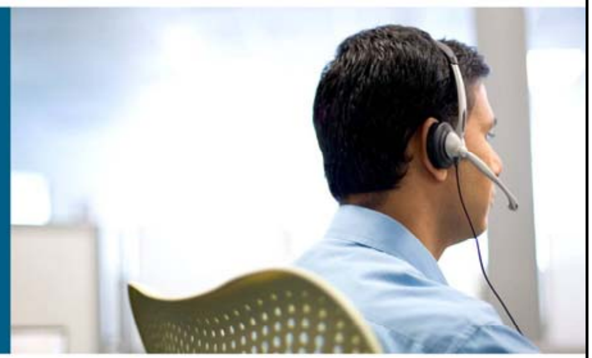
Network Design Language™

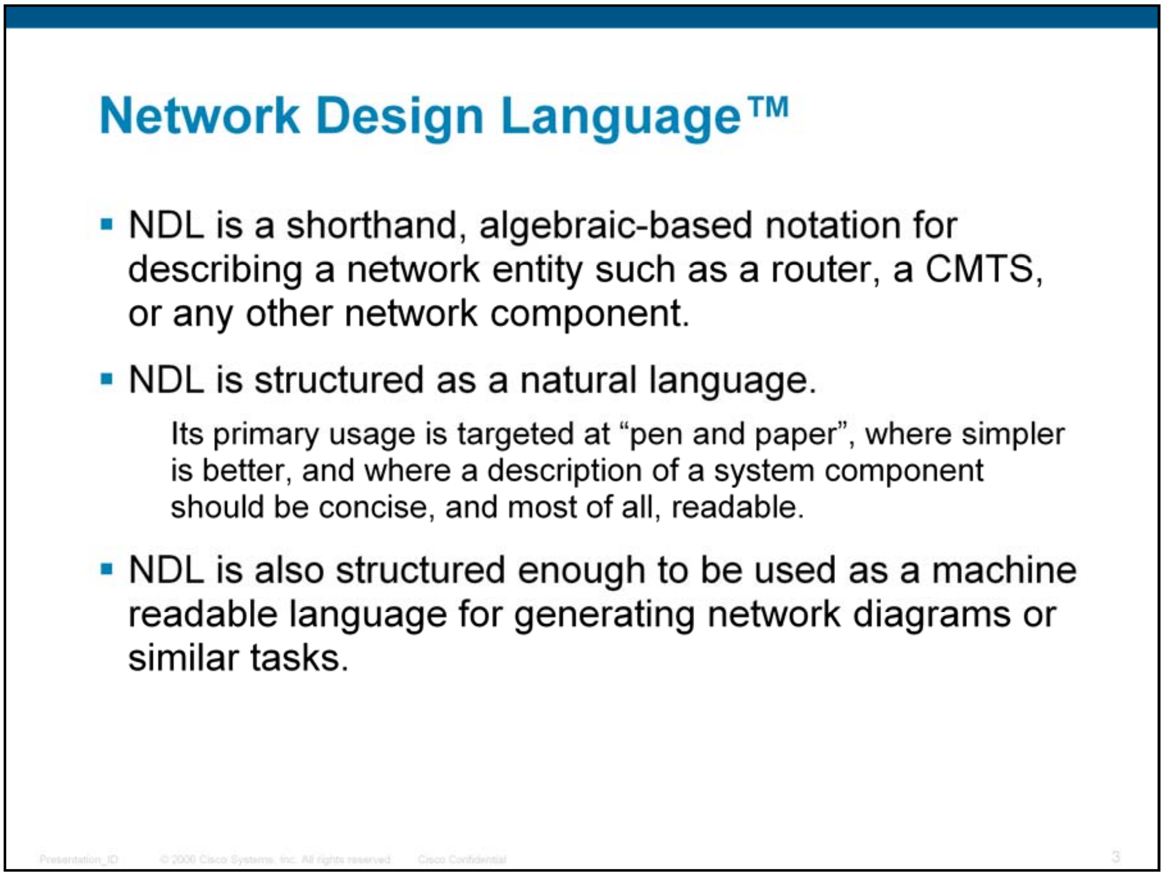
NDL 1.0



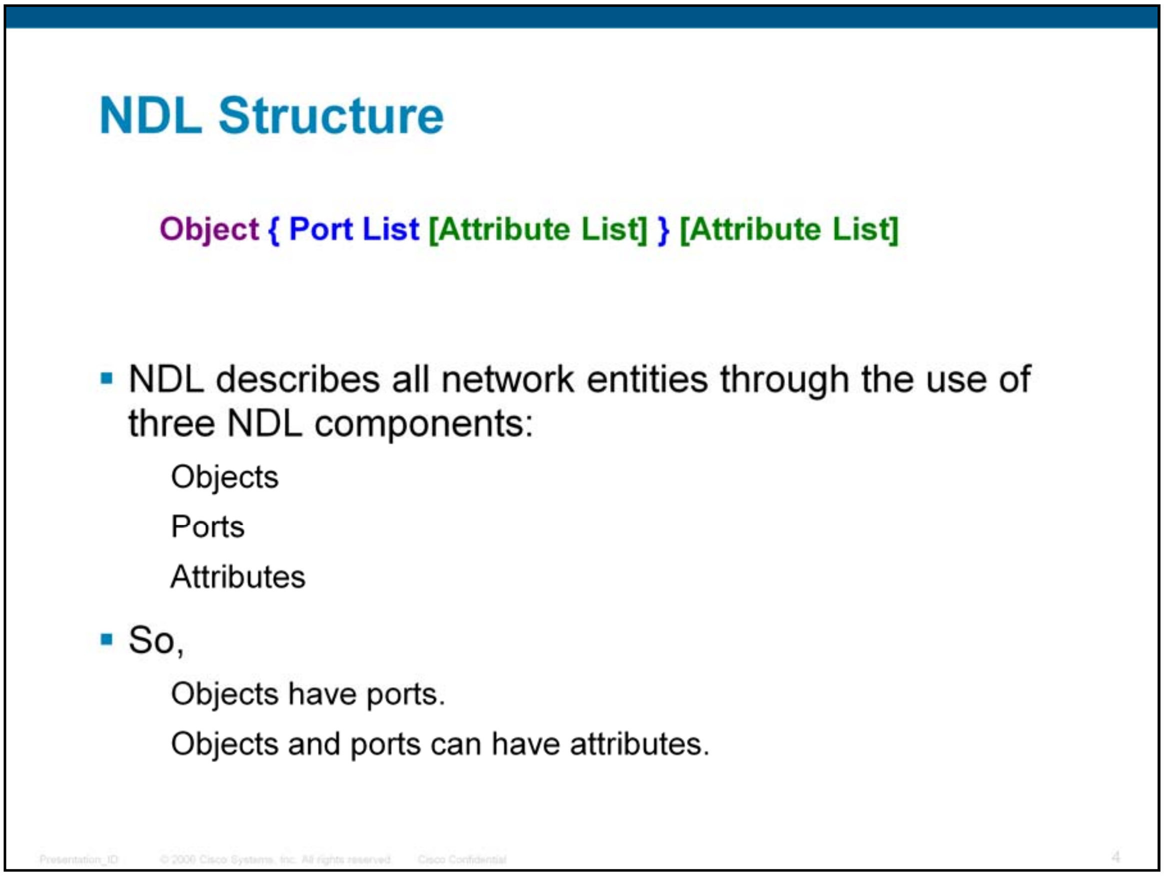
John T. Chapman

Cisco Fellow & CMTS BU Chief Architect


Agenda
Introduction
NDL Syntax
NDL Examples
Conclusion



- NDL is designed to simplify language complexity and provide a simple framework to represent networks and components.
- A representation is devised to decompose complex systems into more tractable subunits, as well as to assemble and map those defined subunits onto actual sentences/formulas that represent the network or component. The intent is to enable anyone involved in any aspect of network design or deployment to easily represent and understand complex systems.
- Threr is an informal and formal version of NDL. The formal version of NDL is intended to be structured enough to be machine readable. Thus, network descriptions could be written in NDL and then software could compile NDL and produce network diagrams.
- This whitepaper describes the basic approach of NDL and explains how certain special cases are handled. It provides case studies as examples, followed by formal language descriptions.



