 bytes, while for explicit message is 228 bytes max. Packet size directly affect the network latency, which become critical for alarming services.

Note about throughput & IP using Lonworks technology: *When LonWorks was initially introduced, the 835 packets max. per second throughput of a TP-1250 channel (bus topology) seemed adequate for any foreseeable project. In the mean time, control systems have become larger and more complicated. Network management tools and monitoring, scheduling and alarming applications are taking a bigger and bigger slice of the network capacity. It only takes six FT-10 channels (168 packets max. per second throughput) running at full capacity to max out a TP-1250 channel. The result is that without careful planning, even moderate-sized LonWorks installations will often exceed the TP-1250 channel capacity. System Integrators have been forced to partition control networks based on infrastructure limitations. The result is a longer (and more expensive) planning cycle, and a final design that is more difficult to manage and maintain than it should be. The flexibility of the FT-10 channel characteristics makes it an ideal solution for retrofitting control networks into existing building. Support for both bus and free topology, long maximum cable length specification, and ability to stub into the channel at any point, makes the installation process straight forward. However, the installation of a TP-1250 backbone into an existing building can present significant challenges. Careful planning is required to ensure that repeaters and routers are installed in locations that meet the length and bandwidth specifications. New wiring must be laid to facilitate the backbone. For multi-building installations, the cost on new wiring quickly makes retrofit proposals cost prohibitive. If the required locations for repeaters does not correspond well to the building layout, allowances must also be made for new power and control panel installations. IP has excellent characteristics for use as a control network backbone. It's fast, it can reach anywhere, and IP infrastructure components are cheap. By tunneling ANSI 709.1 on top of an IP infrastructure, the benefits of IP are realized without having to sacrifice the functionality and robustness of LonTalk. What is the maximum distance of a*

LonTalk/IP channel? Only as far as the Internet reaches. While latency issues must be carefully considered when designing a control network that spans an IP network, distance and it's associated wiring costs no longer need to be the limiting factor. Management, monitoring and alarming applications can now be centralized for multi-site organizations. New value propositions for lowering management costs and improving consistency and efficiency in control networks can be presented to customers. With an estimated throughput of over 50,000 packets per second, a dedicated 100 Mbit LonTalk/IP channel increases the LonTalk backbone capacity by at least sixty times. With this increased bandwidth capacity, System Integrators can design systems that are less complicated and easier to manage, without impacting the stability or capabilities of the system.

3. Delays through routers and gateways: If the network layer of a protocol does not contain provisions for network level routing, routers connecting sub-networks have to be implemented at the application layer. Such an implementation has impact on the application processing power, application code space and ability.

4. Monitoring control-event driven update: Sensors need to be polled or scanned at a fixed time interval. Loops need to be close as fast as they can. Such a fixed interval scanning is sub-optimal. To optimize the network performance, event-driven update is more desirable. This requires an event-driven scheduler at the originating device, peer-to-peer access to the network, and a receiving node that has the communication and computing resources to process information on demand.

Network Management

Network management includes various network functions. Those should include network installation (integration), maintenance, monitoring, controlling and diagnostics. The network integration, maintenance, monitoring and control are important aspects of a control network. Usually, the end users and network integrators pay most attentions to these parts.

Easy access to data is the key to increase efficiency and lower operating costs. This means that users need a network that can accommodate any number of HMIs, SCADA, and data logging stations, plus the ability to exchange data among different control subsystems and crossing the system boundaries if necessary.

End users also need a control network that is easy to expand and reconfigure. Once reconfigured, the system should have enough built-in intelligence to automatically update each of the system-level monitoring stations to reflect the changes. Intelligent devices in the network is preprogrammed (or on line reprogrammable) to know how

to perform its piece of the control puzzle. Motor starters know how to start, how to check for interlocks and how to detect failures. Sensors know how to convert raw sensor readings into linearized readings and how to detect sensor faults. Still, these devices do not know whom to share their data with, nor what their specific setpoints and alarm limits are for any given system.

