Protocol Design Verification of Wireless Mobiles

Agilent E5515C

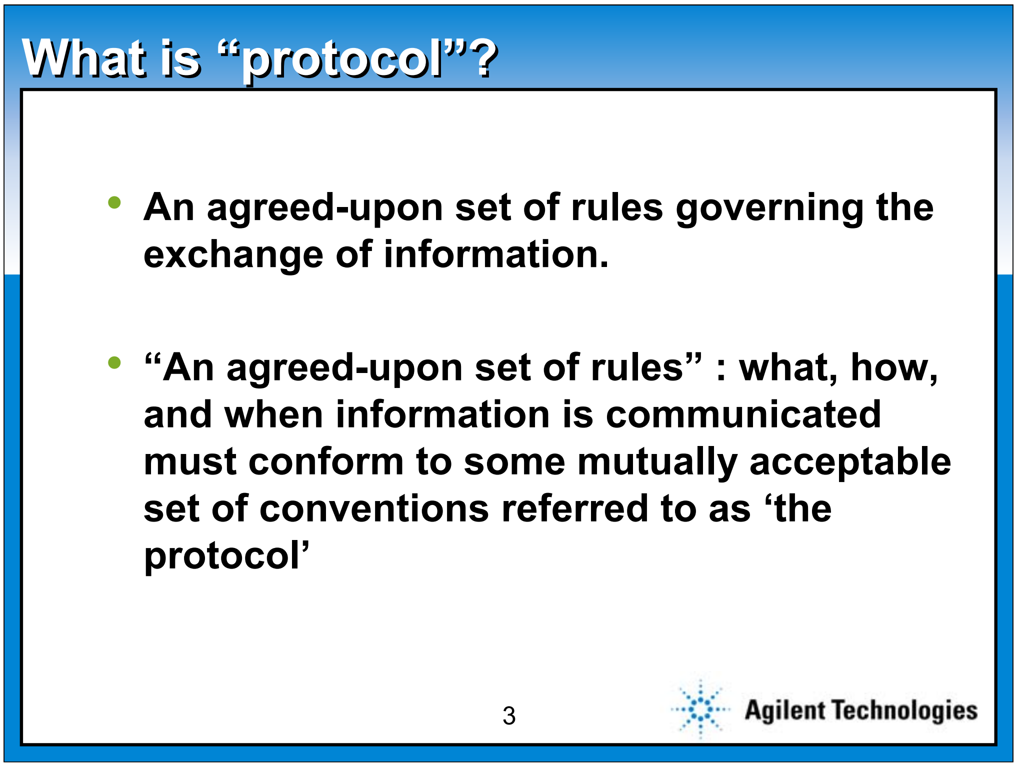
In the E5515C, we have introduced a number of Test Applications, or TA's. Specifically, we cover GSM, GPRS, IS-136, and CDMA. W-CDMA is available using test modes, with call processing to be added soon. The E6701A is the first of our planned Lab Applications, or LA.

The LA is different than the TA in that the prime customer is in an R&D lab. The LA has improvements over the TA in two fronts: enhanced RF capabilities, and Protocol logging and Protocol test.

The focus of this paper is on the Protocol test and Protocol logging portion of the new LA, which focuses on GPRS.

Lets Get Started

- What Wireless Technology is your main interest?
  - A. GPRS
  - B. cdma2000
  - C. W-CDMA
  - D. One of the 2 G Technologies
  - E. None of the Above



What is "protocol"?

- An agreed-upon set of rules governing the exchange of information.
- "An agreed-upon set of rules" : what, how, and when information is communicated must conform to some mutually acceptable set of conventions referred to as 'the protocol'

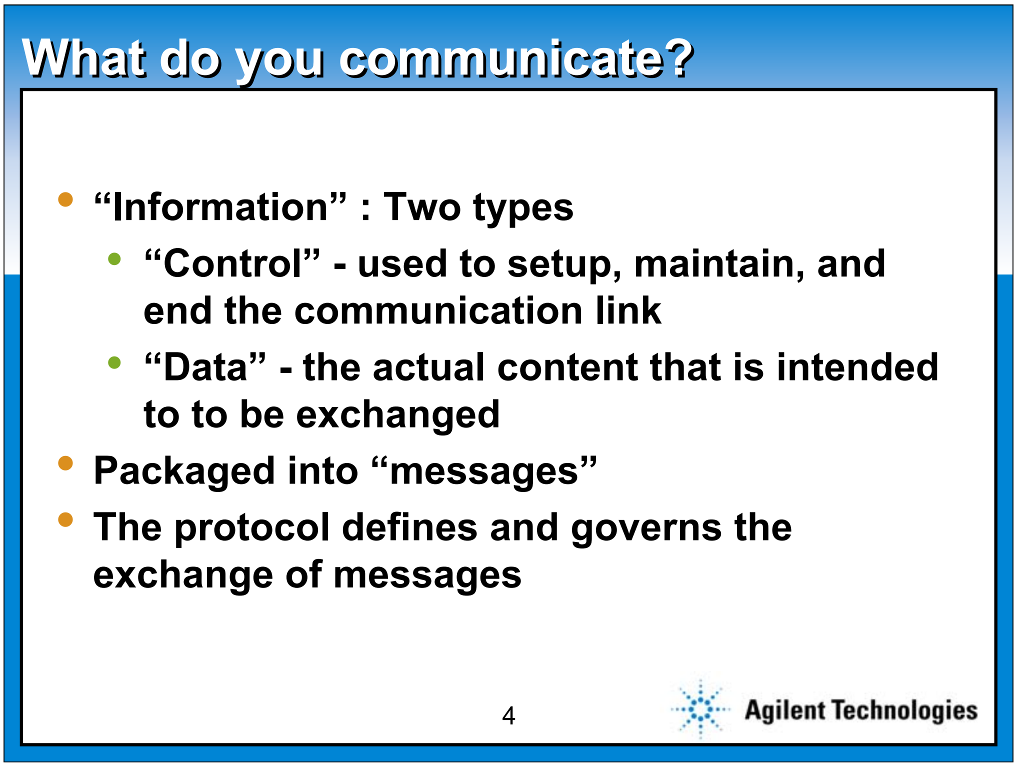
3

Agilent Technologies

So, what is a protocol, anyway? It boils down to a language that has been designed for a specific task. Included in this language are rules about what can be said in specific messages, when those messages can be sent, and the resultant actions by the handset and the network.

Note: I'll use the word, "handset" generically to describe a mobile phone, wireless PDA, PCMCIA modem, or imbedded receiver as long as it is using the GPRS air interface.

These protocols include the modulation, error coding, and transmission of the signals, as well as the messages. As a group, these are defined by a group of individual standards written by either ETSI or 3GPP. Collectively, these work in concert to generate what is usually called the standard, and define the protocol

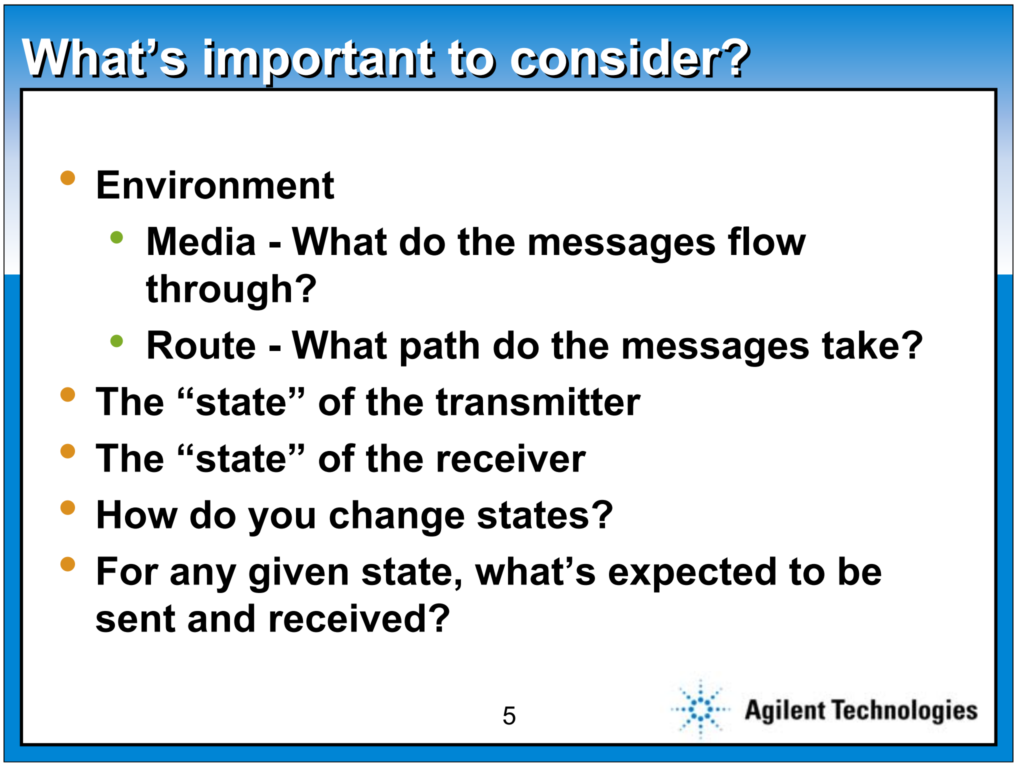


What do you communicate?

- "Information" : Two types
  - "Control" - used to setup, maintain, and end the communication link
  - "Data" - the actual content that is intended to to be exchanged
- Packaged into "messages"
- The protocol defines and governs the exchange of messages

Agilent Technologies

In reality, there are two parallel sets of protocols in use by all wireless systems: Those that control operation of the handset and the base station, and those that control the flow of data from one end to the other. Most of this presentation will focus on the second of these, the data flow. This is particularly important when we consider that GPRS is the first wireless standard to go public in the cell bands with packet access.



What's important to consider?

- Environment
  - Media - What do the messages flow through?
  - Route - What path do the messages take?
- The "state" of the transmitter
- The "state" of the receiver
- How do you change states?
- For any given state, what's expected to be sent and received?

Agilent Technologies

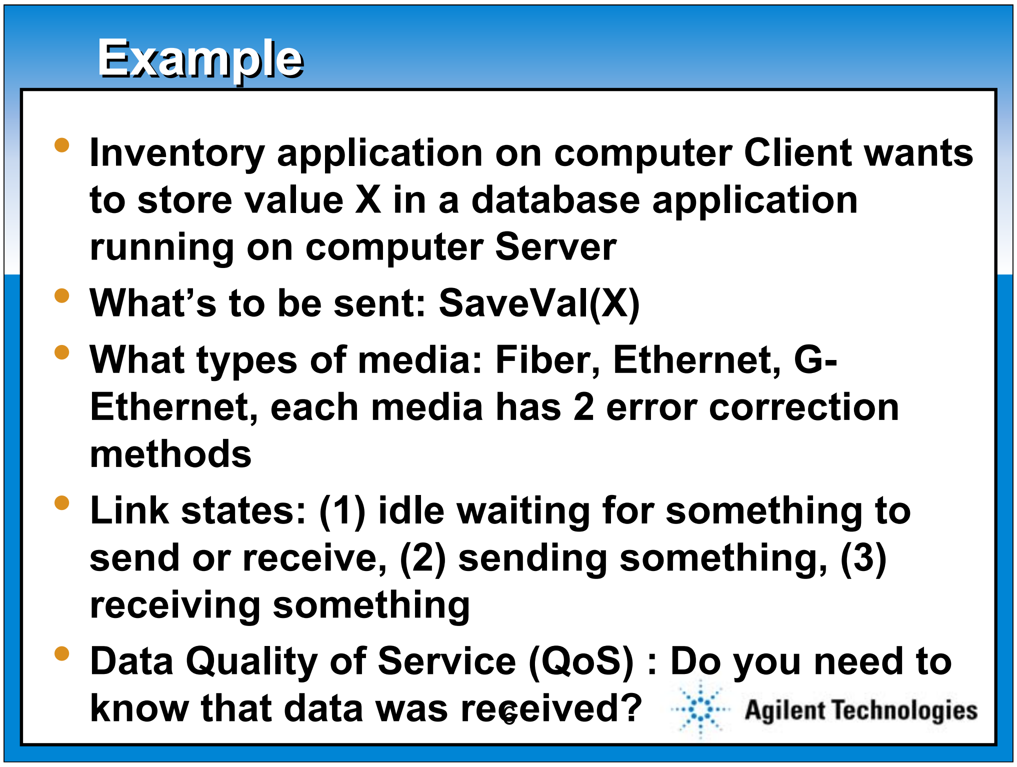
The concept of a protocol is not limited to the wireless world. ATM, SS7, X.25, and Ethernet are all protocols in the wired world. It is important to consider many things when messages and data flow:

What is the transmission medium?

What is the path?

In addition, we need to consider the state of the transmitter and the state of the receiver. Wireless protocols generally are implemented in a state machine in both the network and the handset. The proper exchange of messages cause state transitions. For example, consider making a voice call. The handset starts in an idle state, and will end in a connected or conversation state, after exchanging the needed information and assigning system resources.

This talk will not focus on the state machine, but instead on the messaging. Many of our customers have implemented a state machine that needs debugging, and our protocol logger is the tool of choice.

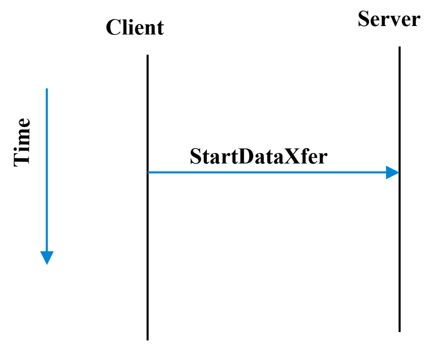


Example

- Inventory application on computer Client wants to store value X in a database application running on computer Server
- What's to be sent: SaveVal(X)
- What types of media: Fiber, Ethernet, G-Ethernet, each media has 2 error correction methods
- Link states: (1) idle waiting for something to send or receive, (2) sending something, (3) receiving something
- Data Quality of Service (QoS) : Do you need to know that data was received?

Let's consider an example. This is a wired example, with a server that manages a data base. Multiple point of sale terminals connect to this server and update the database at each transaction. One terminal wants to store a new value for a given item. Attached to the server, let's assume three alternative network connections, and also assume that two levels of error correction are possible on each. Finally, let's add one more option which, if enabled, will be a guaranteed delivery, with an acknowledgement message from the server back to the terminal.

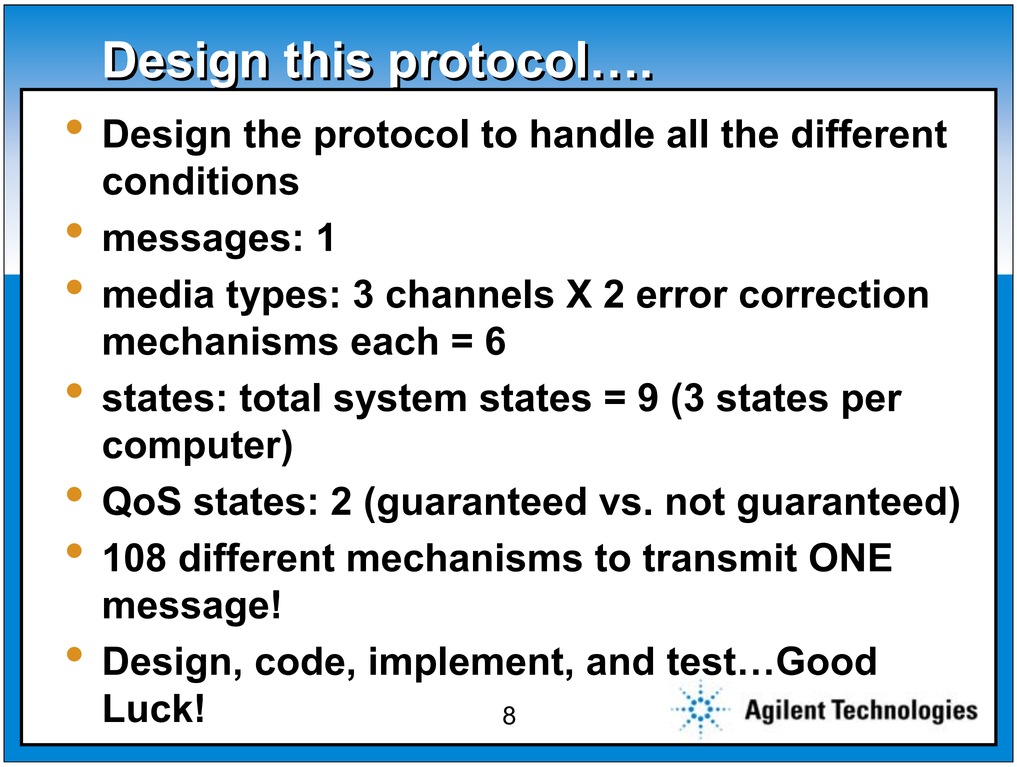
Message Sequence Chart



Bounce Diagram or
Ladder Diagram

Agilent Technologies

This is a pictorial of the data flow. In this simple case, the message is sent from the client to the server. If the server were to send a message back, it would be shown as another arrow, with the head pointing the other direction. It would be lower than the first, as time flows from top to bottom. If there were two messages, this would resemble a ladder much more, and justify its name as a ladder diagram. Another name for this is a bounce diagram.