- NDL has its roots in the cable industry. The author developed NDL as he tried to sort out the twisted interconnectivity of Modular CMTSs (M-CMTSs), fiber nodes, and DOCSIS 3.0 service groups, along with asymmetrical downstream (DS) and upstream (US) directions. However, NDL is broadly applicable to any networking scenario.
- NDL assembles larger objects out of smaller objects. NDL can start with various line card definitions, and then add them together to build and describe an entire network. Objects have ports. Attributes can then be assigned to the object, the ports, or both.

Objects

- Objects
  Basic NDL building block. Usually a field replaceable unit (FRU)
  Examples: routers, switches, racks, CMTS, FN, SG, etc.
- NDL assembles larger objects out of smaller objects.
  NDL can start with various line card definitions, and then add them together to build a chassis, rack, and then even describe an entire network.
- For a basic example, a line card (LC) would be:
  LC
- A chassis (CH) with 8 line cards would be:
  CH = 8 * LC

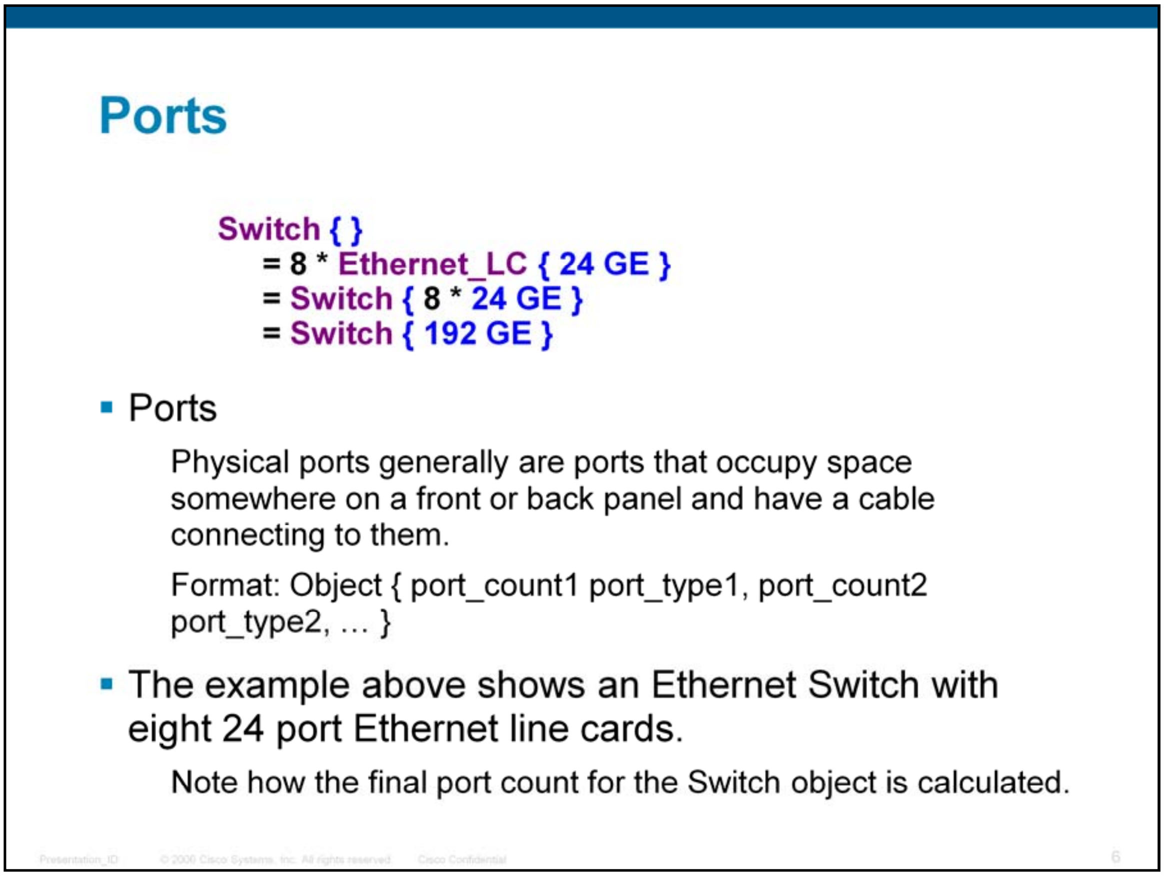
- In NDL, one of the basic building blocks is a line card, also referred to as LC. So, using the informal version of NDL with no port list, a line card would be shown as:

  LC

- Now, let's build a generic chassis that has eight line cards. NDL uses the reserved expression of CH for chassis.