Instead of developing and loading a complex set of control algorithm into the central controller, the software commissioning task for a network consists of loading each intelligent device with its network configuration – typically an address and a list of devices with which it shares data. It also loads its application configuration, for example, the setpoints, high and low alarm limits and other calibration data.

It is because a network may contain hundreds or thousands of intelligent devices that must be configured, the network must let multiple installers work on the system at the same time, without conflict. Ideally, each tool could have a different user interface, depending on how it's used. By building application intelligence into network toold, much of the commissioning processes can be automated, cutting commissioning time and costs.

Similarly, the network must allow multiple tools for maintenance. Multiple technicians should be able to diagnose problems and repair devices simultaneously, without having to coordinate their actions or even be aware of one another. They should be able to plug portable tools into the network near the suspected fault. The network should also allow fault isolation, and perhaps even repair can be done without sending someone to the site.

In summary, a good network management software should

1. reduce commissioning time and cost
2. great access to data (since the client tools do not need to contain a network database, they can be anything from PC, PDA or simple LCD display).
3. simplified systems integration
4. increase system up time (as multiple maintenance & repair works can be done simultaneously).

Example of Network Management Tool

With more than 400 companies world-wide developing applications for the Lonworks control network, Echelon's Lonworks Network Services (LNS) and Lonworks Component Architecture (LCA) are the most widely used control network operating system. As an expansion of the Lonworks technology, the LNS is a multi-client, multi-server network operating system for control networks as depicted in Fig.12.

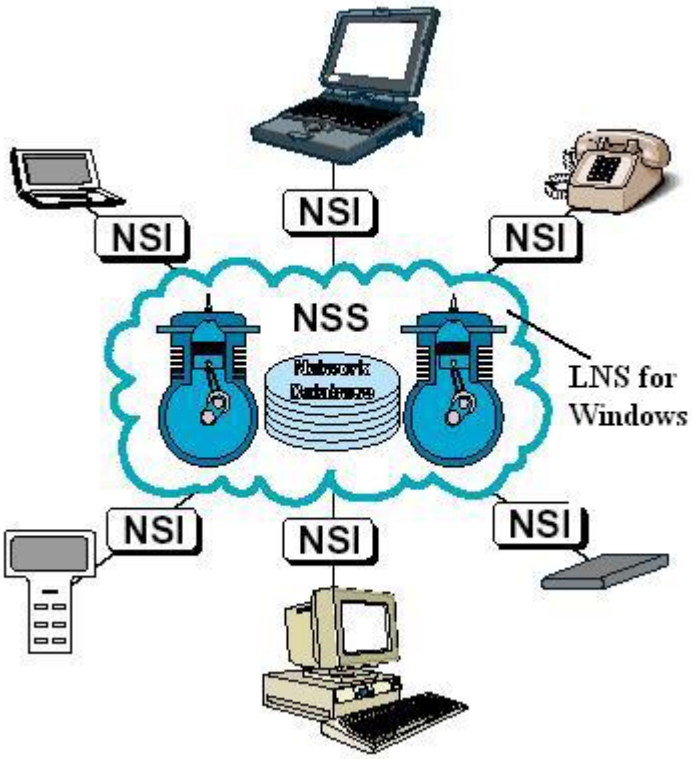

Fig.12 Multiple tools on the same control network by using LNS

LCA performs object linking and embedding for common database access. LNS models the system as a hierarchy of objects. These objects provide services that clients can request, have properties that clients can get and set, and generate events to report changes. Each LNS device contains a hardware component called a Network Service Interface (NSI) that provides physical access to the network and transparent access to the services and resources of servers.

LNS includes a default server-the Network Services Server (NSS) – that provides a core set of network installation, commissioning, maintenance, diagnostics, monitoring, and control services. It also provides directory services and server management functions.