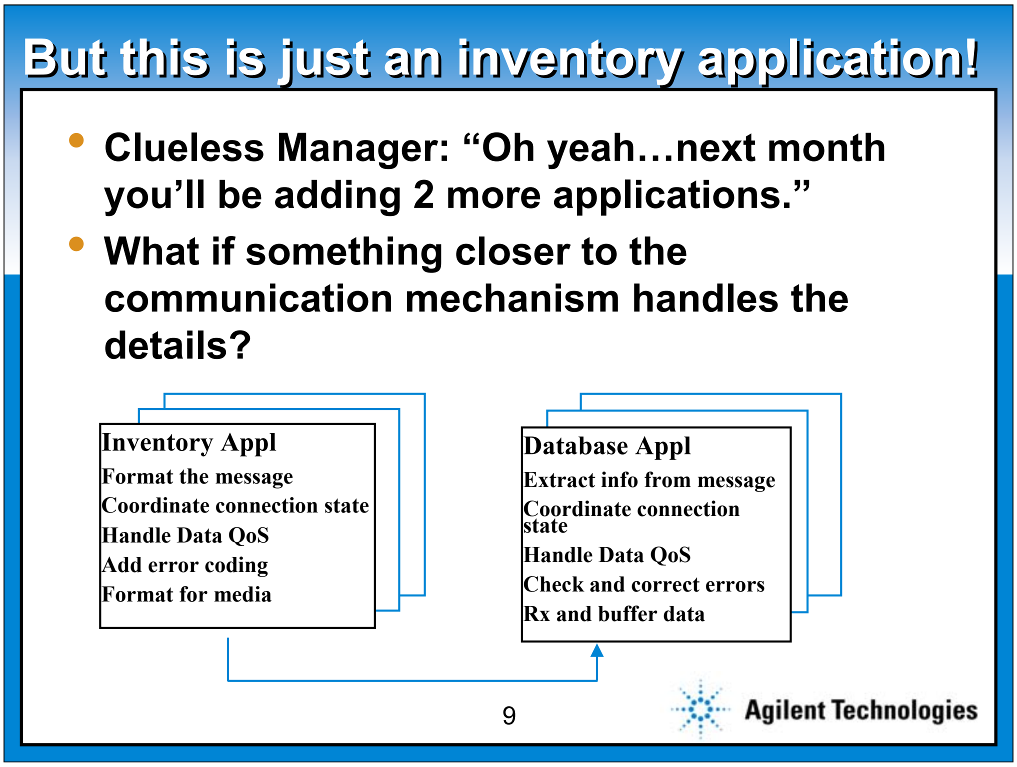


Design this protocol….

- Design the protocol to handle all the different conditions
- messages: 1
- media types: 3 channels X 2 error correction mechanisms each = 6
- states: total system states = 9 (3 states per computer)
- QoS states: 2 (guaranteed vs. not guaranteed)
- 108 different mechanisms to transmit ONE message!
- Design, code, implement, and test…Good Luck!

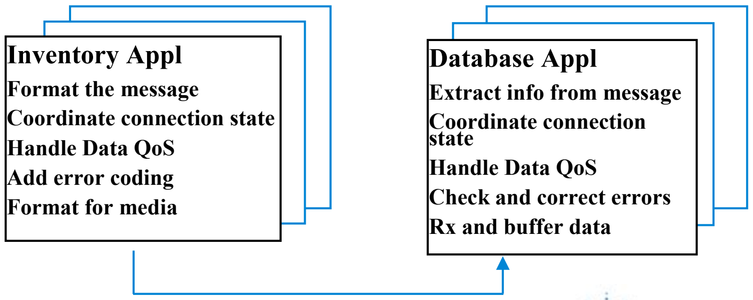
Agilent Technologies

It seems that this protocol would be very easy, but let's consider the number of possible combinations. Multiplying the number of options for each element gives up 108 possible combinations. In practice, many of these may be disallowed, but this still seems extreme for this simple example. The problem here is that the application has been tasked with all operation of the network connection.



But this is just an inventory application!

- Clueless Manager: “Oh yeah…next month you’ll be adding 2 more applications.”
- What if something closer to the communication mechanism handles the details?

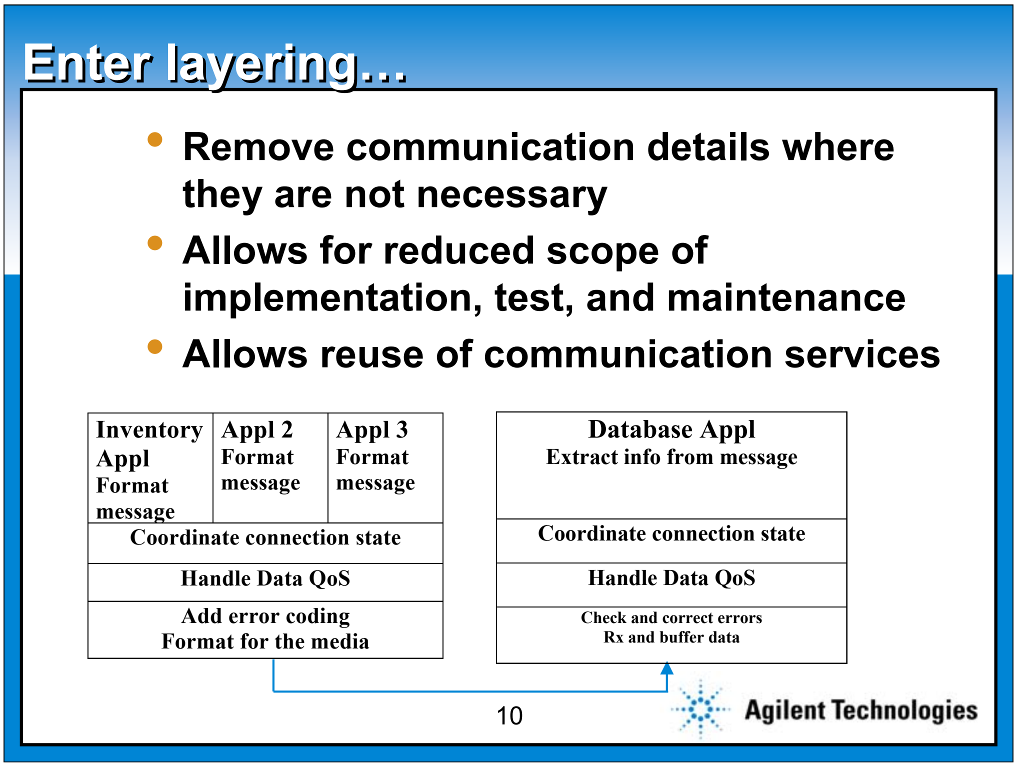


9

Agilent Technologies

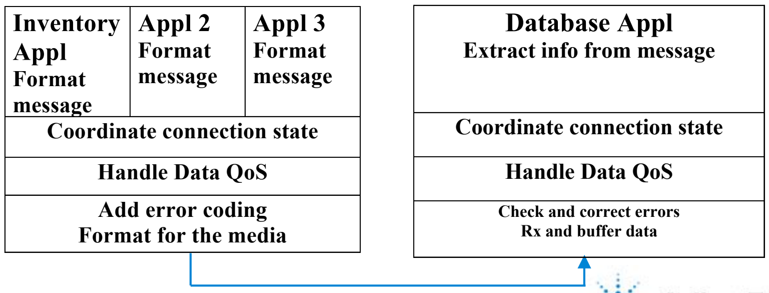
Let’s continue this example to the case where we want to add parallel applications on the same terminal to the network. This would multiply the number of options again, and quickly becomes too complex to allow for practical test.

Why don’t we relieve the applications of the job of managing the data flow, the error correction, the QoS?



Enter layering…

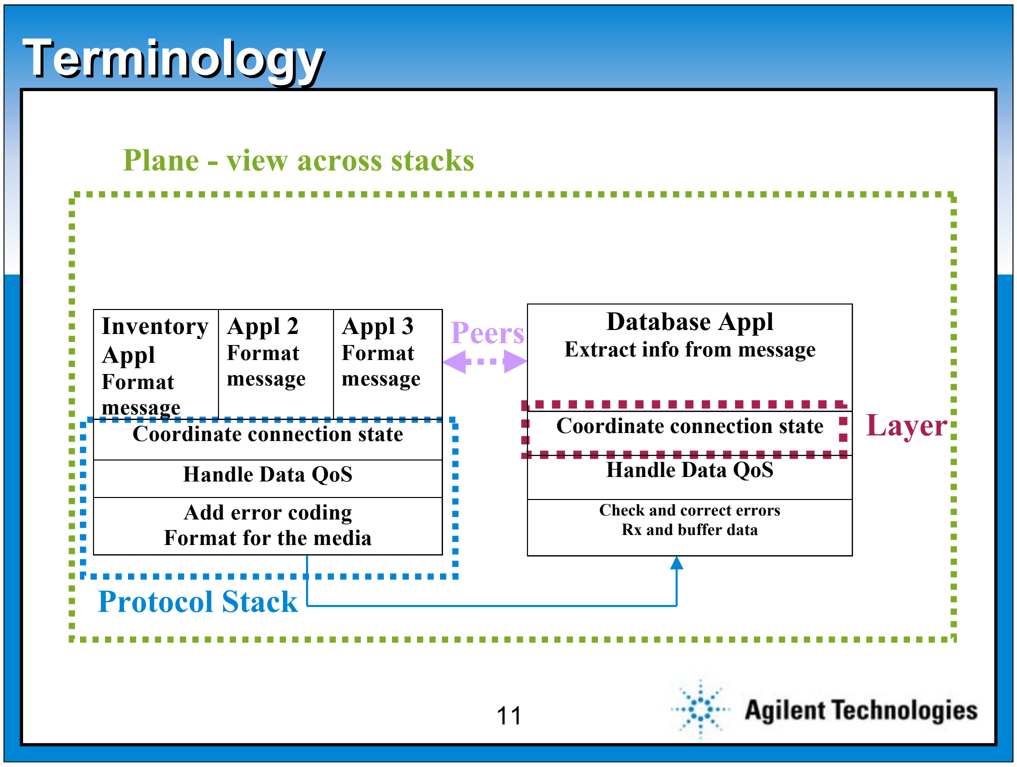
- Remove communication details where they are not necessary
- Allows for reduced scope of implementation, test, and maintenance
- Allows reuse of communication services



Inventory Appl Format message
Appl 2 Format message
Appl 3 Format message
Coordinate connection state
Handle Data QoS
Add error coding Format for the media
Database Appl Extract info from message
Coordinate connection state
Handle Data QoS
Check and correct errors Rx and buffer data

Agilent Technologies

This leads us to what has been designed for all types of networks, layering. The lowest layer has the physical connection. The application sits on top of the whole stack. Adding multiple applications requires the application itself, and modifications to the layer called, “coordinate connection state” to allow selection and identification of which application is active. AS shown here, it isn’t important to the network end because they all talk to the database. It easily could be that a second application were added to the network, as well, where different Client Applications talk to different Server Applications.



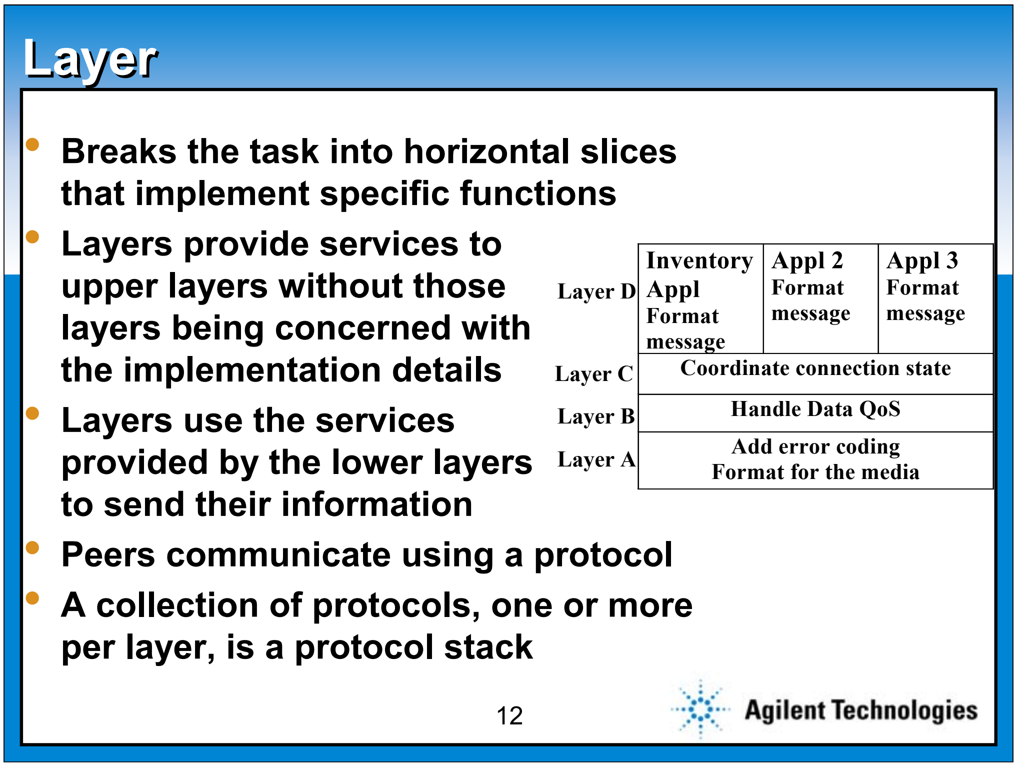
Terminology

Plane - view across stacks
Inventory Appl Format message
Appl 2 Format message
Appl 3 Format message
Coordinate connection state
Handle Data QoS
Add error coding Format for the media
Protocol Stack
Peers
Database Appl
Extract info from message
Coordinate connection state
Handle Data QoS
Check and correct errors
Rx and buffer data
Layer

Agilent Technologies

A view like this is called the plane view. It looks down on the layered structure, and shows the stack on each side. The stack is the collection of layers, and a layer is a single entity with specific role in the overall process.

Each layer in has two roles: transport messages to and from higher layers, and to exchange messages with its peer. A peer is always at the same layer on the opposite side of the link. So, while the inventory application is transported down by each of the lower layers, transported by wire to the bottom of the alternate stack, it rises up the stack and ends at the same layer. The QoS layer cannot communicate with any layer on the other side other than the OoS layer.



Layer

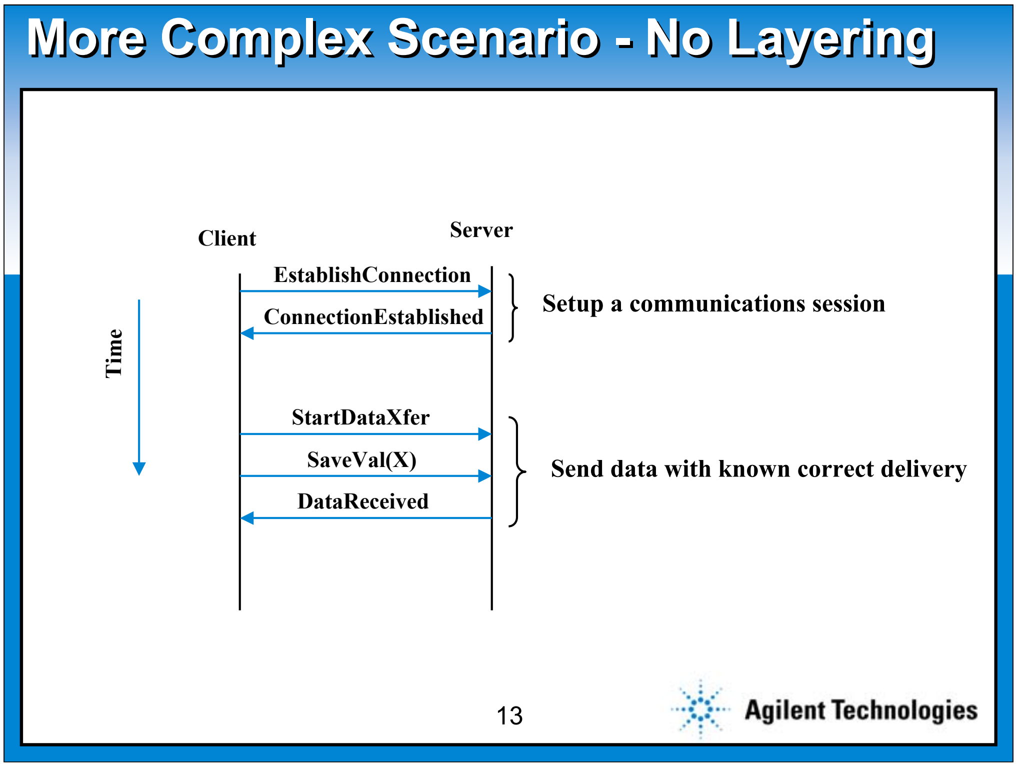
- Breaks the task into horizontal slices that implement specific functions
- Layers provide services to upper layers without those layers being concerned with the implementation details
- Layers use the services provided by the lower layers to send their information
- Peers communicate using a protocol
- A collection of protocols, one or more per layer, is a protocol stack

Layer D	Inventory Appl Format message	Appl 2 Format message	Appl 3 Format message
Layer C	Coordinate connection state		
Layer B	Handle Data QoS		
Layer A	Add error coding Format for the media		

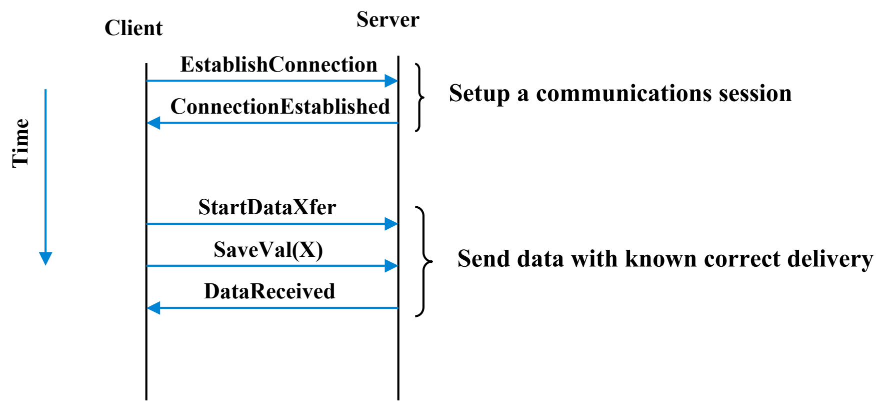
12

Agilent Technologies

By using layers, we have broken up the task into slices that can be reused. Changing an application does not require any differences in the coding or the QoS.



