RDP Test Suites Open Source & Updates

Windows Protocol Test Suite Developer Team

Vivian TIAN,   Huiren JIANG
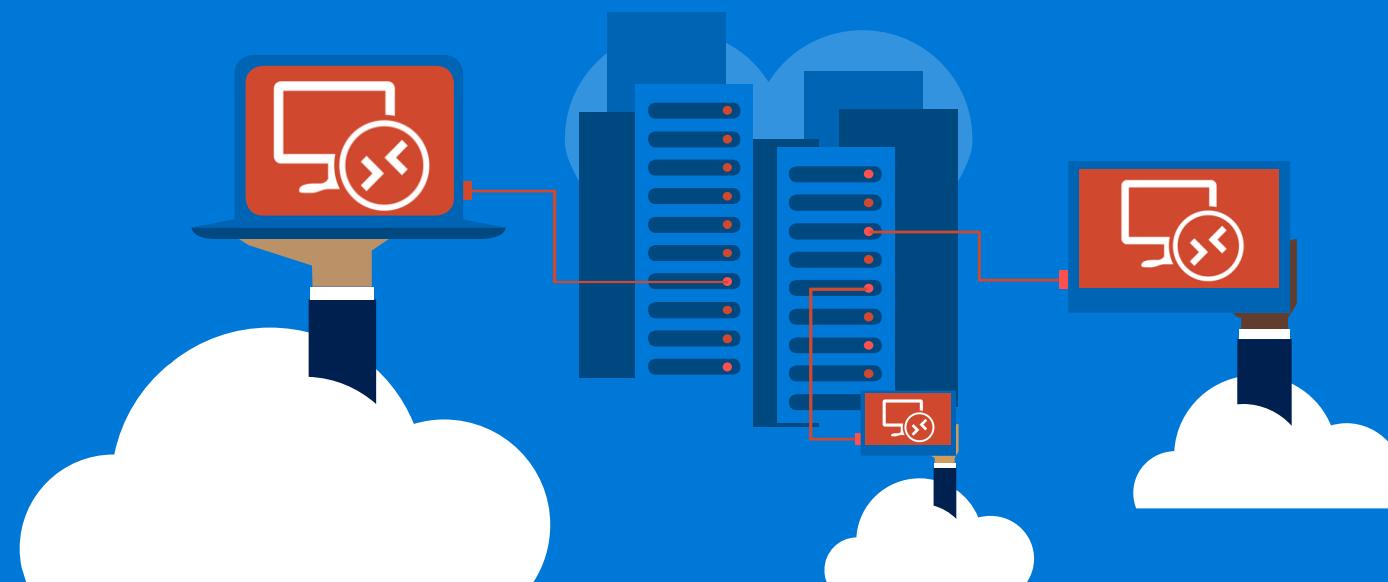
Agenda

Open Source Status

RDP Test Suites Overview

RDP Test Suites Updates

Demo

Open Source Status

- Open Source to GitHub in 2016

- MIT License

- 11 Test Suites(family)

- 45 Windows Protocols

- Protocol Test Framework

- Protocol Test Manager

- Utilities

https://github.com/Microsoft/WindowsProtocolTestSuites
https://github.com/Microsoft/protocoltestframework

Test Suite Name	Protocols Covered
RDP Client Family Test Suite	MS-RDPBCGR, MS-RDPDISP, MS-RDPEDYC, MS-RDPEGFX, MS-RDPEGT, MS-RDPEI, MS-RDPEMT, MS-RDPEUDP, MS-RDPEUSB, MS-RDPEVOR,MS-RDPRFX
RDP Server Test Suite	MS-RDPBCGR
File Server Family Test Suite	MS-SMB2, MS-DFSC, MS-SWN, MS-FSRVP, MS-FSA, MS-RSVD, MS-SQOS, MS-HVRS
AD Family Test Suite	MS-ADTS, MS-ADA1, MS-ADA2, MS-ADA3, MS-ADLS, MS-ADSC, MS-APDS, MS-DRSR, MS-FRS2, MS-LSAD, MS-LSAT, MS-NRPC, MS-SAMR
Kerberos Test Suite	MS-PAC, MS-KILE, MS-KKDPC
SMB1 Server Test Suite	MS-SMB
Branch Cache Test Suite	PCCRC, PCCRTP, PCHC, PCCRR
MS-SMBD Test Suite	MS-SMBD
MS-AZOD Test Suite	MS-AZOD
BYOD Test Suite	MS-ADFSPIP
MS-ADOD Test Suite	MS-ADOD
Protocol Test Framework	N/A
Protocol Test Manager	N/A
RDP Protocol Based Control Agent	N/A

Open Source Status

- Download count increased by 118% last year
- 34/41 issues closed
- 5 pull requests from external partners
- 6 releases since 2016