  CH = 8 * LC

- An algebraic expression is used to indicate that eight line cards equal one router.
- This example illustrates how objects are hierarchical. That is, objects can be added together to make bigger objects. As the level of hierarchy continues, the objects become more abstract in nature, but still represent a physical collection of network entities. Examples include:
  - a service group, which is a collection of frequencies on a set of fiber nodes;
  - all the equipment in one rack which may be operating as a unit;
  - all the equipment in an entire network.
- Acronyms:
  - FRU: Field Replaceable Unit
  - CMTS: Cable Modem Termination System
  - FN: Fiber Node
  - SG: Service Group
  - LC: Line Card
  - CH: Chassis



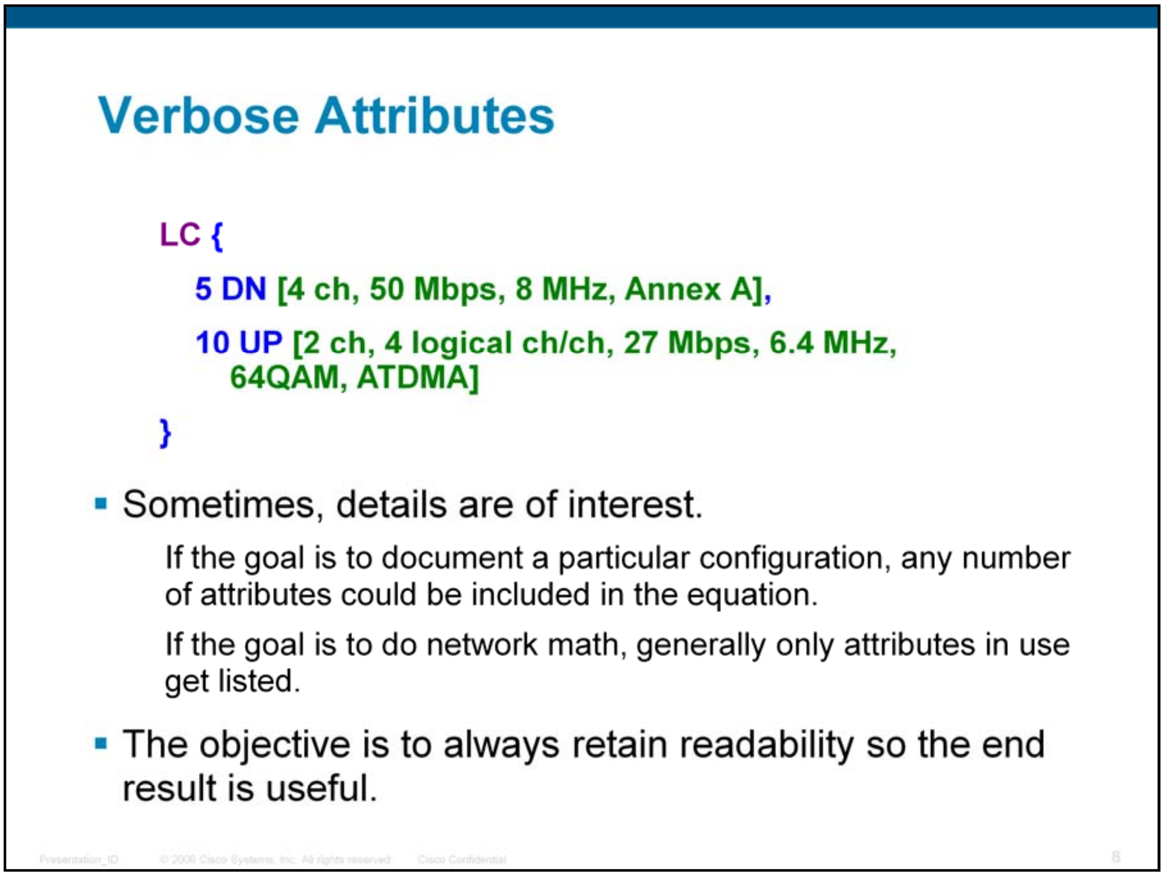
- Port Lists are contained in curly brackets { } immediately following the object name. There are two variables used in describing a port. There is the port type which indicates the function of the port, and the port count which indicates how many of those ports exist for that object.
- The equation starts declaring that a switch is being defined, but the port types and counts are not known. An algebraic equation is used to indicate that there are eight line cards with 24 Gigabit Ethernet ports per line card. The result is a Switch with 192 Gigabit Ethernet ports.
- Also note the formatting used. The original declaration is placed on the first line. The subsequent equal signs are lined up for readability, while multiplications are done in line. The final result is on the last line. This style of format is chosen for readability. Even though everything could fit into one line, it is much more readable if the lines are short and repeated or if like objects are lined up in columns rather than spread out in rows.