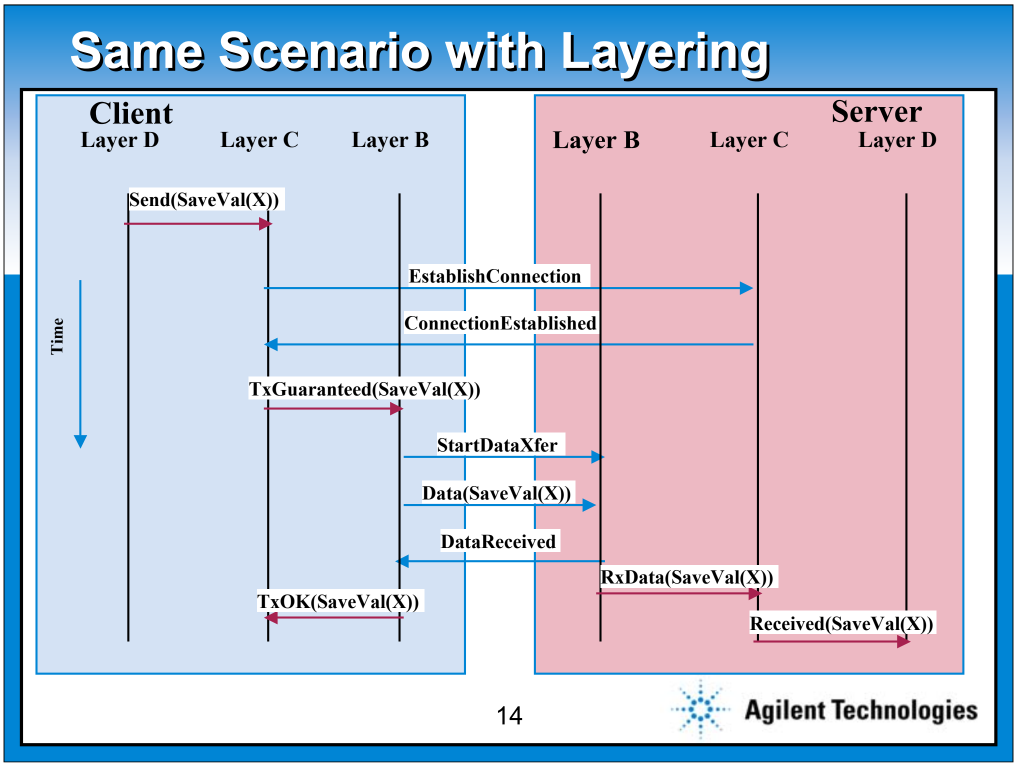
More Complex Scenario - No Layering



Agilent Technologies

Let’s consider the earlier example, but add a few more elements to the protocol. First, I want to add a request for connection, with connection granted. Then I want to add a flag before transfer of data, and I want an acknowledgement when the data is received.

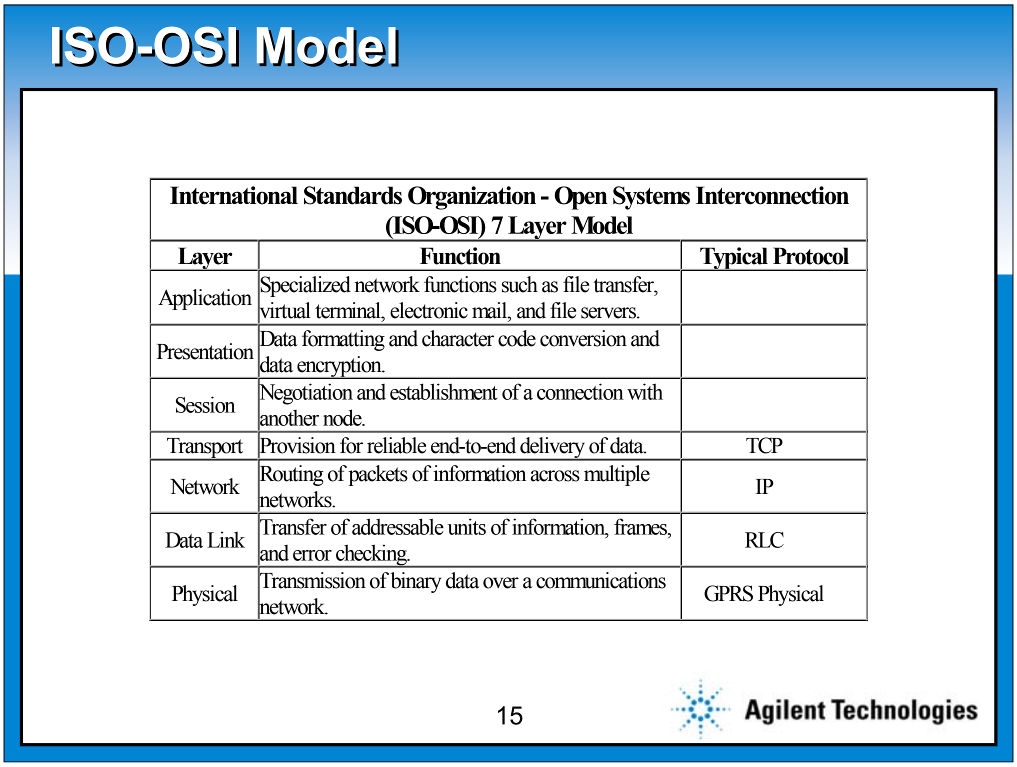
A ladder diagram for this is really quite simple.



Same Scenario with Layering

When I add layering, the ladder diagram seems to explode. This ladder diagram is unusual in that it mixes level of abstraction, which normally wouldn't be done. By this I mean that the two links labeled, Establish Connection and Connection established are Peer communications at Layer C. The actual transport below C at the Client and up to C and the Server is not shown. But the data transfer shows each transport step, from D to C, C to B, B to B Peer, B back to C, and finally C up to D. All of these steps are the transport of the D to D Peer communication.

One of the key contributions of the 8960 with the GPRS LA is the ability to put a protocol probe at any layer, and see all the peer communication that happens at that layer. By probing multiple layers, more of the picture is available.



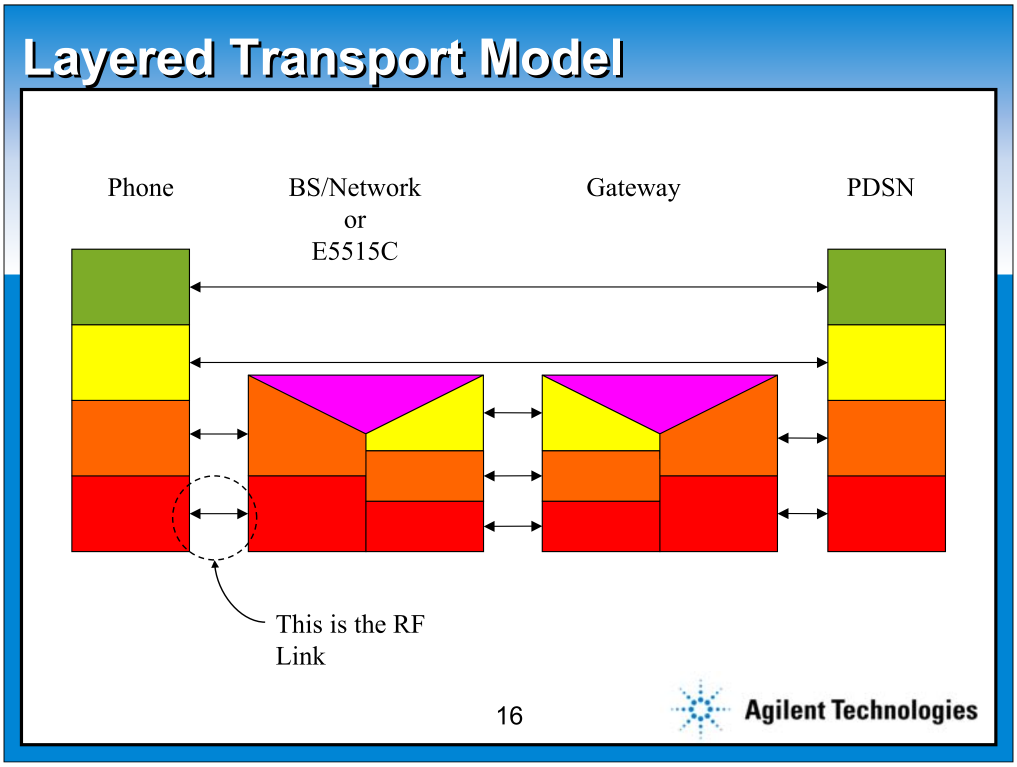
ISO-OSI Model

International Standards Organization - Open Systems Interconnection (ISO-OSI) 7 Layer Model		
Layer	**Function**	**Typical Protocol**
Application	Specialized network functions such as file transfer, virtual terminal, electronic mail, and file servers.	
Presentation	Data formatting and character code conversion and data encryption.	
Session	Negotiation and establishment of a connection with another node.	
Transport	Provision for reliable end-to-end delivery of data.	TCP
Network	Routing of packets of information across multiple networks.	IP
Data Link	Transfer of addressable units of information, frames, and error checking.	RLC
Physical	Transmission of binary data over a communications network.	GPRS Physical

Agilent Technologies

All of this layering is based on work done by the International Standards Organization in their Open Systems Interconnection 7 layer model. The physical link of GPRS does not have a unique name, it includes the coding, which can be to four different levels of error protection and the modulation. Layer 2 is comprised by the Radio Link Control (RLC) and another sub layer called the Medium Access Control (MAC). On the transmit side, these break apart a large data file into smaller packets suitable for transmission, and number each. On the Receive side, the RLC/MAC rebuilds the original large block. The higher layers are the same Internet Protocols we normally use in wired applications.

Most wireless systems violate the ISO-OSI model frequently. An example is the addition of a CRC on each data block. This is typically implemented in hardware; comprised of a shift register and a few XOR gates. As this is physical in nature, this is done as part of the physical layer, even though is specifically is a layer 2 operation in the model.



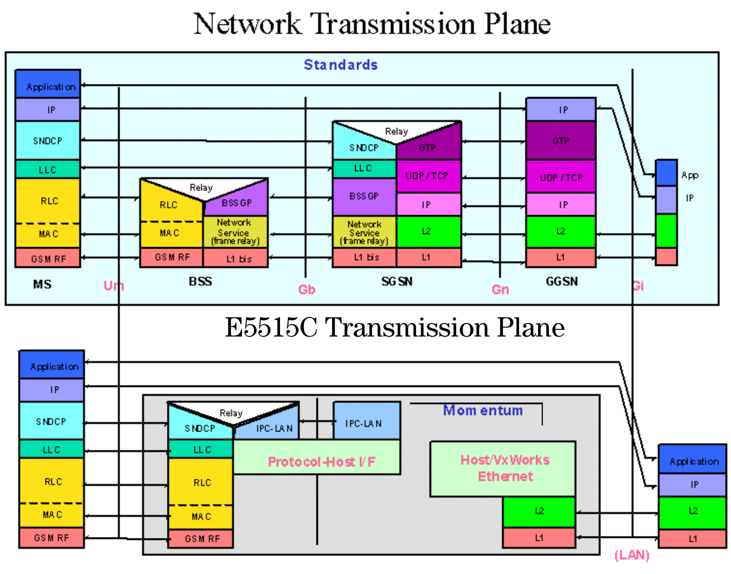
Layered Transport Model

This shows how different layers of a stack can come from different devices. The stack on the left represents the phone, the second the combination of the base station, the network, and a mapping function needed to get the data to the Internet. The third stack, the gateway, is not used for wired connections, it will be discussed later. The stack on the right is the internet. The phone has physical and data flow connections with its own cell, but the next layer up, IP, is transferred from the internet. It has been passed through each of the intermediate pathways.

The gateway is necessary to allow WAP web pages to be adjusted for a specific phone. At my latest count, there were 17 different screen sizes available for wireless web applications. The gateway would map generic data onto each of these screens.

It should be noted that the black line with two arrows at the bottom left represents the RF link in both directions. We have spent a major part of our careers measuring power, noise, harmonics, sensitivity, spectral purity of this link. Now, if you add some smarts at layer 2, it is just a modem.

GPRS and E5515C Models



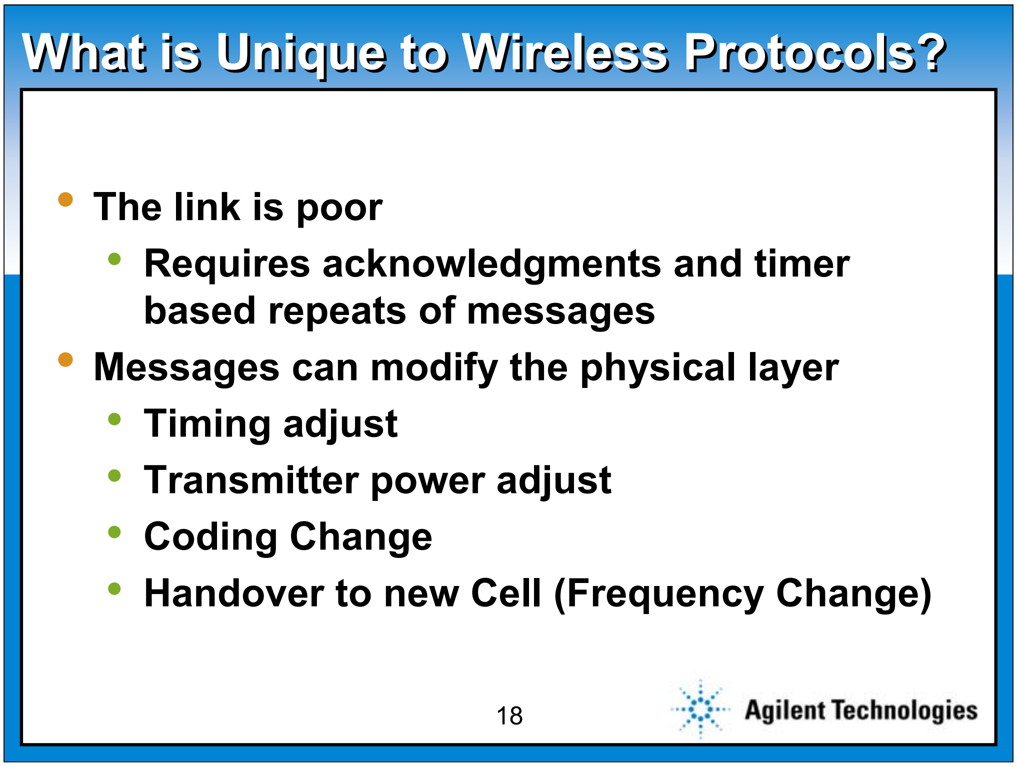
17

Agilent Technologies

The top picture shows the actual protocol stacks for the data portion of GPRS. On the left is the phone, next is the cell, then the switch, followed by a special switching network to get to the internet. Finally, on the right, is the internet itself. The gateway is not part of the GPRS standards, so it is not shown here.

On the bottom is a representation of what is inside the 8960 with the GPRS LA. All of the network elements have been elevated onto one stack, and the layers starting at IP and going have been reflected to the network port on the 8960. It has its own physical layer and flow control, using the Ethernet, and the IP layer and higher is passed through to the phone, just as in a real network.

This is the second major contribution of the 8960 with the GPRS LA. We call this the data channel, and it is a major selling point in dealing with our customers.



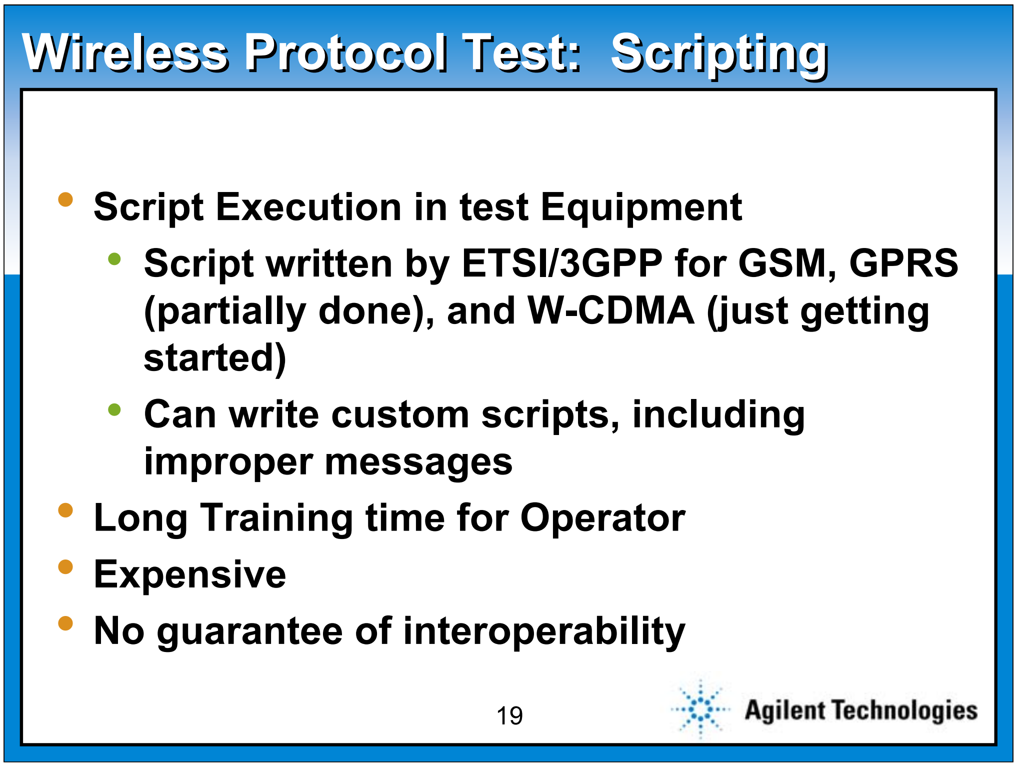
What is Unique to Wireless Protocols?