GitHub Download Count

701	FY17Q2
1109	FY17Q3
1442	FY17Q4
1754	FY18Q1
2232	FY18Q2

Jan 2016
Kickoff
Windows Protocol Test Suites Open Source Project

May 2016
1st Release
Protocol Test Framework
1.0.5000.0

Sep 2016
2nd Release
Windows Protocol Test Suites
1.0.5000.0

May 2017
3rd Release
Protocol Test Framework
1.0.5500.0

Jun 2017
4th Release
Windows Protocol Test Suites
2.0.0.0

Aug 2017
5th Release
Windows Protocol Test Suites
2.0.66.0

Feb 2018
6th Release
Windows Protocol Test Suites
3.18.3.0

Major.Year.Month.Minor

Build an Active Community

Report Issues
- Report issues
- Bug fix
- Verify fix

Update Documentation
- User Guide
- Test Design Specification

New Features
- Log an issue before working on the feature

New Test Cases
- Add new test cases
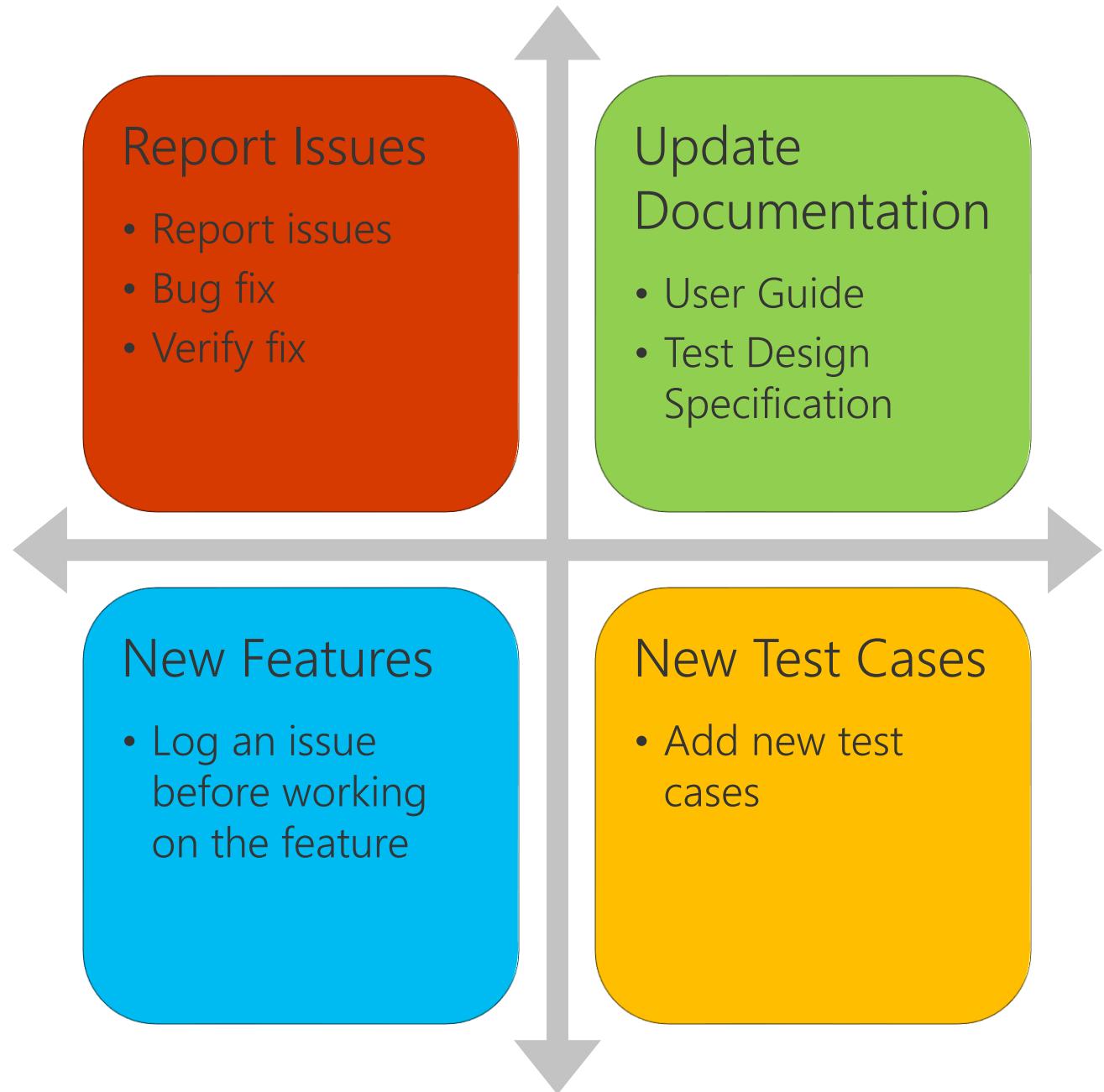
Agenda

Open Source Status

RDP Test Suites Overview

RDP Test Suites Updates

Demo

RDP Test Suites Overview

RDP Client Test Suite Family

- 11 RDP protocols
- System Under Testing (SUT) is RDP client implementation:
  - mstsc.exe, FreeRDP, Thin Client ...