Attributes

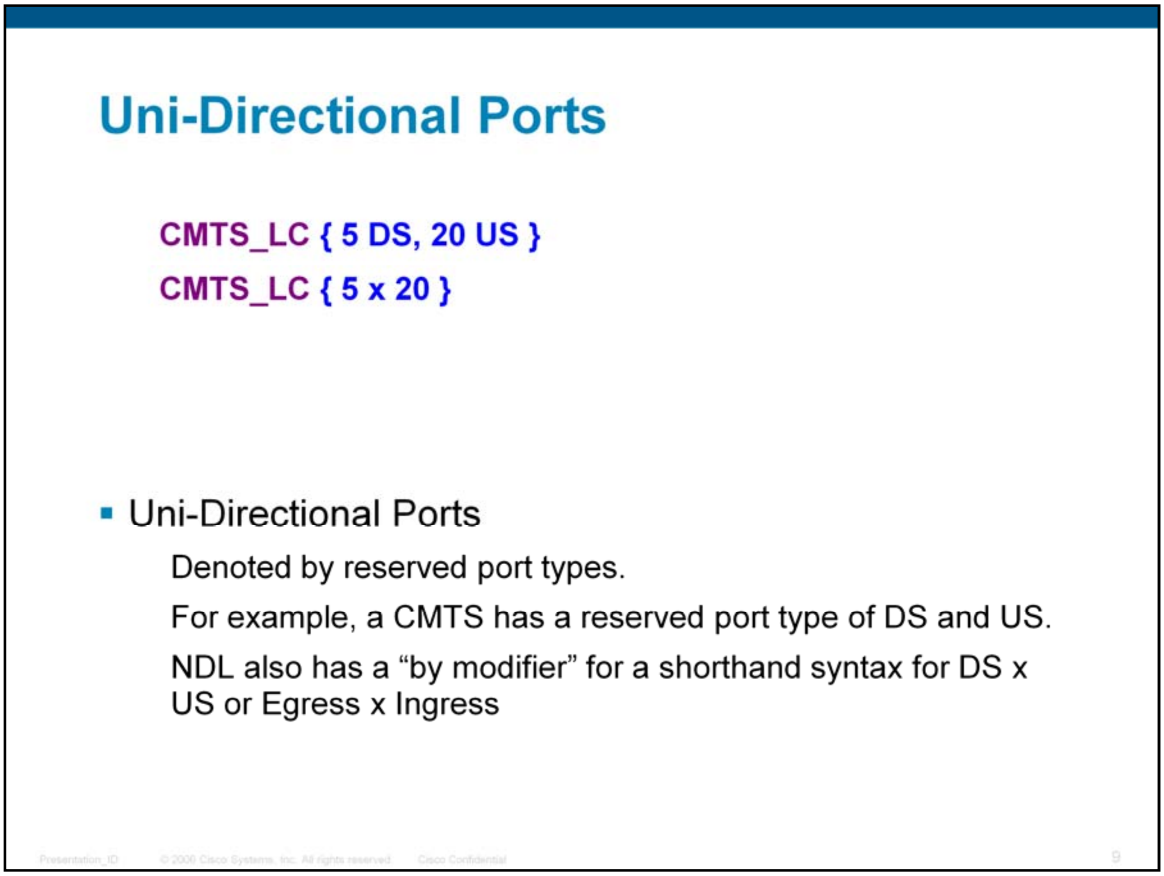
```
LC { 24 GE [1 Gbps] }
   = LC { 24 GE } [24 * 1 Gbps]
   = LC { 24 GE } [24 Gbps]
```

- Each object or port may contain one or more attributes.
  - Attributes are uniquely identified by their units.
  - Informational text attributes also are allowed.
  - Examples: **10 Gbps, 6 MHz, 500 HHP, SFP**
- Attribute lists are contained in square brackets [ ] immediately following a port or an object definition.
- Note the attribute list is moved from the port to the object.

- In the first line of this example, the 1 Gbps attribute is attached to the port type, and, therefore, is interpreted as a per port attribute. In the second line, a mathematical expression is shown in the attribute list. In the third line, the final result is shown. Since the attribute list on the second and third line is after the closing curly bracket }, the attribute list is considered as being attached to the object.
- Any number of attributes could be attached to an equation. As a result, care has to be taken to not add so many attributes that readability is sacrificed. In general, only include attributes that are relevant to the equation. The only time all attributes may be included is in a full prototype of an object.
- Also, obvious attributes should be excluded. In this case, the 1 Gbps attribute is obvious because the port type is GE, which is known to imply 1 Gbps. It is only included because it makes the math operation clearer. The second line is obvious and could be excluded. The attribute list on the third line is actually useful as it shows the total port capacity of the switch.
- There are many ways to have done the formatting. For readability, it works best for the port list to stand out. Thus, the curly brackets for the Port List have one white space immediately inside each one, while the square brackets for the Attribute List have one white space outside them with none immediately inside them. There is always a single white space after a comma. Double spaces should not be used.



- Let's take a CMTS line card that has frequency stacking in both directions, where logical channels exist in the upstream, and where certain channel parameters are of interest.
- Many of these text attributes such as 64 QAM are not normally present since they cannot be used in an equation. However, they can be included if needed. Notice how NDL uses a list format for the Port List to maintain readability.
- The opening curly bracket remains with the object name while the closing curly bracket is on its own line, aligned with the first character of the object's name. The Port List is indented four spaces and has one port per line. The list is still separated by commas.



Uni-directional Ports

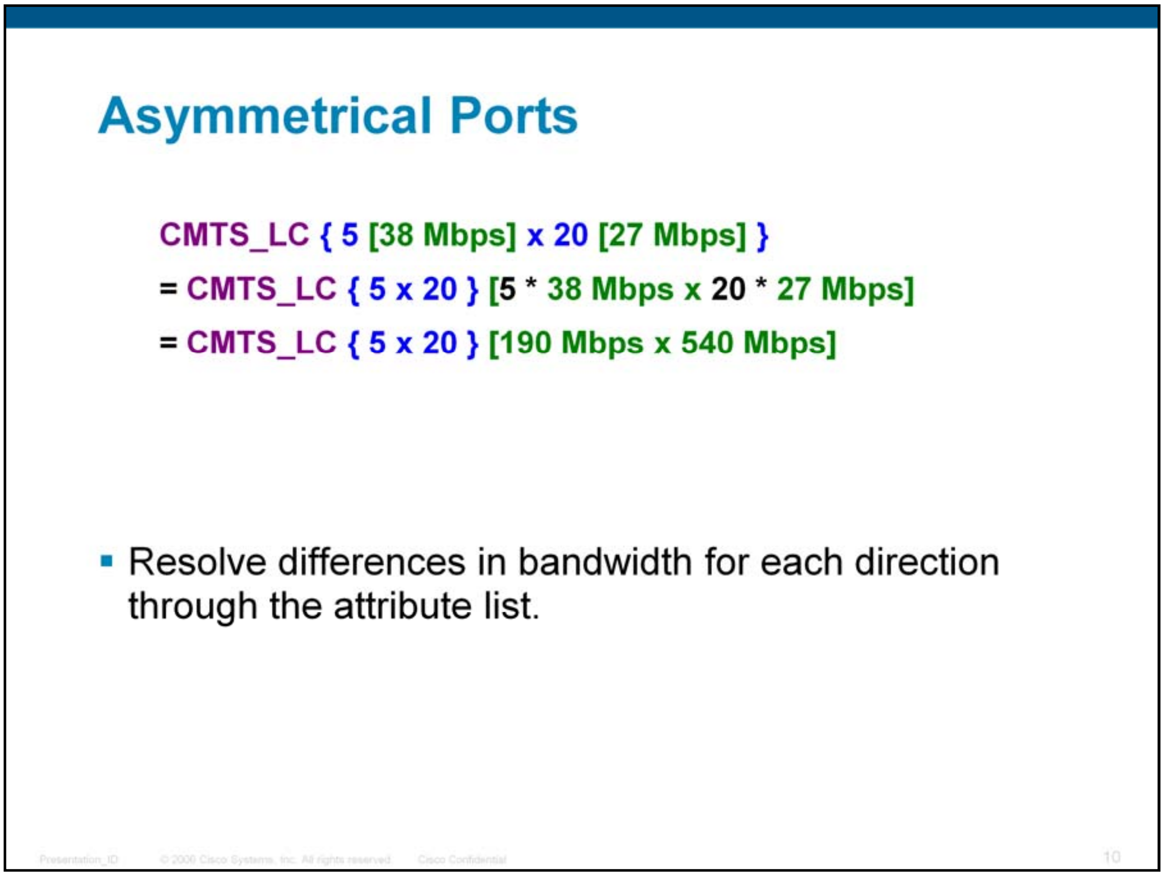
- Some devices have a separate port for each direction of traffic. A CMTS is a classic example, as not only are the ports unidirectional, but they are asymmetrical in both quantity and capacity.