- The link is poor
  - Requires acknowledgments and timer based repeats of messages
- Messages can modify the physical layer
  - Timing adjust
  - Transmitter power adjust
  - Coding Change
  - Handover to new Cell (Frequency Change)

Agilent Technologies

Wireless Protocols differ from wireline ones in several ways. The link generally is not as good as what can be doe with real wires. This requires the use of error correcting codes and a system that forces an acknowledgement of each message along with a timer that will automatically repeat the message if it is not acknowledged.

Messages exist in wireless protocols that directly change the physical layer. Examples of this include timing adjust (in TDMA systems), power adjustments, changes in the level of coding in use, and a handover to another cell, which will be to a new channel.



Wireless Protocol Test: Scripting

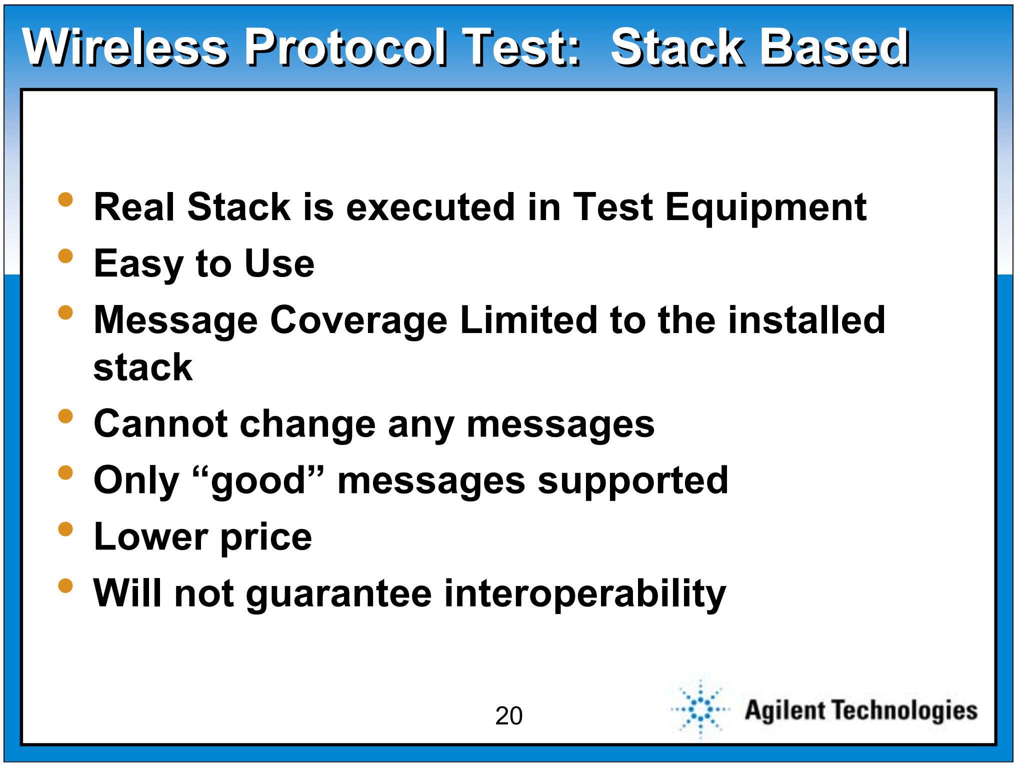
- Script Execution in test Equipment
  - Script written by ETSI/3GPP for GSM, GPRS (partially done), and W-CDMA (just getting started)
  - Can write custom scripts, including improper messages
- Long Training time for Operator
- Expensive
- No guarantee of interoperability

Agilent Technologies

The current test methodology for Protocol Test is to use scripts. There is a scripting language, Tree and Tabular Concatenated Notation, or TTCN. This is the basis of conformance test procedures for GSM, GPRS, and will be used for W-CDMA. The GPRS test script is currently not complete, the one for W-CDMA is in early development.

An advantage of using scripts is that any and every message may be supported. In addition, improper messages may be built to evaluate the error trapping in the device being tested.

The flexibility comes at a price, though. There is extensive training time required for a dedicated operator. The equipment is very expensive, and there is no guarantee that a phone that passes a script based test will interoperate with real network equipment.



Wireless Protocol Test: Stack Based

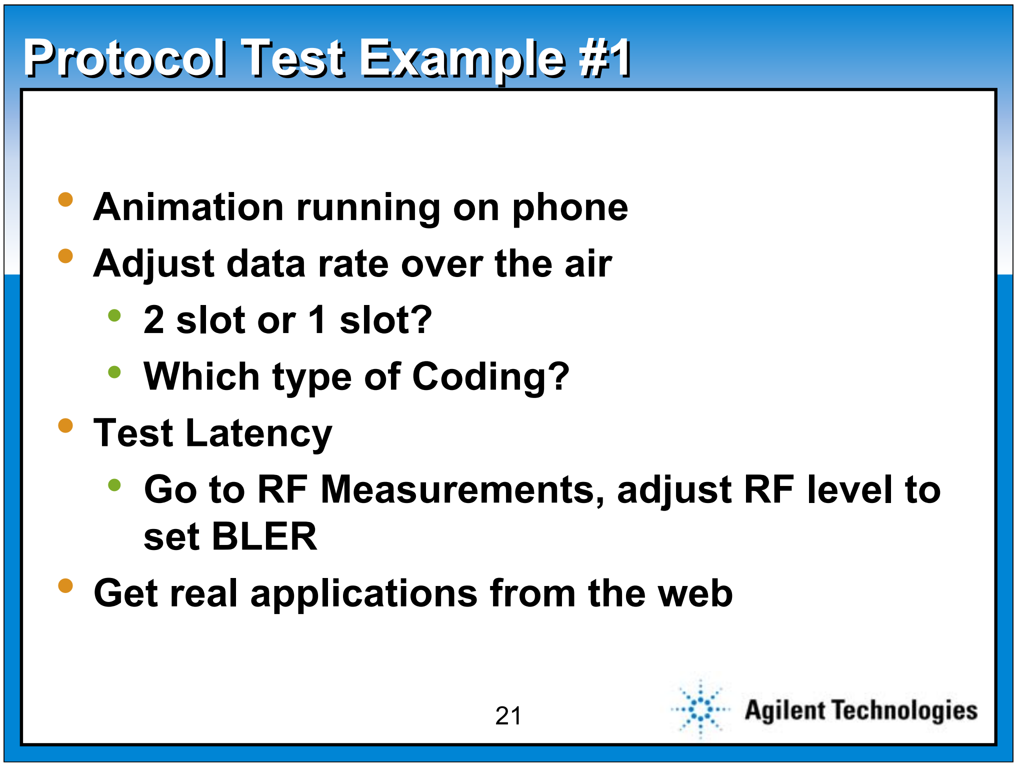
- Real Stack is executed in Test Equipment
- Easy to Use
- Message Coverage Limited to the installed stack
- Cannot change any messages
- Only “good” messages supported
- Lower price
- Will not guarantee interoperability

20

Agilent Technologies

An alternative to script based testing is to use test equipment that has a implemented the real protocol stack. Such equipment is relatively easy to use, and does not require a trained operator.

The message coverage of such a machine is limited to the implementation of the stack in the equipment. Alternative messages or fields in messages generally are open to change. Only proper messages will be sent. There is no guarantee of interoperability, here, as well.



Protocol Test Example #1

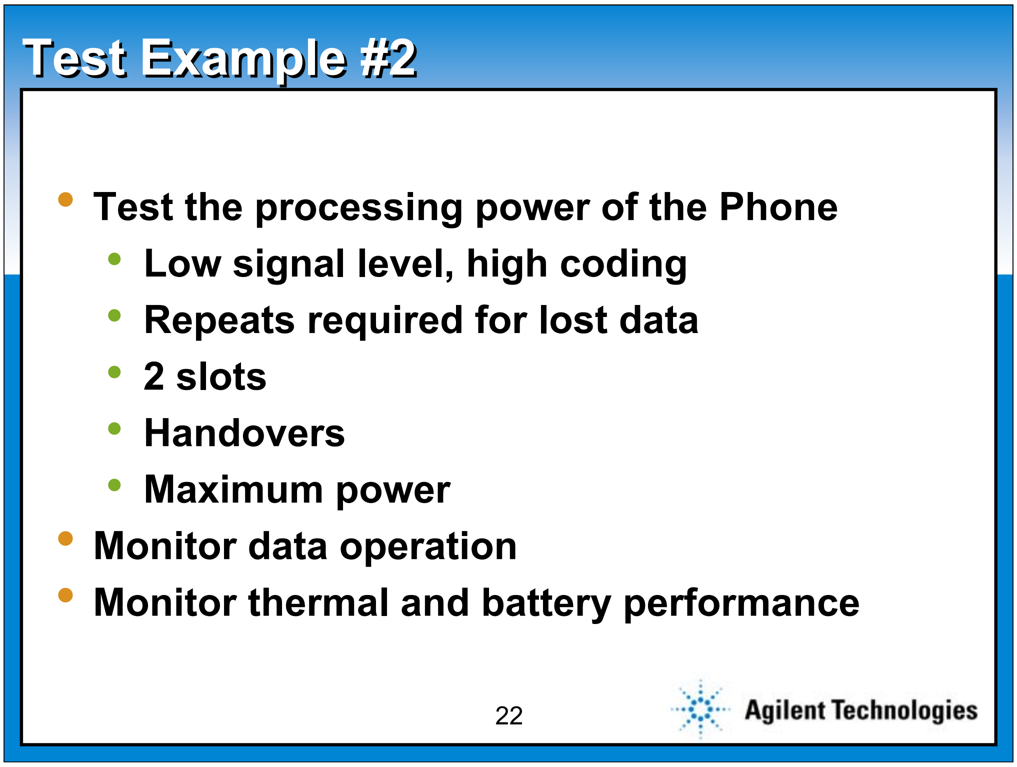
- Animation running on phone
- Adjust data rate over the air
  - 2 slot or 1 slot?
  - Which type of Coding?
- Test Latency
  - Go to RF Measurements, adjust RF level to set BLER
- Get real applications from the web

Agilent Technologies

Here’s a few examples of what questions a customer may want answered. All of these assume the Data Path is in use; in a typical application, the data source would provide the content and the protocol logger would be like the oscilloscope to look at the messages at any level.

Assume we wanted to test the capability of a browser on our phone. It can run an animation, and I want to see the effect of lowering the data rate. This can be adjusted by changing the settings on the 8960 down to one slot, and the coding to the maximum. This is the lowest data rate in GPRS, a little over 10 kbps.

An added element is to see what happens if there is variable latency in the data path. This can be done by introducing errors by setting the signal strength low. The Packet Error Rate can be monitored on the Agilent BLER mode, and the level adjusted to typical system levels, about 1%. This will introduce latency as frames are lost and re-transmitted.



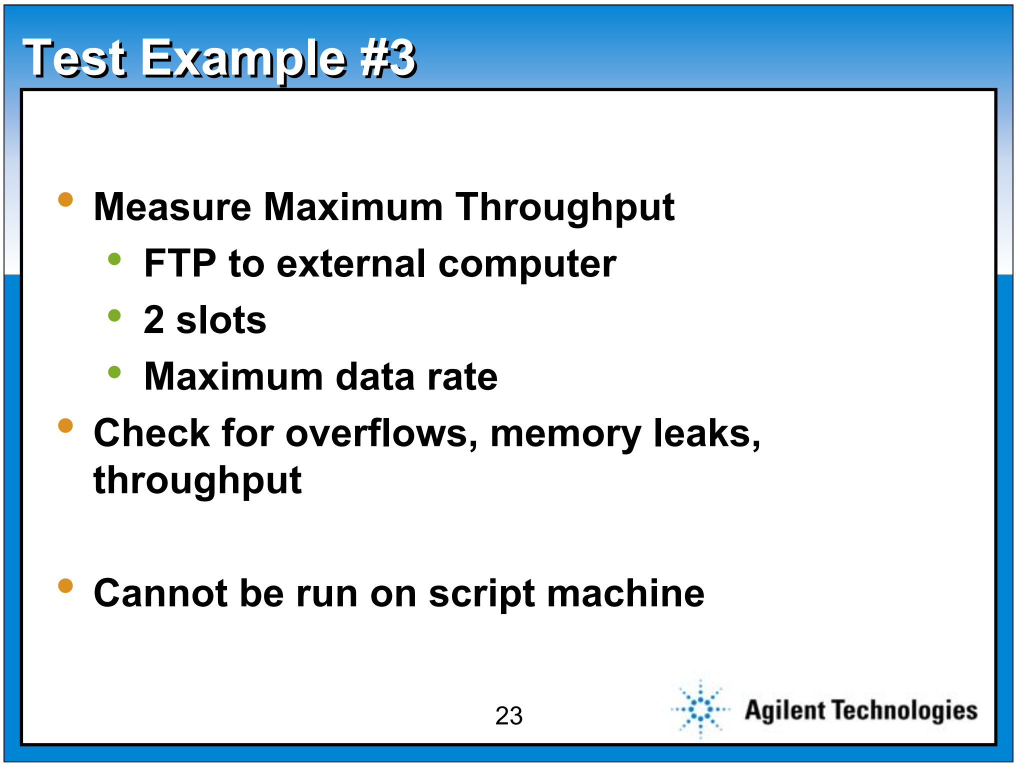
Test Example #2

- Test the processing power of the Phone
  - Low signal level, high coding
  - Repeats required for lost data
  - 2 slots
  - Handovers
  - Maximum power
- Monitor data operation
- Monitor thermal and battery performance

Agilent Technologies

In some designs, the processor in the phone performs part of the encode and decode. This may be done in hardware in some designs, as well. For those with soft processing, we would like to burden the processor with as many tasks simultaneously as we can, and look for any degradation in performance. The settings for this would be small signal level, with needs for re-transmission and insertion of the lost packets at the appropriate spot. Also, use bi-directional packet transmission. Use two slots, but with some error correction active.

It may be desirable to set the handset to maximum power, and the battery at either highest or lowest allowable voltage to see if there is any problem with that, as well.


Test Example #3

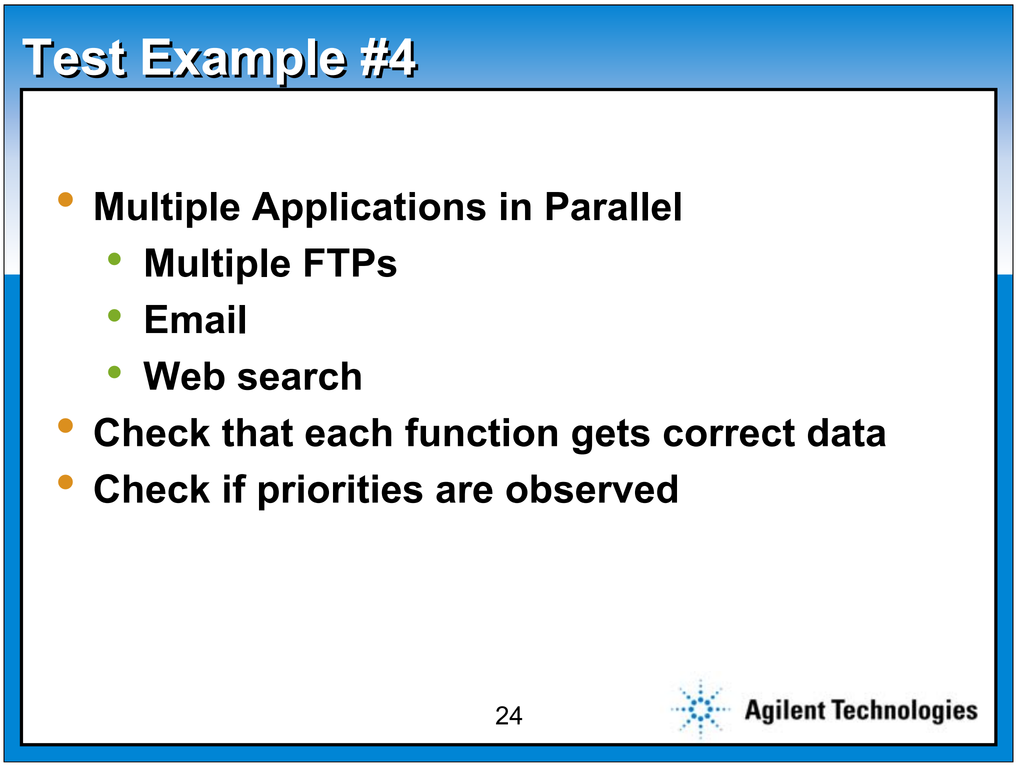
- Measure Maximum Throughput
  - FTP to external computer
  - 2 slots
  - Maximum data rate
- Check for overflows, memory leaks, throughput
- Cannot be run on script machine

23

Agilent Technologies

One thing every customer wants to test is the maximum throughput available for their handset. Every network operator wants this data, and until now, there was no way to test for it. We believe the 8960 with the E6701A GPRS LA is capable of delivering data on four timeslots, with the lowest coding (no coding). This corresponds to about 85 kbps. This has never been verified with any phone, because there aren't any available that support that rate.

Current phone designs use one Local Oscillator for both Transmit and Receive. This limits the design to about 5 total slots combined for up and down links. Going to higher bandwidths, as allowed by the standards, would require much more complexity in the phone designs.