In addition to being a client of NSS services, each LNS application can provide its own set of application-specific services, properties and events. These custom services, properties, and events are registered with the NSS, and can be invoked by any other LNS tool in the system.

LNS/LCA uses ActiveX (OLE) and Microsoft Windows O/S. One example of such all-in-one Lonworks Network Management Tool is ***LonMaker For Windows*** *and its use as a tool in network design, install, commission, maintenance and management is shown in Fig.13, and the process is briefly described below.*

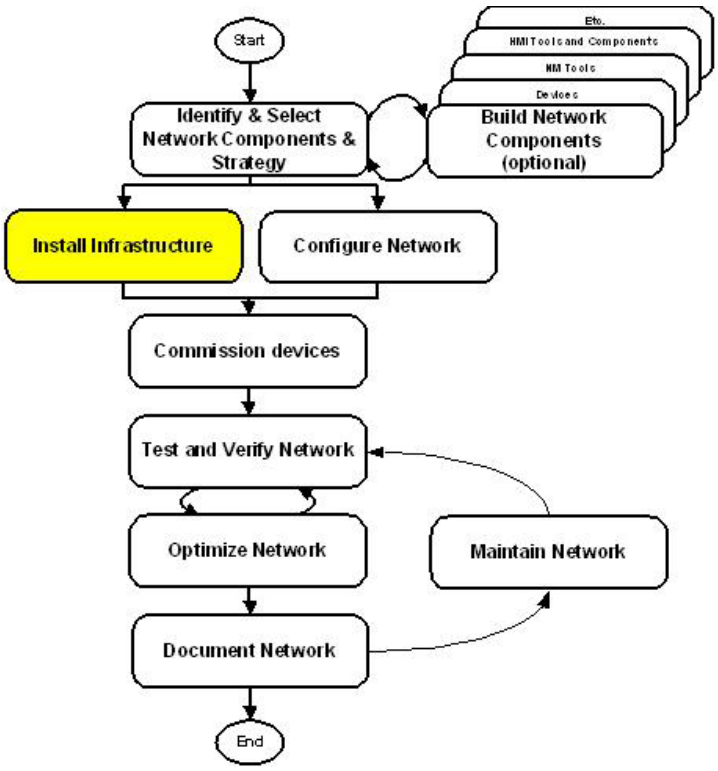

Fig.13 Network installation, commissioning, maintenance and management process

- *Indentifying & selecting components*
  - ◊ *Describing the project*
  - ◊ *Identifying network architecture*
  - ◊ *Identifying installation and maintenance scenarios*
  - ◊ *Identifying components*
- *Installing the physical network*
  - ◊ *Installing cabling*
  - ◊ *Installing infrastructure devices*
  - ◊ *Installing application devices*
- *Programming the network*
  - ◊ *Acquiring the external interface*
  - ◊ *Configuring devices and objects*
  - ◊ *Binding network variables*
- *Applying the program to the physical network (commissioning)*
  - ◊ *Acquiring the Neuron ID*
  - ◊ *Commissioning routers*
  - ◊ *Commissioning devices and propagating the program*
- *Testing & verifying the network*
  - ◊ *Verifying application devices' health*
  - ◊ *Verifying network communications*
  - ◊ *Verifying infrastructure devices*

    - ◊ *Verifying cabling and termination*
- *Optimizing or fine-tuning the network*
    - ◊ *Identifying possible changes*
    - ◊ *Modifying infrastructure*
    - ◊ *Optimizing connection properties*
- *Maintaining the network*
    - ◊ *Replacing application devices*
    - ◊ *Updating application devices*
    - ◊ *Moving the network tool*
    - ◊ *Maintaining the LNS server*
- *Documenting the network*
    - ◊ *Creating an operating manual or user's guide*
    - ◊ *Documenting network physics and network design & program*
    - ◊ *Documenting device health*
    - ◊ *Documenting channel and network health*