NDL Port Syntax

- NDL manages this with two special port types and a special port operator called the NDL “by modifier”.
- A downstream port has a port type of DS. Downstream is defined as from the cable operator to the subscriber.
- An upstream port has the port type of US. Upstream is defined as from the subscriber to the cable operator’s network.
- For example, a common CMTS line card in the industry is the MC520H which plugs into the Cisco uBR10012 CMTS. It has 5 downstream ports and 20 upstream ports . It would be referred to as:

  MC520H { 5 DS, 20 US }
- NDL has a short-hand notation for this configuration known as the “by modifier”. The operator is denoted by a lowercase “x”. The use of the “by modifier” infers DS x US. The “by modifier“ may be used in the port list or in the attribute list. Thus, another way to refer to this line card is:

  MC520H { 5 x 20 }
- Note that when the “by modifier” is used, the DS and US port types are not used. Also not that the “by modifier” is preceded by one white space and followed by one white space.



Asymmetrical Ports

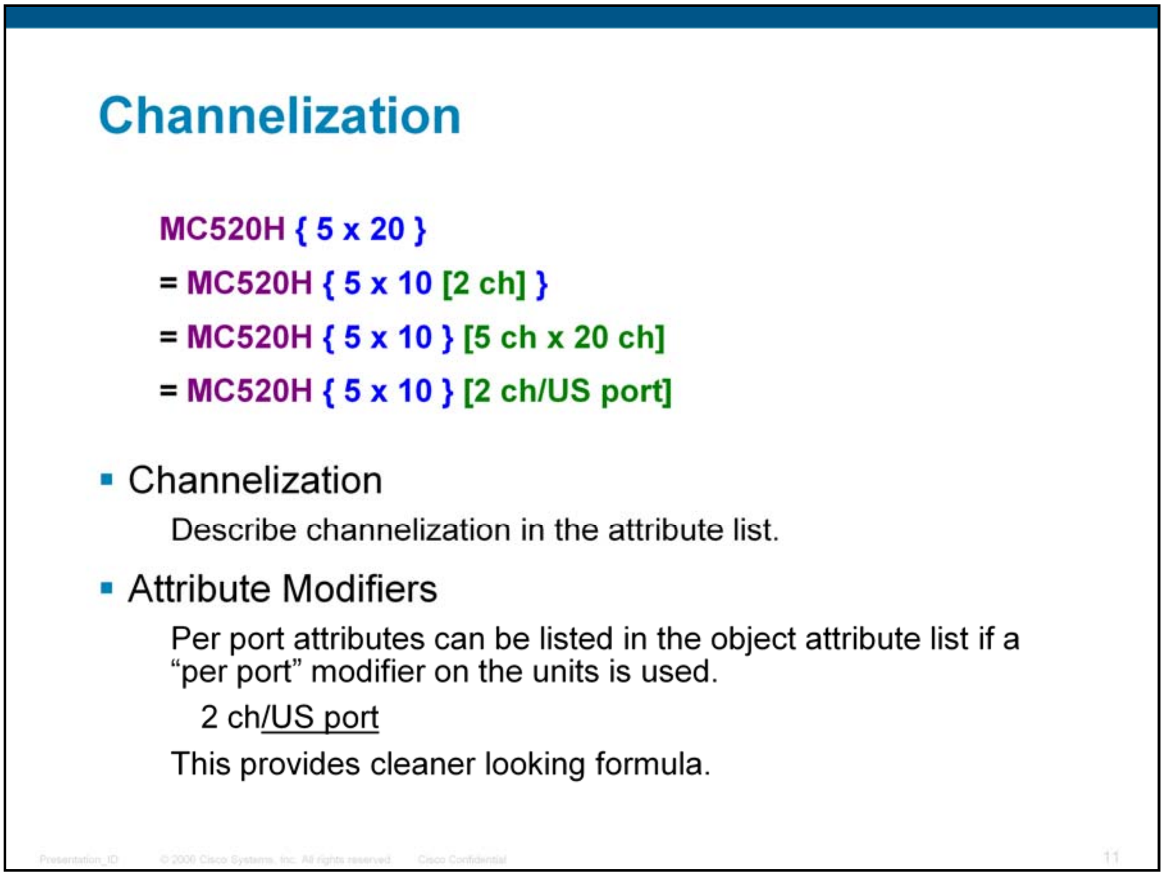
- Most network ports such as Ethernet are bi-directional and support the same data rate in either direction. Sometimes, a Wide Area Network interface such as the serial interface V.35 will have a different egress data rate than an ingress data rate. This can be handled through the attribute list.