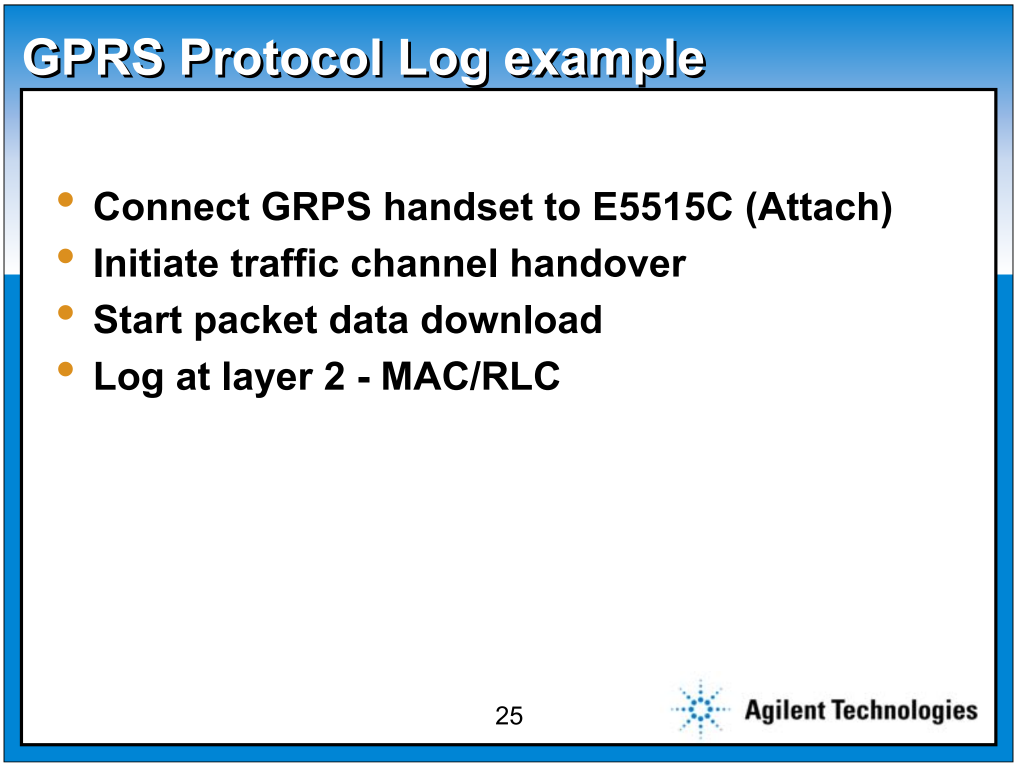
Test Example #4

- Multiple Applications in Parallel
  - Multiple FTPs
  - Email
  - Web search
- Check that each function gets correct data
- Check if priorities are observed

24

Agilent Technologies

It is possible to run multiple applications on one network connection. For example, on my PC I can start 4 simultaneous FTP downloads, download an email attachment, surf a web different than the FTPs, and share resources such as disk drives or printers. How all this data comes through one cable and gets to the right application is unknown to me, but it certainly does happen. There could be a wireless equivalent that needs to be tested.



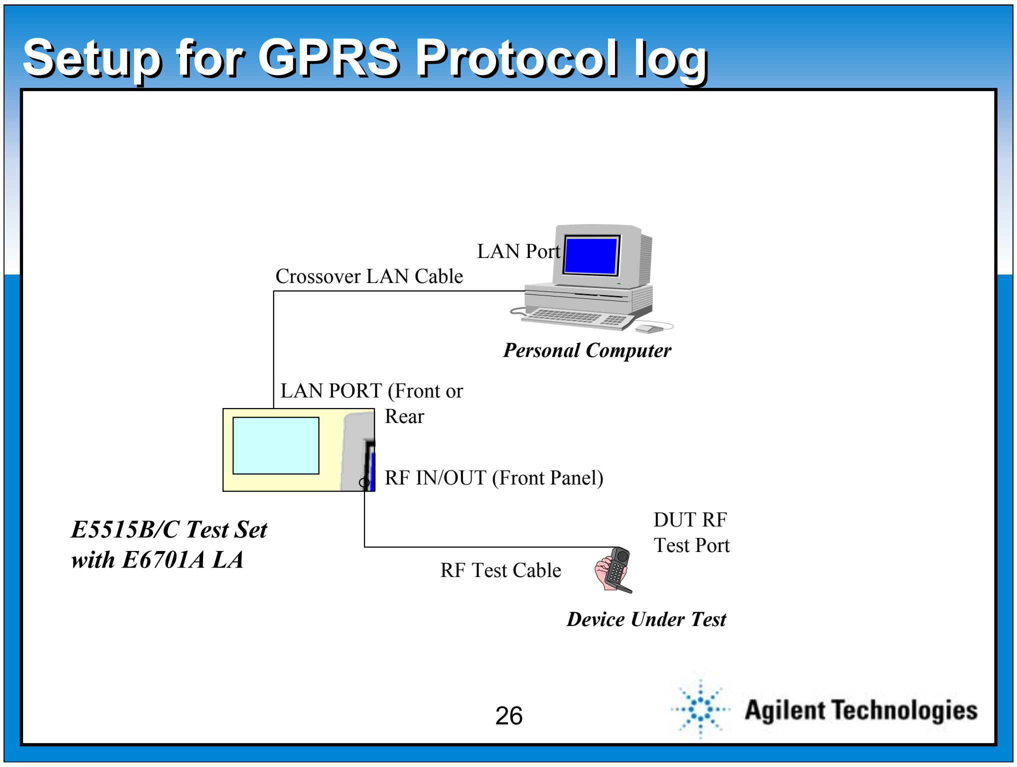
GPRS Protocol Log example

- Connect GRPS handset to E5515C (Attach)
- Initiate traffic channel handover
- Start packet data download
- Log at layer 2 - MAC/RLC

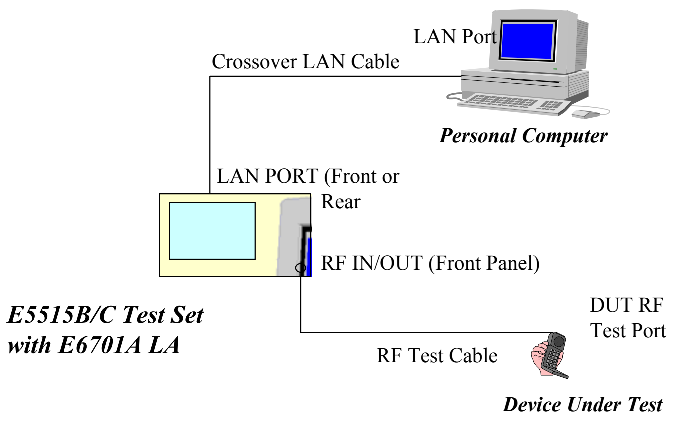
Agilent Technologies

We are now going to see a demonstration of the 8960 and a GPRS phone. We will turn on a phone, and watch it attach to the test set. We will then initiate what appears to be a handover to channel 60 on the test set, followed by the retrieval of an Icon from our gateway software. All of this will be logged with a probe set to layer 2, the MAC/RLC layer.

The available probe locations and their explanations will follow.



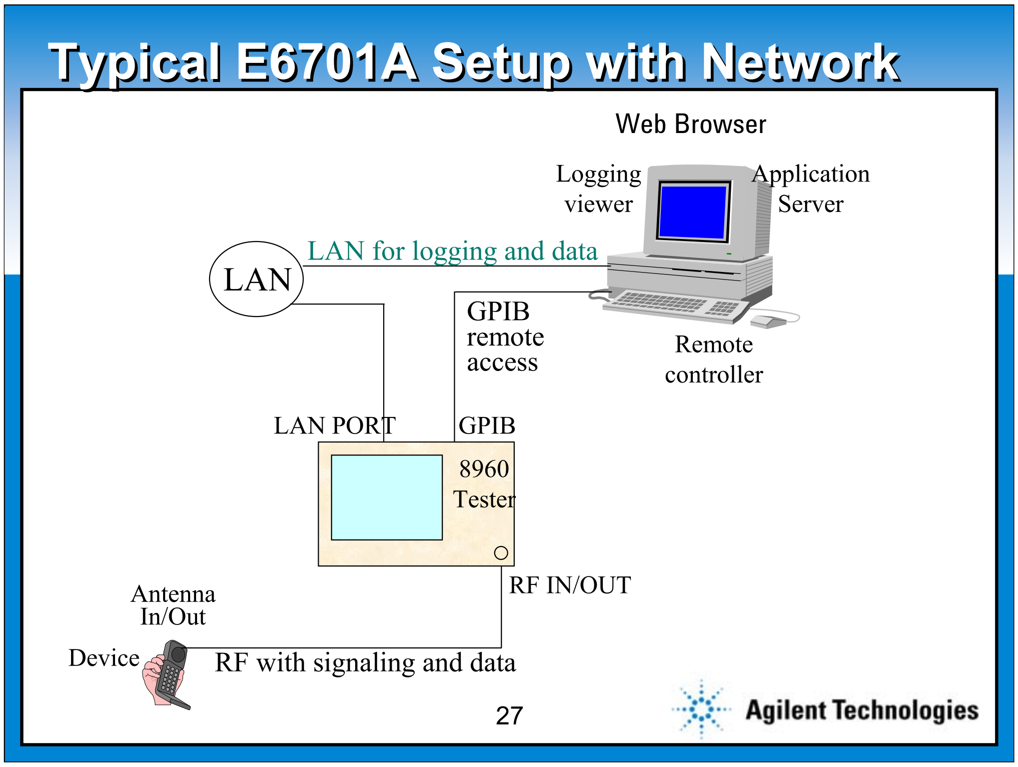
Setup for GPRS Protocol log



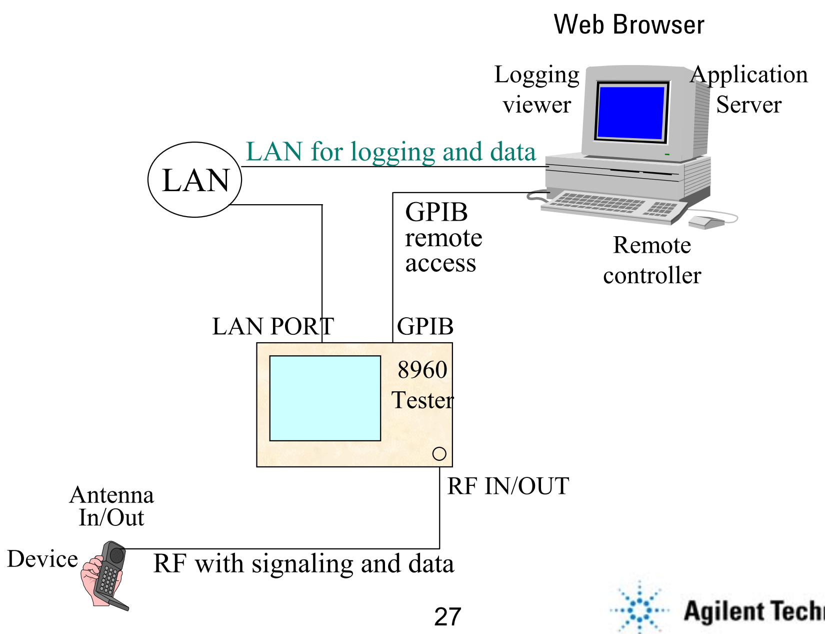
Agilent Technologies

Two setups are possible for this test. This picture shows a PC connected directly to the 8960 using a crossover LAN cable. An alternate configuration would add a real network between the 8960 and the PC, shown in the next slide. Since all the actions are driven by real internet addresses, there is no functional difference between the two.

This setup can be used where connection to a real LAN is not practical; two fixed IP addresses are needed for this, one for the test set, and one for the phone. This may be difficult, particularly in a customer's facility. This setup will limit the web content to applications directly supported by the PC, as there will be no internet connection available.



Typical E6701A Setup with Network



Agilent Technologies

Shown here is the alternate setup for protocol testing. The use of a LAN that has a firewall poses a few unique problems, but access to the web is still available by using a Proxy.

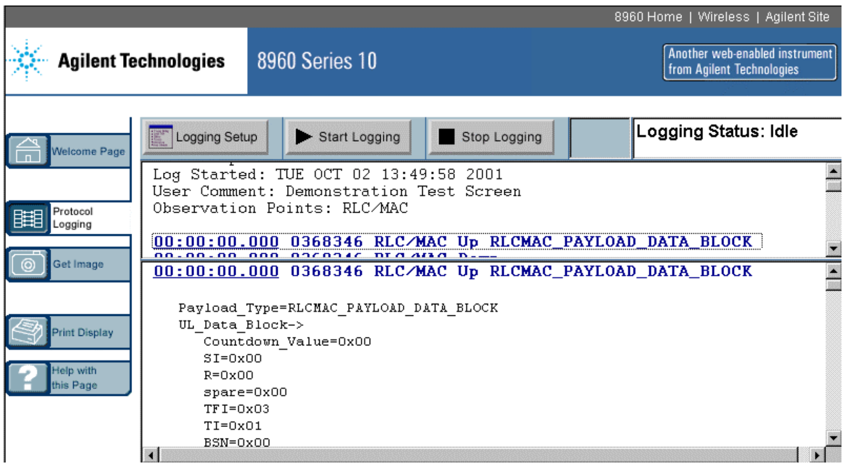
Using an external network is preferable to the setup with crossover cable because it allows access to real internet sites. This setup requires the assignment of two fixed IP addresses, one for the 8960, and one for the phone, for it to work.

Interactive Question

- What can the Internet Connect be used for?
  - A. Throughput test
  - B. Browser Functions
  - C. Evaluate a web application
  - D. All of the above

Agilent Technologies

Typical Protocol Log Displayed on PC



```
Log Started: TUE OCT 02 13:49:58 2001
User Comment: Demonstration Test Screen
Observation Points: RLC/MAC

00:00:00.000 0368346 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
```

```
00:00:00.000 0368346 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK

    Payload_Type=RLCMAC_PAYLOAD_DATA_BLOCK
    UL_Data_Block->
       Countdown_Value=0x00
       SI=0x00
       R=0x00
       spare=0x00
       TFI=0x03
       TI=0x01
       BSN=0x00
```

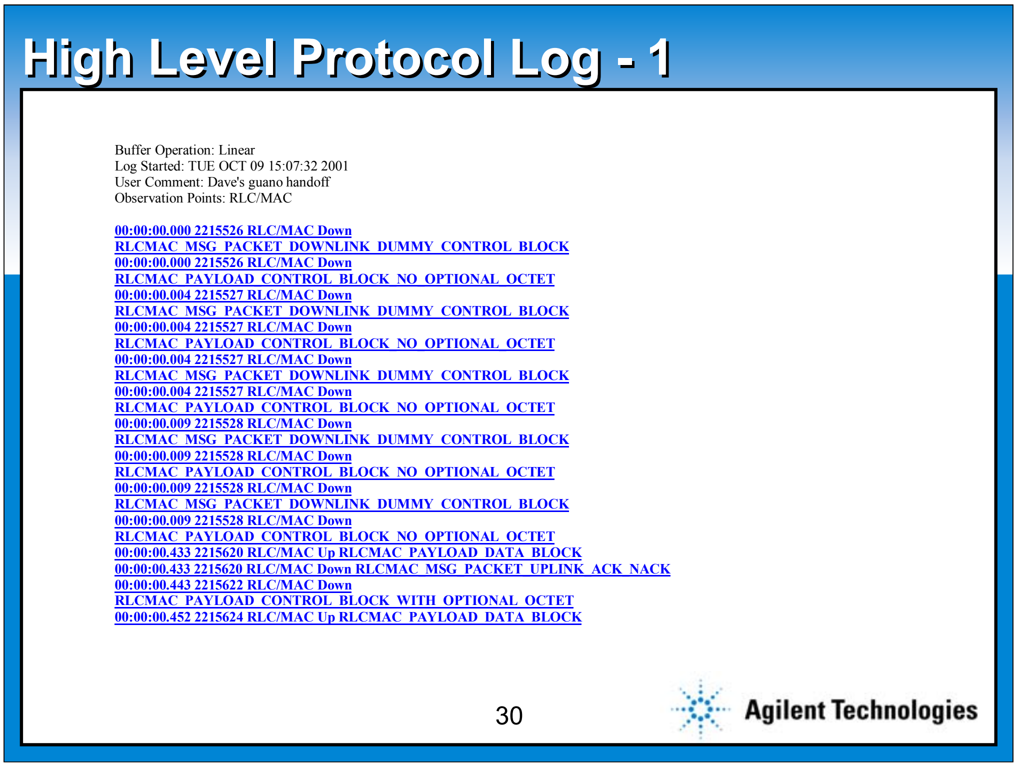
Agilent Technologies

The protocol log interface runs on a web page supported by the test equipment. The upper window is the high level protocol log. This is a time sequenced list of each message and its direction. The lower window is the detailed protocol log. It contains all the messages from the upper log, as well as the breakdown of each message with display of each internal field. The two screens are hot linked to each other; clicking on a message in the top screen will bring that message to the lower screen.

Also available on the instrument web page is the ability to capture the screen. This can then be saved either as a bitmap or a TIFF file.

Questions that are sure to come from our customers are about triggers and filters, common functions on a protocol analyzer.

The answer is that we support neither of these. This is our first release, and a simplistic User Interface was chosen to get the product out quickly. It is expected that these features will be included in later upgrades to the protocol analysis environment.



High Level Protocol Log - 1

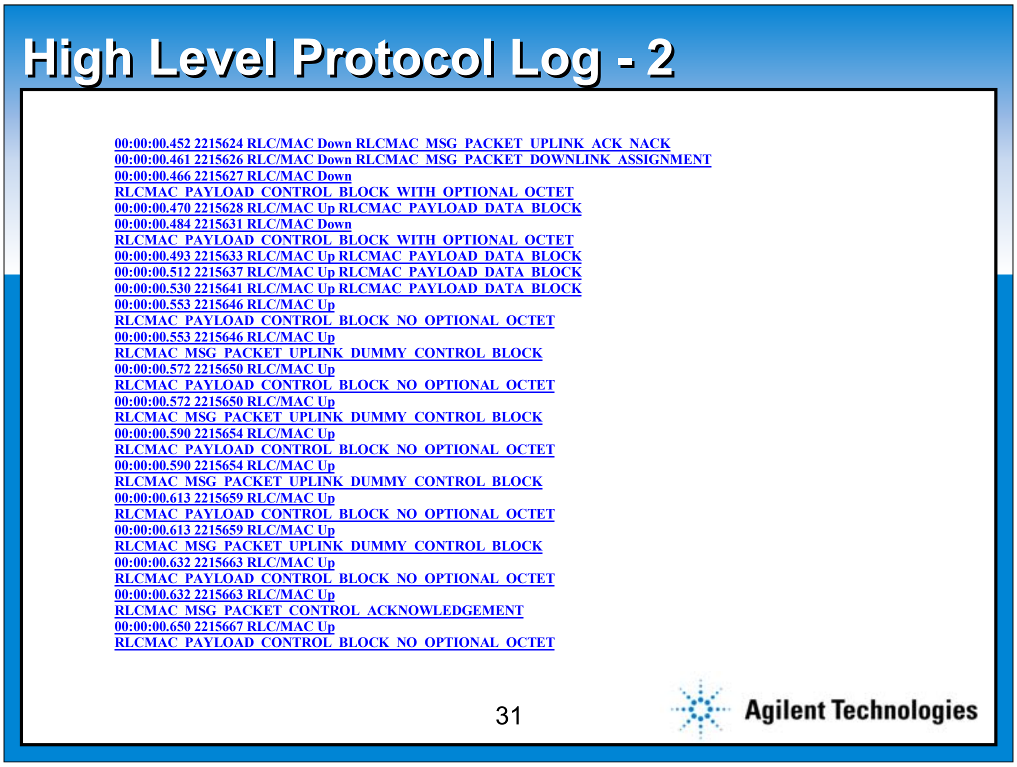
```
Buffer Operation: Linear
Log Started: TUE OCT 09 15:07:32 2001
User Comment: Dave's guano handoff
Observation Points: RLC/MAC

00:00:00.000 2215526 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:00.000 2215526 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.004 2215527 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:00.004 2215527 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.004 2215527 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:00.004 2215527 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.009 2215528 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:00.009 2215528 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.009 2215528 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:00.009 2215528 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.433 2215620 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.433 2215620 RLC/MAC Down RLCMAC_MSG_PACKET_UPLINK_ACK_NACK
00:00:00.443 2215622 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:00.452 2215624 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
```

30

Agilent Technologies

The output of the protocol logger is two sets of logged data: a list of all of the messages, and a list of all the messages with each message broken down and showing the contents of every field in each message. The two are hyperlinked to each other, clicking a message in the high level screen will jump to that message in the low level screen, and vice versa.



High Level Protocol Log - 2

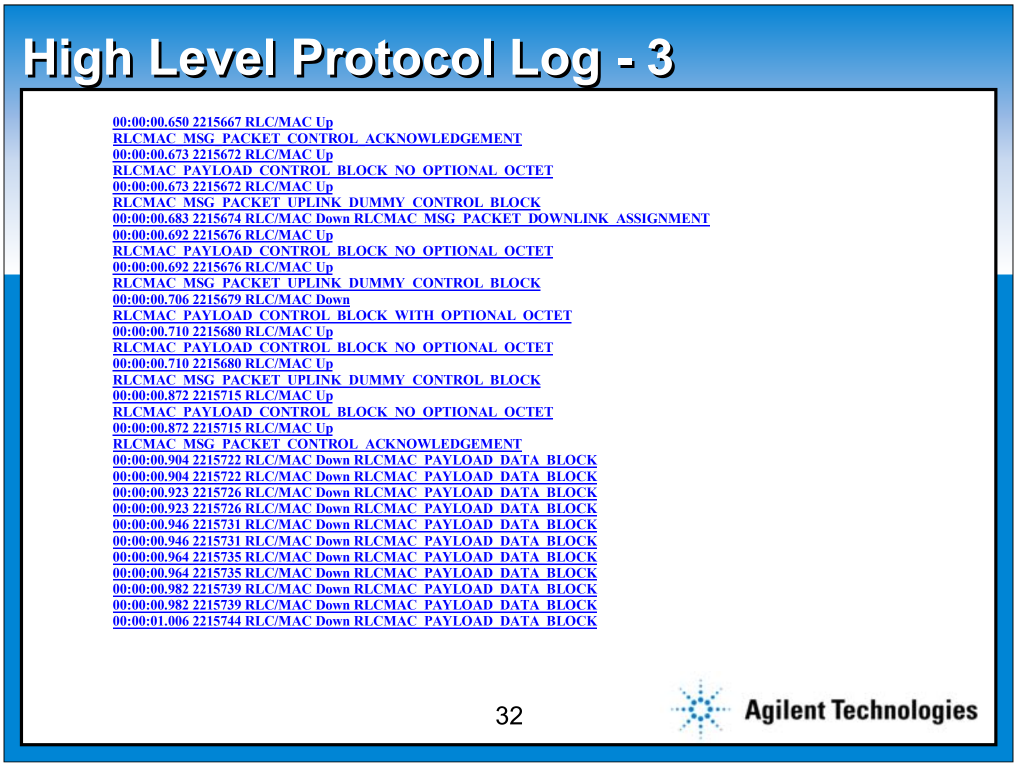
```
00:00:00.452 2215624 RLC/MAC Down RLCMAC_MSG_PACKET_UPLINK_ACK_NACK
00:00:00.461 2215626 RLC/MAC Down RLCMAC_MSG_PACKET_DOWNLINK_ASSIGNMENT
00:00:00.466 2215627 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:00.470 2215628 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.484 2215631 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:00.493 2215633 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.512 2215637 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.530 2215641 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.553 2215646 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.553 2215646 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:00.572 2215650 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.572 2215650 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:00.590 2215654 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.590 2215654 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:00.613 2215659 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.613 2215659 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:00.632 2215663 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.632 2215663 RLC/MAC Up
RLCMAC_MSG_PACKET_CONTROL_ACKNOWLEDGEMENT
00:00:00.650 2215667 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
```

31

Agilent Technologies

You can see that a whole lot of overhead messages are being exchanged between the handset and the test set; this is a characteristic of packet data.

The message of interest has not yet been passed to the handset.

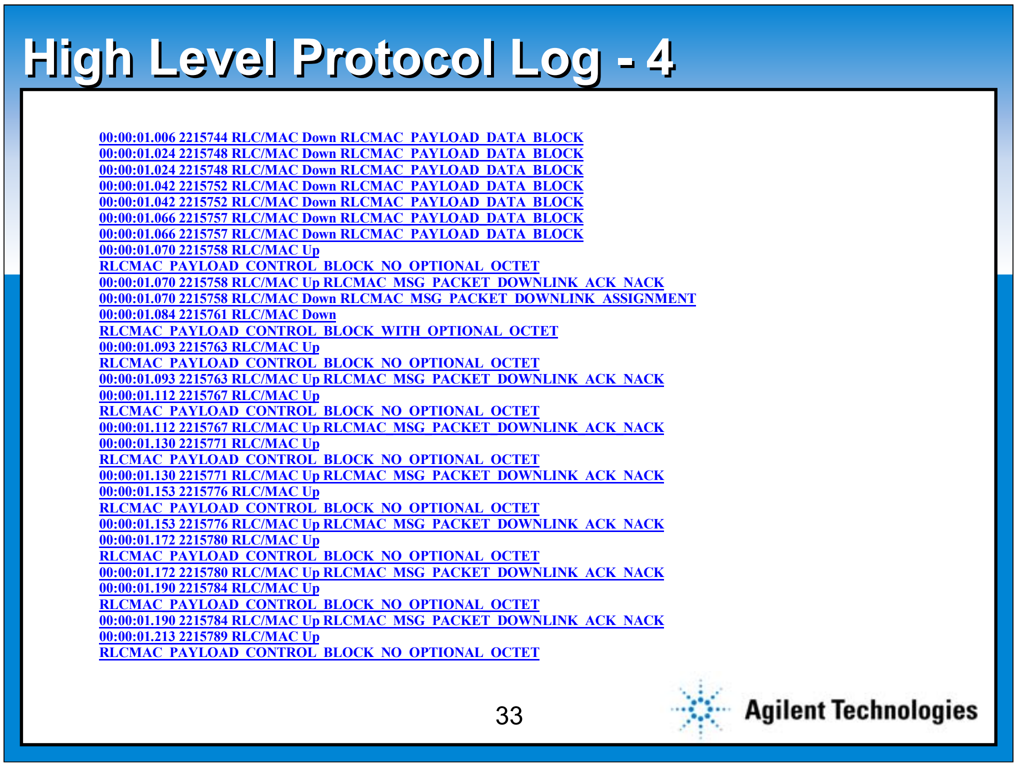


High Level Protocol Log - 3

```
00:00:00.650 2215667 RLC/MAC Up
RLCMAC_MSG_PACKET_CONTROL_ACKNOWLEDGEMENT
00:00:00.673 2215672 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.673 2215672 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:00.683 2215674 RLC/MAC Down RLCMAC_MSG_PACKET_DOWNLINK_ASSIGNMENT
00:00:00.692 2215676 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.692 2215676 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:00.706 2215679 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:00.710 2215680 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.710 2215680 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:00.872 2215715 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:00.872 2215715 RLC/MAC Up
RLCMAC_MSG_PACKET_CONTROL_ACKNOWLEDGEMENT
00:00:00.904 2215722 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.904 2215722 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.923 2215726 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.923 2215726 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.946 2215731 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.946 2215731 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.964 2215735 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.964 2215735 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.982 2215739 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:00.982 2215739 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.006 2215744 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
```

Agilent Technologies

More ...

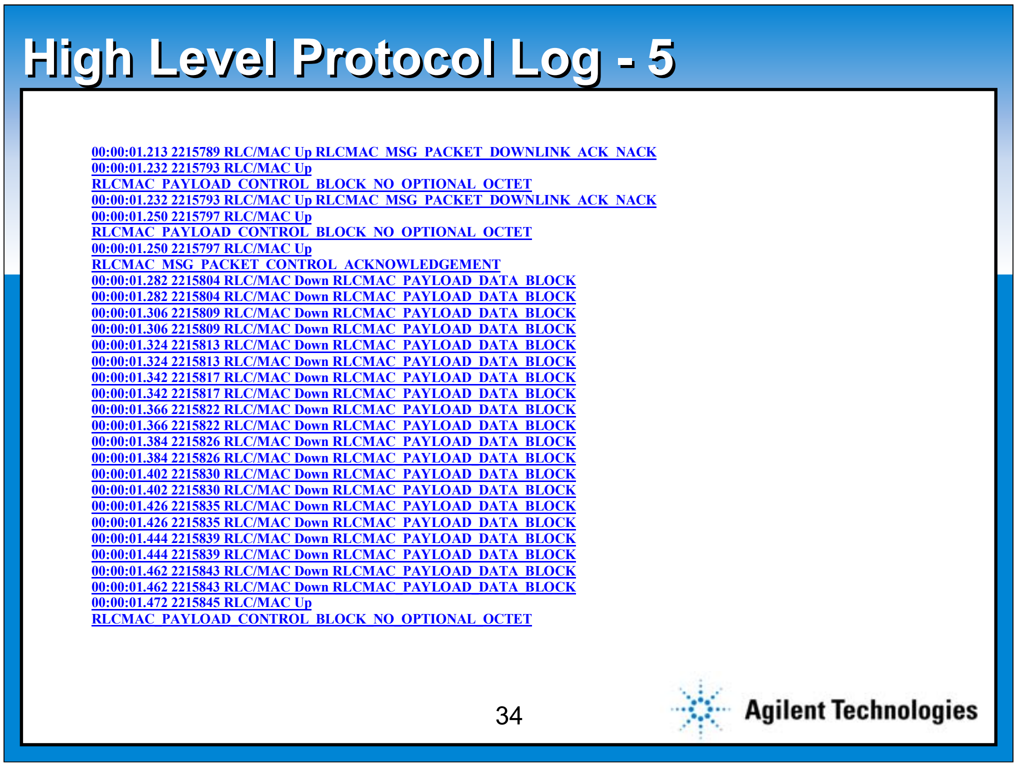


High Level Protocol Log - 4

```
00:00:01.006 2215744 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.024 2215748 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.024 2215748 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.042 2215752 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.042 2215752 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.066 2215757 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.066 2215757 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.070 2215758 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.070 2215758 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.070 2215758 RLC/MAC Down RLCMAC_MSG_PACKET_DOWNLINK_ASSIGNMENT
00:00:01.084 2215761 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:01.093 2215763 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.093 2215763 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.112 2215767 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.112 2215767 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.130 2215771 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.130 2215771 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.153 2215776 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.153 2215776 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.172 2215780 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.172 2215780 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.190 2215784 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.190 2215784 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.213 2215789 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
```

Agilent Technologies

More ...

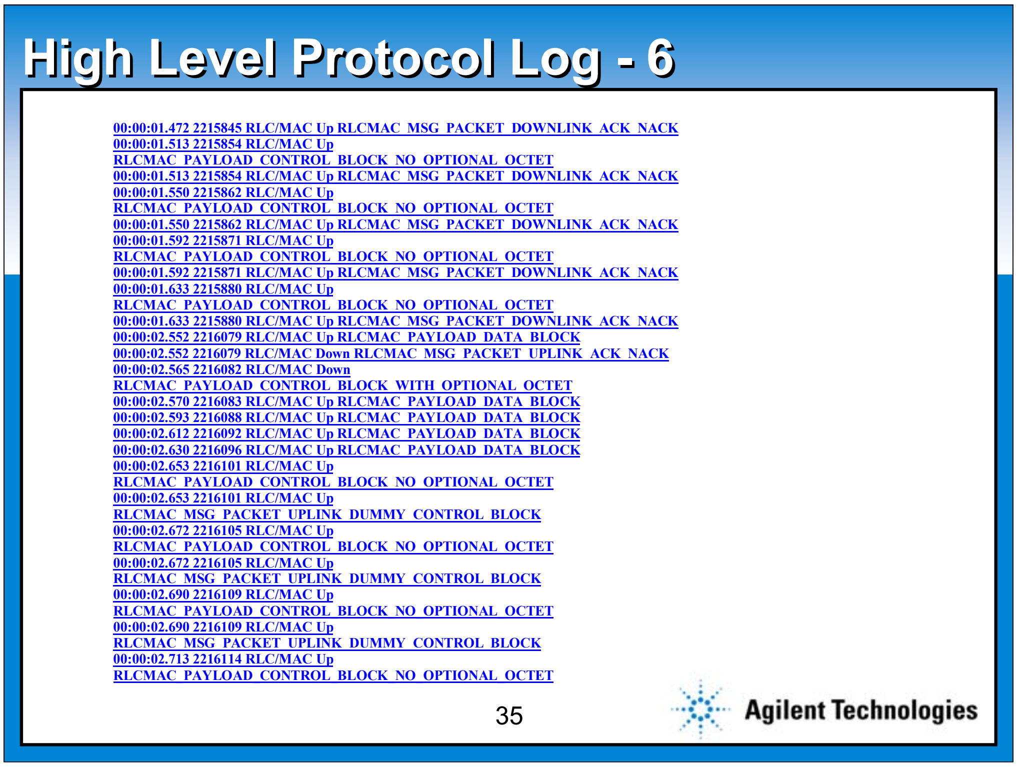


High Level Protocol Log - 5

```
00:00:01.213 2215789 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.232 2215793 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.232 2215793 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.250 2215797 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.250 2215797 RLC/MAC Up
RLCMAC_MSG_PACKET_CONTROL_ACKNOWLEDGEMENT
00:00:01.282 2215804 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.282 2215804 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.306 2215809 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.306 2215809 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.324 2215813 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.324 2215813 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.342 2215817 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.342 2215817 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.366 2215822 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.366 2215822 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.384 2215826 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.384 2215826 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.402 2215830 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.402 2215830 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.426 2215835 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.426 2215835 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.444 2215839 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.444 2215839 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.462 2215843 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.462 2215843 RLC/MAC Down RLCMAC_PAYLOAD_DATA_BLOCK
00:00:01.472 2215845 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
```

Agilent Technologies

More again ...