  Serial_LC { 4 V.35 [1 Mbps x 256 kbps] }

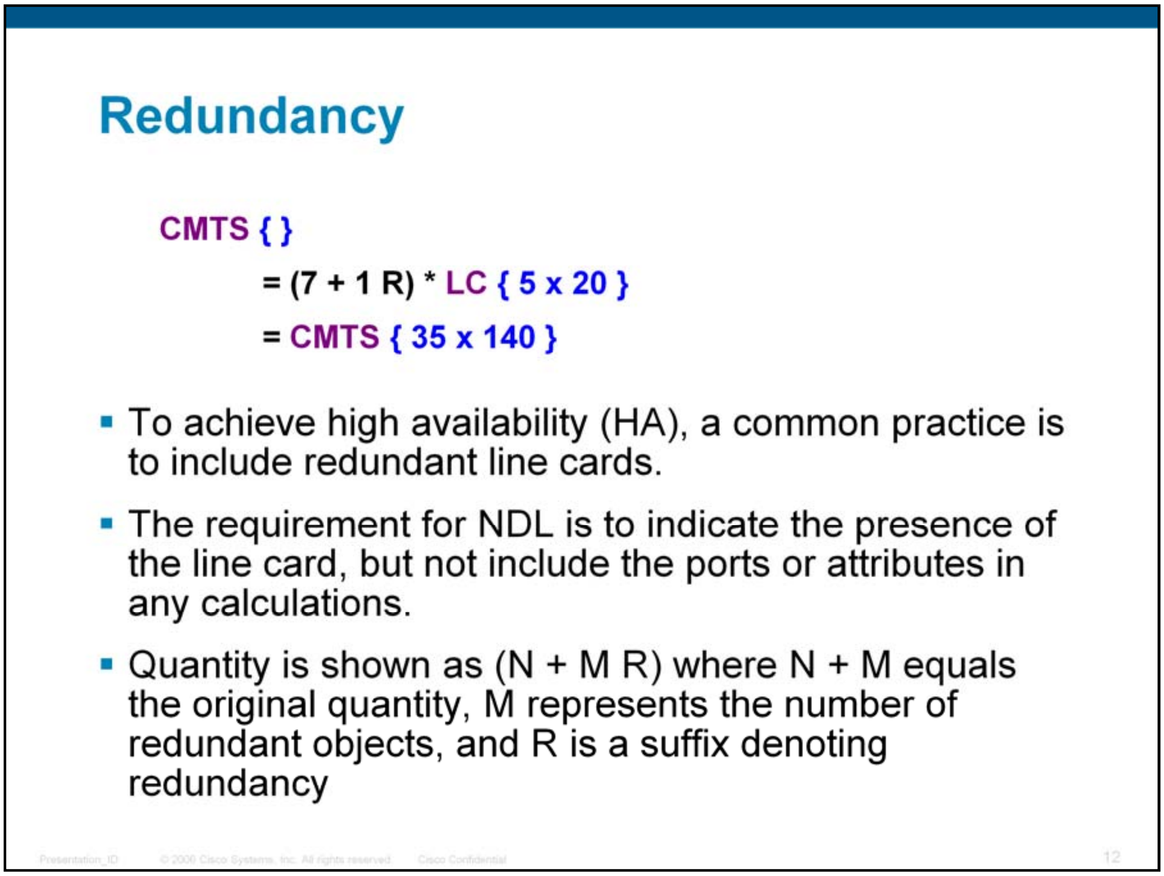
  Serial_LC { 4 V.35 } [1 Mbps/port x 256 kbps/port]

  Serial_LC { 4 V.35 } [4 Mbps x 1 Mbps]

- These examples show different ways of describing the same assembly which has four satellite ports, each of which is 1 Mbps egress and 256 kbps ingress.



- Ports may also be channelized. To continue with the CMTS example, in a CMTS, channelization is achieved by stacking multiple carrier frequencies on the same port. Channelization can take place in either DS or US directions. DOCSIS also defines logical channels. A logical channel is a further level of sub-channelization within an upstream channel.
- NDL treats channels as attributes, not as ports. Channels are attributes of ports. The attribute unit for channels is "ch". When there is only one channel per port, the attribute should not be used unless it is to provide clarity in a calculation.
- The MC520H has an operational mode where it can frequency stack two upstream channels into one port. In that mode, only ten upstream ports are available.
- The first line shows five downstream ports and 20 upstream ports. The second line shows 10 upstream ports at two channels per port. The third line contains the number of upstream ports and upstream channels per line card. In the fourth line, since there is no channelization of the DS, it is not mentioned when the "by modifier" is not used.
- Usually having the attribute list buried on the port list reduces readability. NDL provides a syntax for extracting port level attributes out to the object list by using a per port nomenclature. If there is only one port type, then "/port" is sufficient. Otherwise, the usage is "/port_type port". During calculations, attribute lists may be pushed into the port list, and then extracted out in the final expression. This is a recommended practice as it keeps the Port List simple and readable.



- The line card in use is often called the “working” card. The line card on standby is called the “protect” card. Redundancy groups are either 1+1 or N+1.
- The requirement for NDL is to indicate the presence of the line card, but to not include the ports or attributes in any calculations.
- Let’s start with a non-redundant system. It is a CMTS with eight line cards.

  CMTS { }

  = 8 * LC { 5 x 20 }

  = CMTS { 40 x 160 }

- Now, to make the line cards redundant, the 8 line cards now become a 7+1 group with seven working cards and one protect card. The quantity of the protect card(s) contains a “R” suffix to indicated redundancy and that the count should not be included in the calculations.

  CMTS { }

  = (7 + 1 R) * LC { 5 x 20 }

  = CMTS { 35 x 140 }

- The equation shows the intent of the system, and the result shows the correct number of ports.

Example 1: I-CMTS

CMTS { }

= 8 * Cable_LC { 5 x 20 }

= CMTS { 40 x 160 }

- In an Integrated CMTS (I-CMTS) system, all RF ports are contained within the CMTS.
- Say a basic I-CMTS system consisted of 8 cable line cards. That can be expressed in NDL as shown.