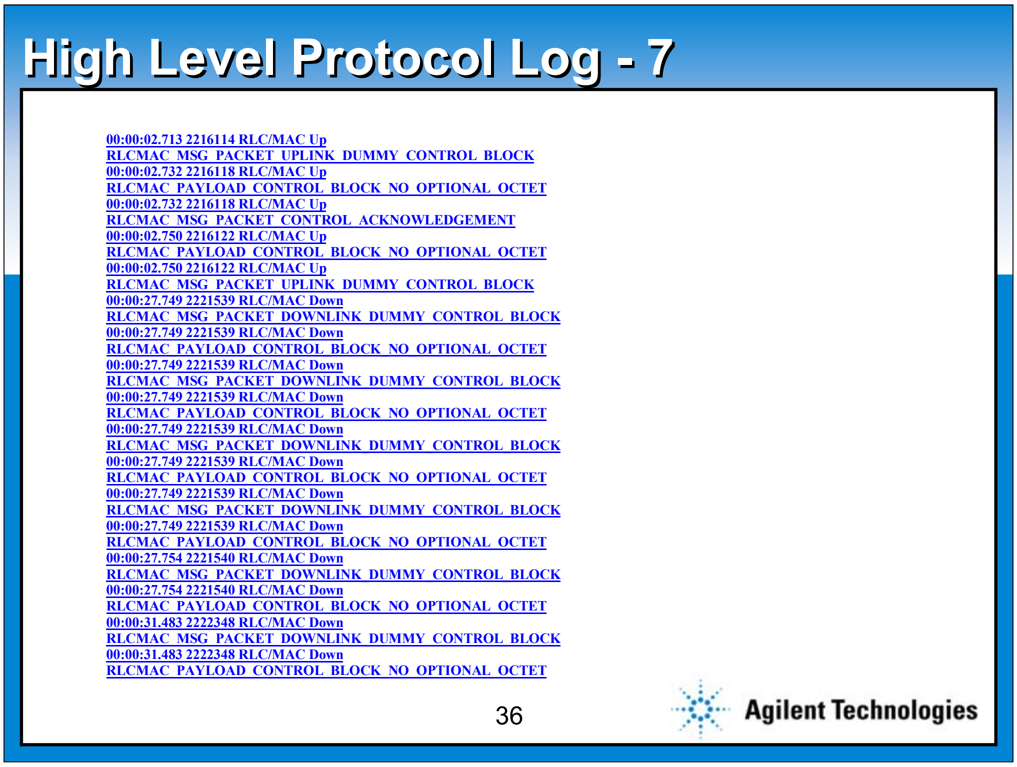
High Level Protocol Log - 6

```
00:00:01.472 2215845 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.513 2215854 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.513 2215854 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.550 2215862 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.550 2215862 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.592 2215871 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.592 2215871 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:01.633 2215880 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:01.633 2215880 RLC/MAC Up RLCMAC_MSG_PACKET_DOWNLINK_ACK_NACK
00:00:02.552 2216079 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:02.552 2216079 RLC/MAC Down RLCMAC_MSG_PACKET_UPLINK_ACK_NACK
00:00:02.565 2216082 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:02.570 2216083 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:02.593 2216088 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:02.612 2216092 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:02.630 2216096 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:02.653 2216101 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:02.653 2216101 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:02.672 2216105 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:02.672 2216105 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:02.690 2216109 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:02.690 2216109 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:02.713 2216114 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
```

35

Agilent Technologies

More again …



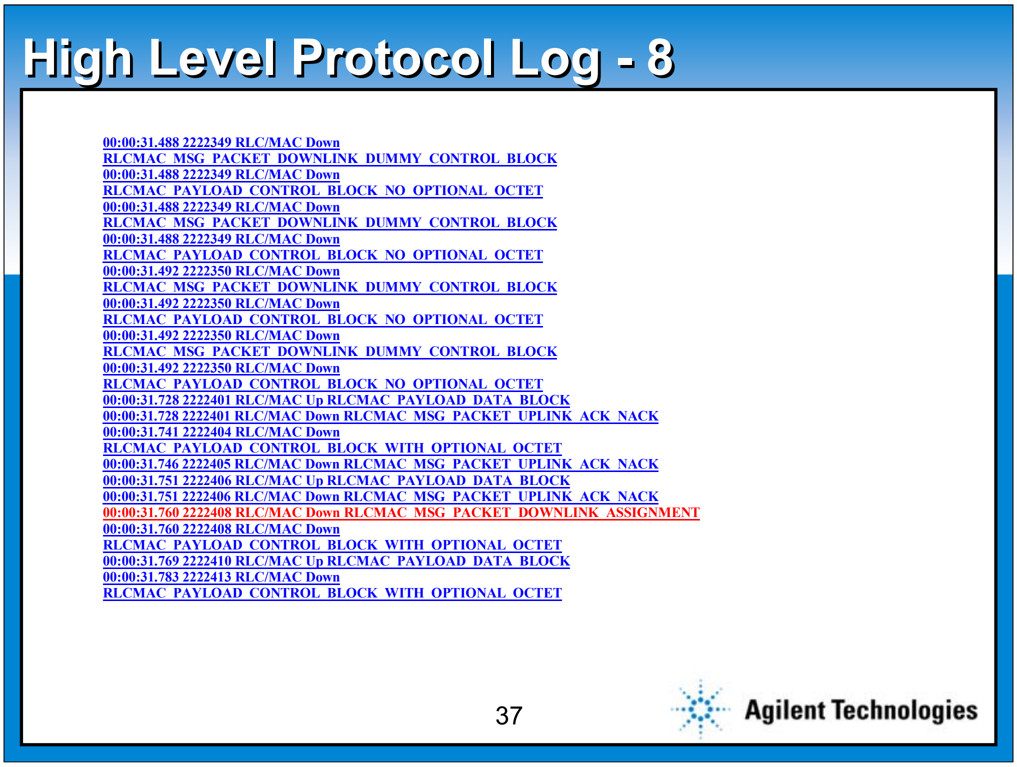
High Level Protocol Log - 7

```
00:00:02.713 2216114 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:02.732 2216118 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:02.732 2216118 RLC/MAC Up
RLCMAC_MSG_PACKET_CONTROL_ACKNOWLEDGEMENT
00:00:02.750 2216122 RLC/MAC Up
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:02.750 2216122 RLC/MAC Up
RLCMAC_MSG_PACKET_UPLINK_DUMMY_CONTROL_BLOCK
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:27.749 2221539 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:27.754 2221540 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:27.754 2221540 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:31.483 2222348 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:31.483 2222348 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
```

36

Agilent Technologies

Still more messages ...

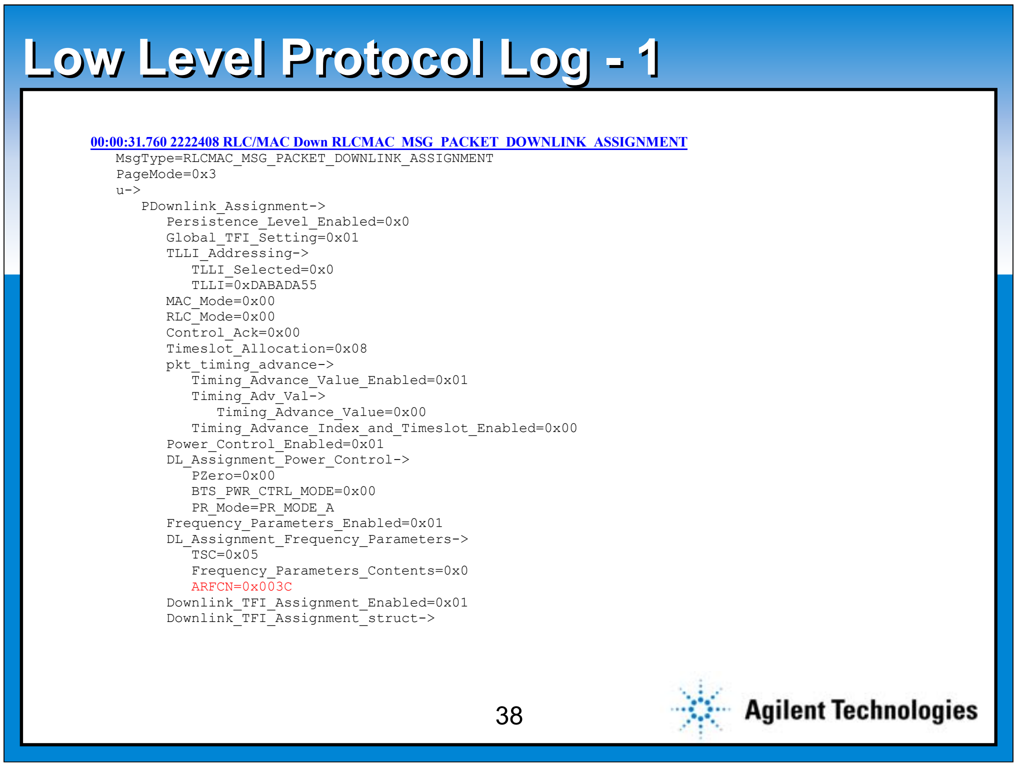


High Level Protocol Log - 8

```
00:00:31.488 2222349 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:31.488 2222349 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:31.488 2222349 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:31.488 2222349 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:31.492 2222350 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:31.492 2222350 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:31.492 2222350 RLC/MAC Down
RLCMAC_MSG_PACKET_DOWNLINK_DUMMY_CONTROL_BLOCK
00:00:31.492 2222350 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_NO_OPTIONAL_OCTET
00:00:31.728 2222401 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:31.728 2222401 RLC/MAC Down RLCMAC_MSG_PACKET_UPLINK_ACK_NACK
00:00:31.741 2222404 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:31.746 2222405 RLC/MAC Down RLCMAC_MSG_PACKET_UPLINK_ACK_NACK
00:00:31.751 2222406 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:31.751 2222406 RLC/MAC Down RLCMAC_MSG_PACKET_UPLINK_ACK_NACK
00:00:31.760 2222408 RLC/MAC Down RLCMAC_MSG_PACKET_DOWNLINK_ASSIGNMENT
00:00:31.760 2222408 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
00:00:31.769 2222410 RLC/MAC Up RLCMAC_PAYLOAD_DATA_BLOCK
00:00:31.783 2222413 RLC/MAC Down
RLCMAC_PAYLOAD_CONTROL_BLOCK_WITH_OPTIONAL_OCTET
```

Agilent Technologies

Finally, the message of interest happened; I have colored it red for this presentation. It contains the Assignment of a traffic channel to the handset, in this case the new channel that looked like a handover.



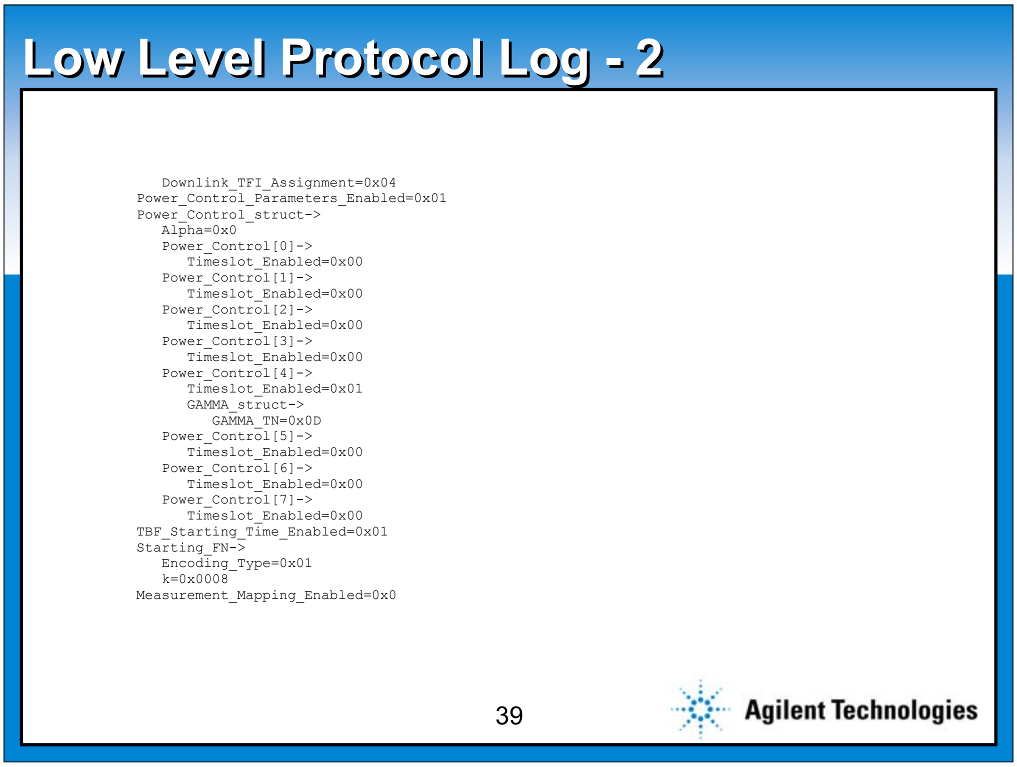
Low Level Protocol Log - 1

```
00:00:31.760 2222408 RLC/MAC Down RLCMAC_MSG_PACKET_DOWNLINK_ASSIGNMENT
   MsgType=RLCMAC_MSG_PACKET_DOWNLINK_ASSIGNMENT
   PageMode=0x3
   u->
      PDownlink_Assignment->
         Persistence_Level_Enabled=0x0
         Global_TFI_Setting=0x01
         TLLI_Addressing->
            TLLI_Selected=0x0
            TLLI=0xDABADA55
         MAC_Mode=0x00
         RLC_Mode=0x00
         Control_Ack=0x00
         Timeslot_Allocation=0x08
         pkt_timing_advance->
            Timing_Advance_Value_Enabled=0x01
            Timing_Adv_Val->
               Timing_Advance_Value=0x00
            Timing_Advance_Index_and_Timeslot_Enabled=0x00
         Power_Control_Enabled=0x01
         DL_Assignment_Power_Control->
            PZero=0x00
            BTS_PWR_CTRL_MODE=0x00
            PR_Mode=PR_MODE_A
         Frequency_Parameters_Enabled=0x01
         DL_Assignment_Frequency_Parameters->
            TSC=0x05
            Frequency_Parameters_Contents=0x0
            ARFCN=0x003C
         Downlink_TFI_Assignment_Enabled=0x01
         Downlink_TFI_Assignment_struct->
```

Agilent Technologies

Looking at this message in the low level, we see the field of interest, again made red by me. It tells the handset to use the Absolute Radio Frequency Channel Number (ARFCN) number 60. Note that 60 in decimal is 3 x 16 + 12, or 0x03C in hex.

It has taken this many messages for the test set to finally send the phone to the traffic channel.

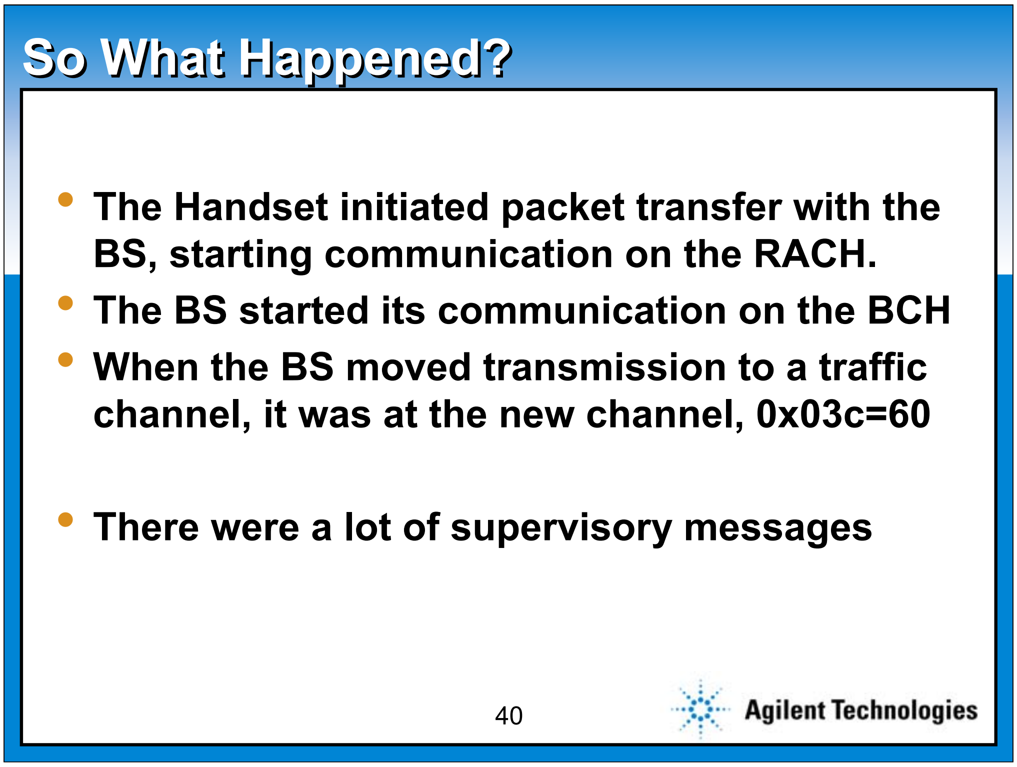


Low Level Protocol Log - 2

```
   Downlink_TFI_Assignment=0x04
Power_Control_Parameters_Enabled=0x01
Power_Control_struct->
   Alpha=0x0
   Power_Control[0]->
      Timeslot_Enabled=0x00
   Power_Control[1]->
      Timeslot_Enabled=0x00
   Power_Control[2]->
      Timeslot_Enabled=0x00
   Power_Control[3]->
      Timeslot_Enabled=0x00
   Power_Control[4]->
      Timeslot_Enabled=0x01
      GAMMA_struct->
         GAMMA_TN=0x0D
   Power_Control[5]->
      Timeslot_Enabled=0x00
   Power_Control[6]->
      Timeslot_Enabled=0x00
   Power_Control[7]->
      Timeslot_Enabled=0x00
TBF_Starting_Time_Enabled=0x01
Starting_FN->
   Encoding_Type=0x01
   k=0x0008
Measurement_Mapping_Enabled=0x0
```

Agilent Technologies

This is added just to show how long this message is, it is a continuation of the prior slide.

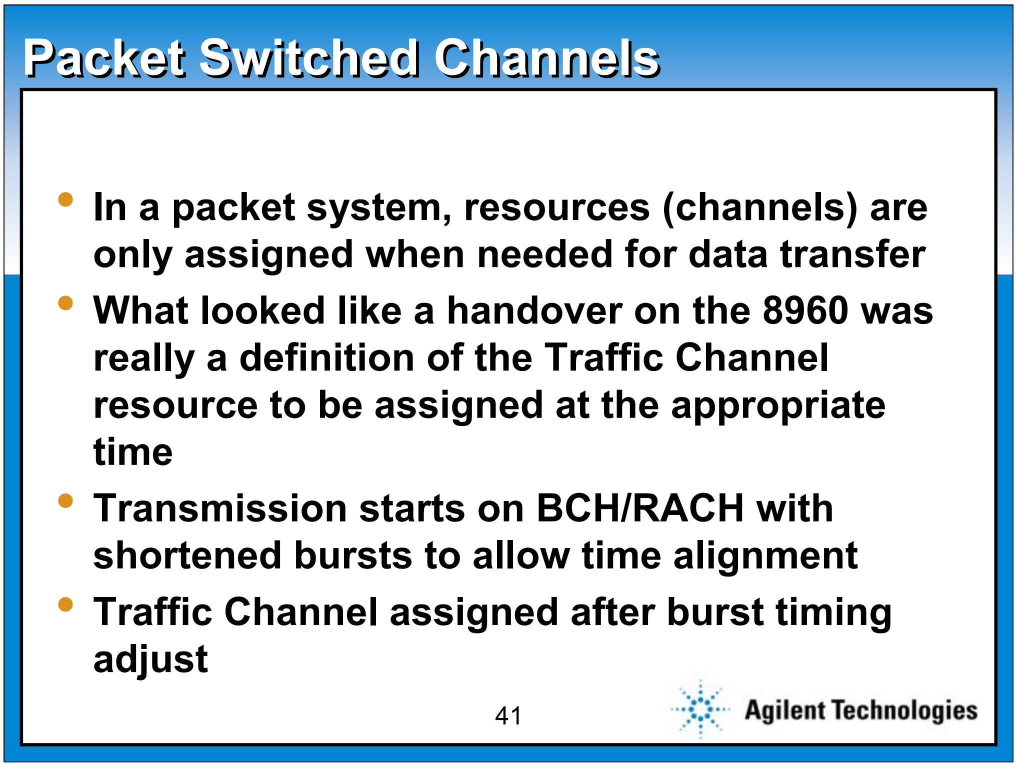


So What Happened?

- The Handset initiated packet transfer with the BS, starting communication on the RACH.
- The BS started its communication on the BCH
- When the BS moved transmission to a traffic channel, it was at the new channel, 0x03c=60
- There were a lot of supervisory messages

Agilent Technologies

In packet data systems, the network is always on. Resources are only used when needed for packet data transfer. The network does not know where the phone is relative to the cell when the packet transfer initiates, so it must perform time alignment of the transmit time of the phone. The phone starts transmission with shortened bursts, and uses the Random Access Channel (RACH) to carry its messages. The network sends its messages on the Broadcast Control Channel (BCH). After a lot of supervisory messages, the phone was assigned to an available traffic channel of the cell, on channel 60. It will use the number of slots and coding as configured on the test set.

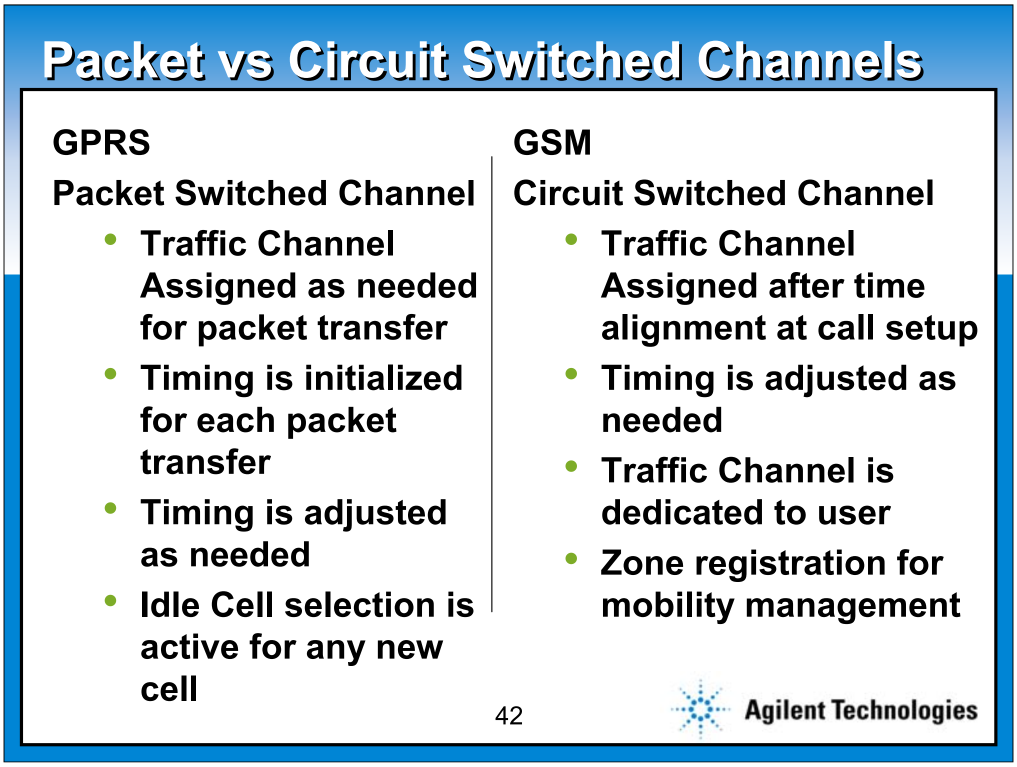


Packet Switched Channels

- In a packet system, resources (channels) are only assigned when needed for data transfer
- What looked like a handover on the 8960 was really a definition of the Traffic Channel resource to be assigned at the appropriate time
- Transmission starts on BCH/RACH with shortened bursts to allow time alignment
- Traffic Channel assigned after burst timing adjust

Agilent Technologies

The network is allowed to have latency during packet delivery. Just because data is available for a user, there is no guarantee that data will be sent on a slot of every frame. Instead, other users may be assigned that channel. The assignment of a traffic channel can change with every packet, though usually several packets would be sent on one channel.



Packet vs Circuit Switched Channels

GPRS

Packet Switched Channel

- Traffic Channel Assigned as needed for packet transfer
- Timing is initialized for each packet transfer
- Timing is adjusted as needed
- Idle Cell selection is active for any new cell

GSM

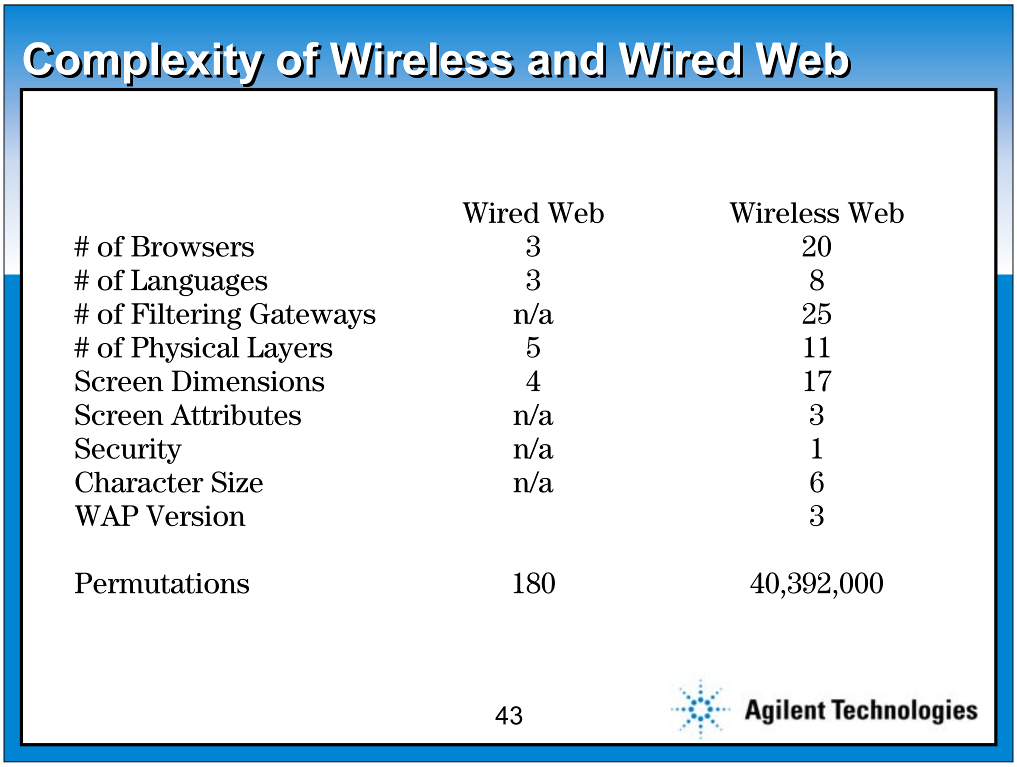
Circuit Switched Channel

- Traffic Channel Assigned after time alignment at call setup
- Timing is adjusted as needed
- Traffic Channel is dedicated to user
- Zone registration for mobility management

Agilent Technologies

The characteristics of packet and circuit switched connections are shown here. The big difference is in the assignment of resources. In packet systems, the resources are only assigned as needed, while in circuit switched applications, the channel is dedicated to one user, even if there is no data to be transferred.

Packet systems have much higher overhead to control the dynamic resource allocation, but use system resources much more efficiently because they are dynamic.



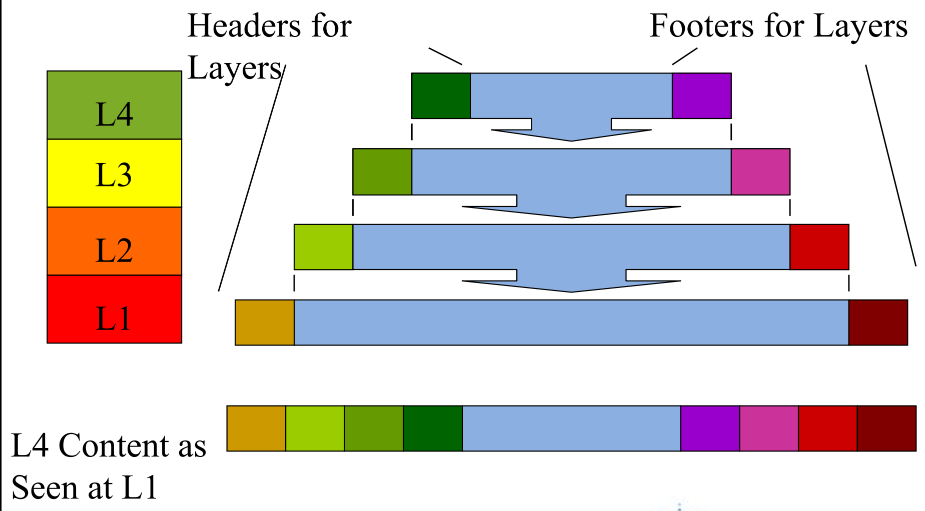
Complexity of Wireless and Wired Web

	Wired Web	Wireless Web
# of Browsers	3	20
# of Languages	3	8
# of Filtering Gateways	n/a	25
# of Physical Layers	5	11
Screen Dimensions	4	17
Screen Attributes	n/a	3
Security	n/a	1
Character Size	n/a	6
WAP Version		3
Permutations	180	40,392,000

Agilent Technologies

The wireless web has significantly more complexity to it than the wired web. If you consider all the possible combinations, the wired web only comes to 180, and changes typically are not made for screen size. On the wireless web, the number of combinations is over 40 million, though many of these combinations don't make sense.

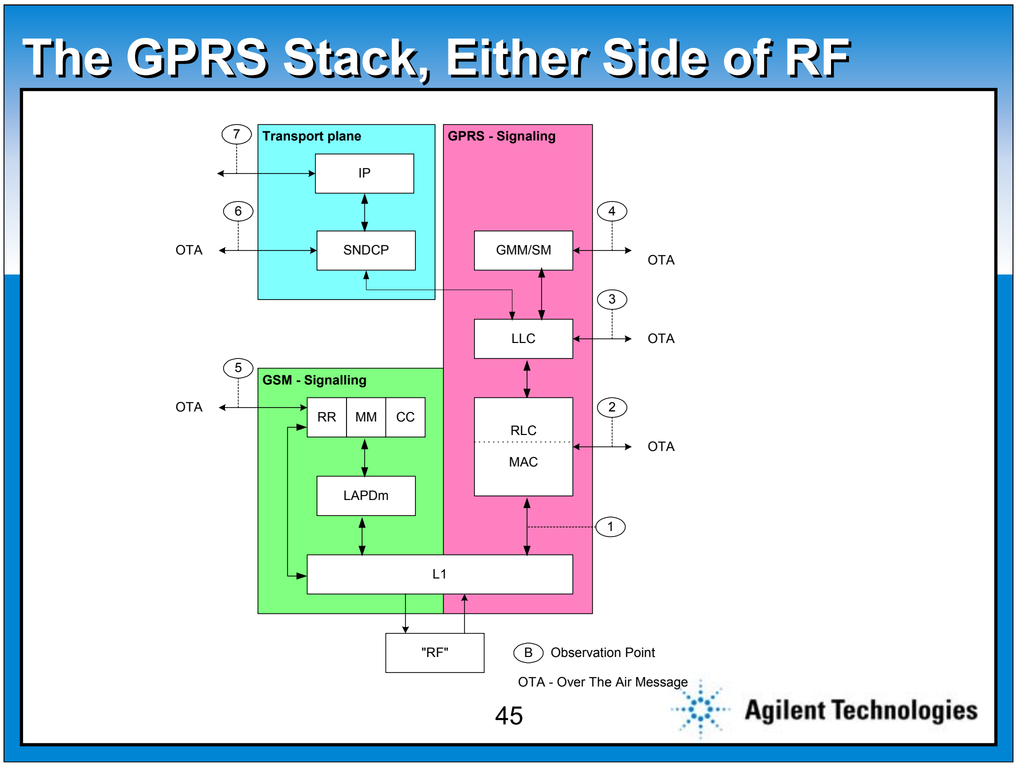
Layered Transport



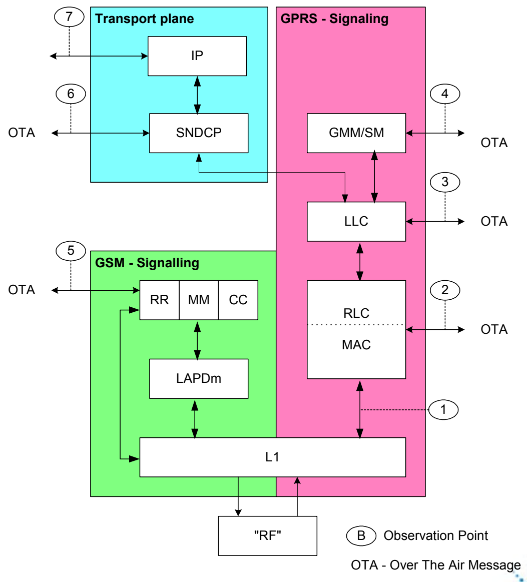
Agilent Technologies

This shows how a message generated at layer 4 will be transported down all the lower layers and transmitted across the physical layer. At the other side, it will be transported back up all the layers until it reaches layer 4, where the message will be properly identified as a layer 4 peer to peer communication.

In theory, it is possible to read all the messages at layer 1, and by properly interpreting all the headers and footers, determine the active layer. This is not the implementation used in the GPRS LA, however. We have actual taps at each layer.

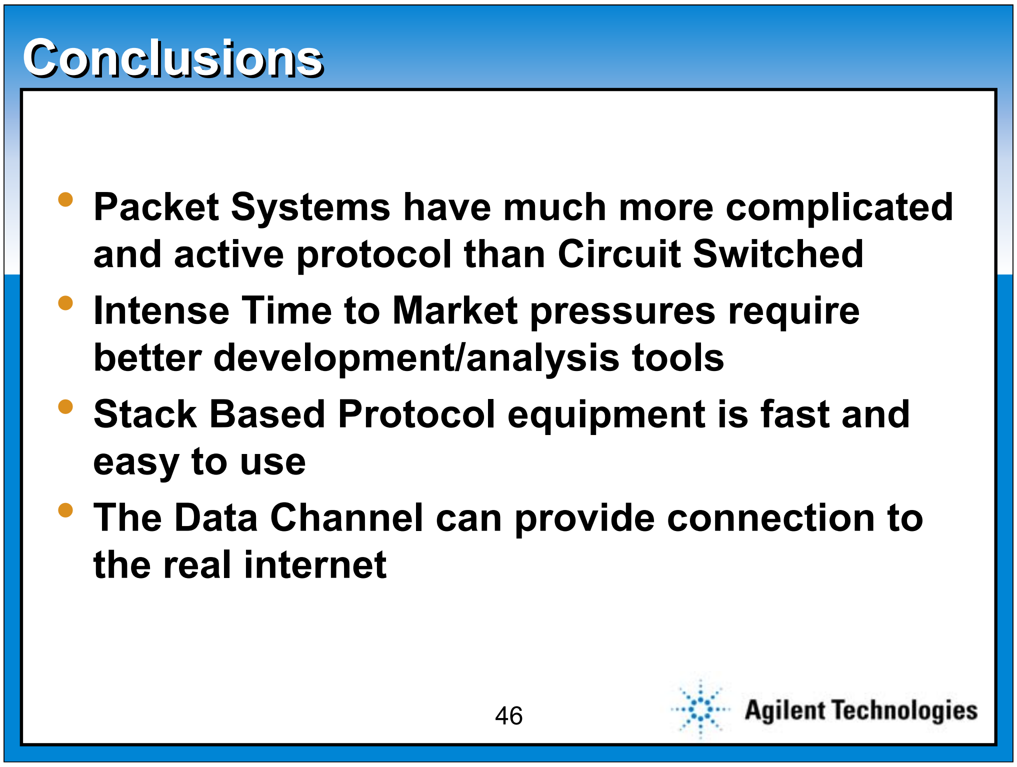


The GPRS Stack, Either Side of RF



Agilent Technologies

This is an alternate view of the GPRS protocol stack. Since the layers are symmetrical, this could be either the base station/network, or the handset. The difference between this and the prior layers slide is that this one shows both the transmission plane and the control plane. The green block is the GSM transmission plane, the pink is the control plane, and the blue is the GPRS transmission plane. Note that the lower layers of the GPRS transmission are handled by the control plane. The bubbles with numbers inside correspond to probe locations in the protocol logger. These will be discussed in more detail in the following slides.



Conclusions

- Packet Systems have much more complicated and active protocol than Circuit Switched
- Intense Time to Market pressures require better development/analysis tools
- Stack Based Protocol equipment is fast and easy to use
- The Data Channel can provide connection to the real internet

Agilent Technologies

The protocol associated with any Packet Data system are more complicated than those associated with a circuit switched connection. The needs of instrumentation in this area have given rise to new tool that can significantly reduce the time of debugging complex software.

The use of a stack based tool allows easier use without as much operator training as equipment that is based on a script.

The ability to connect to the internet to run any available application is significant in providing broad choice of content for evaluation.