Example 2: Fully Loaded Cisco CMTS.

uBR10012_CMTS { }

= 8 * MC520H { 5 x 20 }

+ 2 * DTI_LC { 2 DTI }

+ 2 * PRE { Serial }

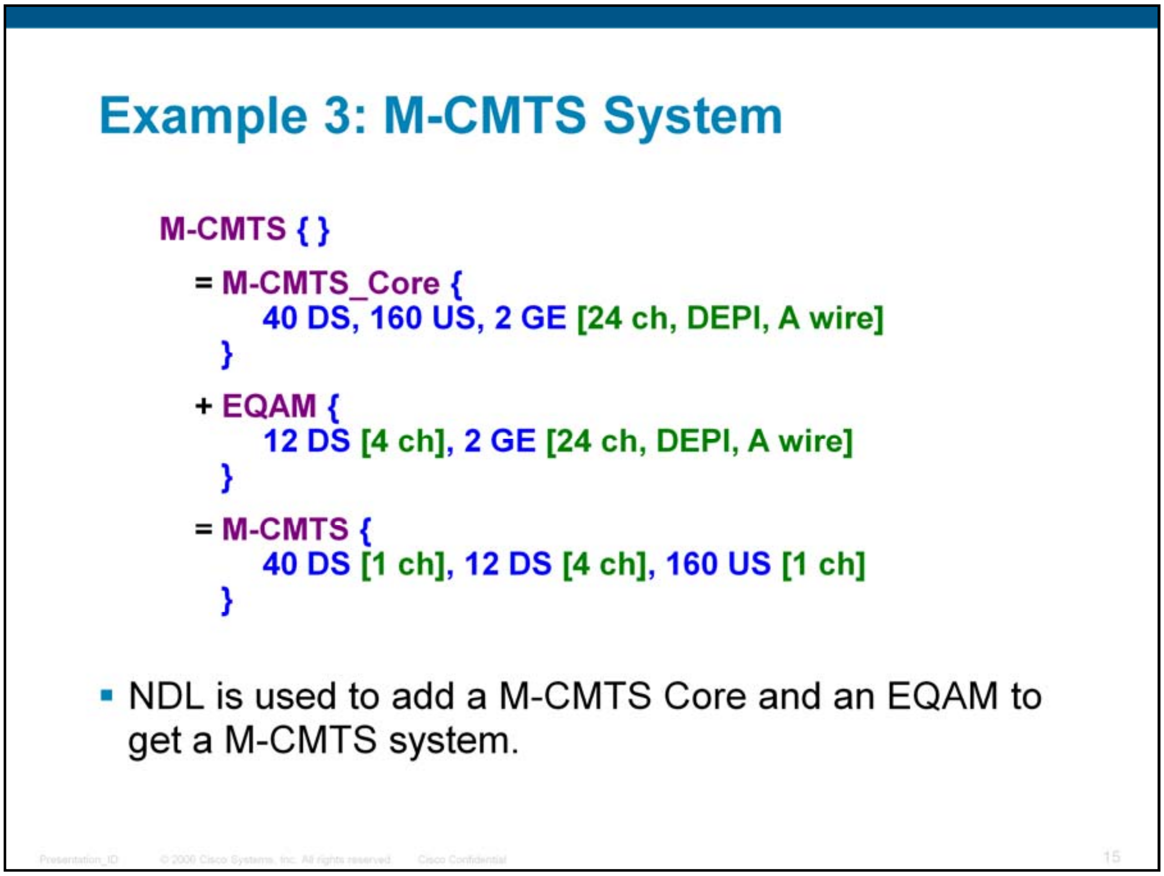
+ 1 * SPA_Carrier { 2 SPA_I/F }

+ 2 * DEPI_SPA { SPA_I/F, GE [24 ch, DEPI] }

+ 2 * GE_LC { 1 GE } [WAN]

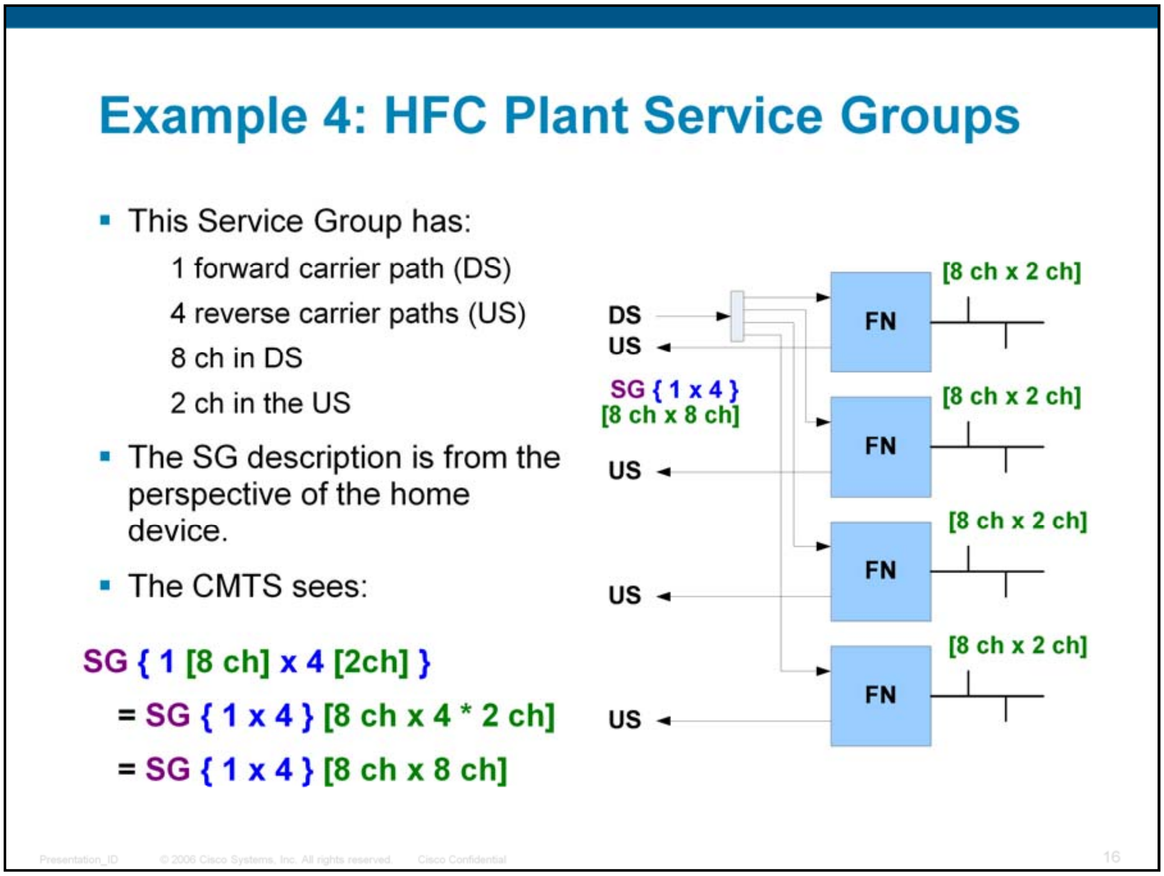
= uBR10012_CMTS {
40 x 160, 2 DTI, 2 GE [DEPI], 2 GE [WAN], Serial
}

- An entire set of line cards is reduced to one port list.



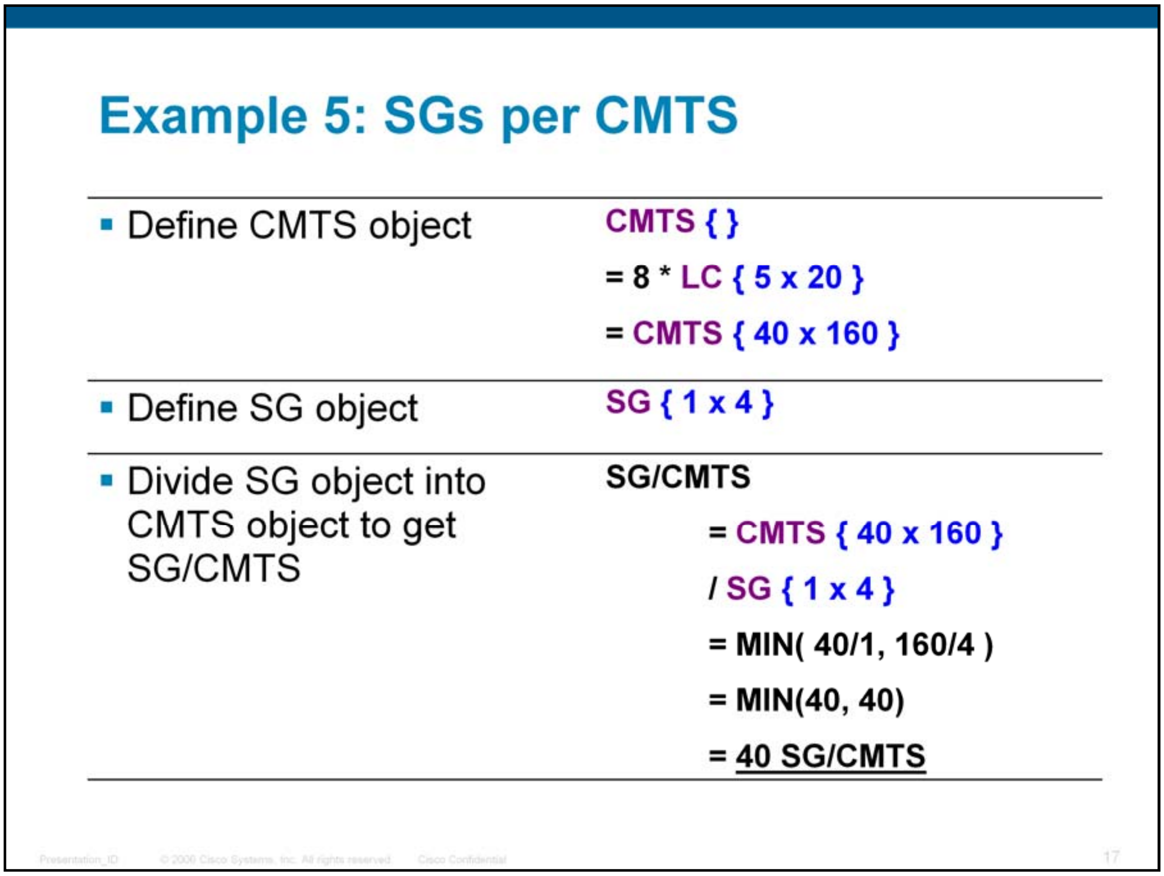
Modular CMTS

- This example shows the power of NDL to definitely, yet tersely describe two separate systems and then combine them into one larger system.
- A Modular CMTS System consists of a Modular CMTS Core plus an EQAM.
- The first line describes the ports of the M-CMTS Core. The second line describes the ports of an EQAM. The third line describes the resulting M-CMTS system.
- Note that there are two types of DS RF ports in the final system. There are:
  - One channel DS RF ports on the M-CMTS Core
  - Four channel DS RF ports on the EQAM.
- They are listed differently by their attribute in the port list.



Service Groups

- A portion of the HFC Plant is defined as a Service Group (SG), and is connected to a CMTS or an EQAM. A SG is one or more downstream frequencies and zero or more upstream frequencies that are to be used on one or more Fiber Nodes (FN). A Fiber Node is an HFC Plant transmission element that interfaces between the fiber and the coax. The significance of a FN is that is represents a fixed number of Households passed (HHP).



- How many SG will a CMTS support?
- Using the basic CMTS from a previous example (with no redundancy)
- Lets take a simple definition of a SG of 1x4 with only one channel in the DS and US.
- The SG per CMTS is solved by dividing the two objects.
- A minimum function is used as either the DS or US ports could become the limiting factor. Note that an object divided by an object is just a number with units. In this case, it is SG per CMTS. Although the math is simple in this example, NDL provides a completely documented set of calculations.

Summary

- NDL, Network Design Language™, is a shorthand way of describing a small or large system.

  Equations in NDL are self-documenting.

- NDL deals with everything in term of objects, ports, and attributes.
- NDL uses basic spreadsheet math to build bigger objects out of smaller objects.
- The larger objects are defined by their inheritance of attributes from the smaller objects.

- Author Information

John T. Chapman
jchapman@cisco.com
chapmanjohnt@gmail.com
www.johntchapman.com