RDP Server Test Suite

- MS-RDPBCGR
- System Under Testing (SUT): RDP Server Implementation

Microsoft

RDP Client Test Methodology

Client/Server Side Partner Implementation
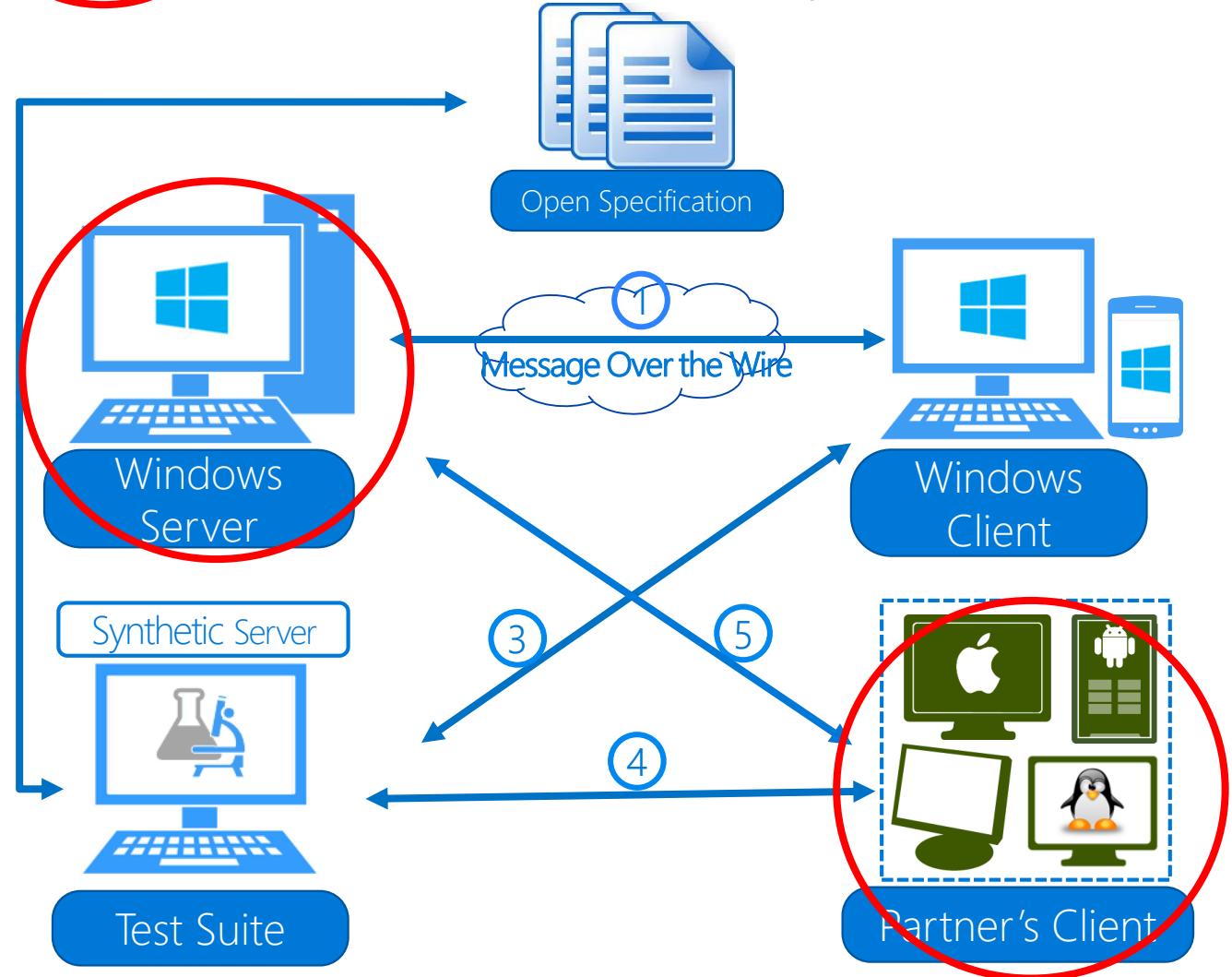
1. Open Specification defines messages, sequence, behavior

2. Develop protocol test suite (synthetic server) according to Open Specification

3. Run against Windows to verify Open Specification

4. Run against partner's Client Implementation to help identify & debug issues

5. Interop between partner's Client and Windows Server

RDP Client Test Suite Family Scope

MS-RDPBCGR
- Connection
- Reactivation
- Auto-Reconnection
- Server Redirection
- Basic Client Input
- Fast-Path/Slow-Path
- Static virtual Channel
- Network Auto-Detection (Win8)
- Connection health monitoring (Win8.1)

MS-RDPRFX
- RFX Codec (RLGR1/RLGR3)
- Image / Video Mode

MS-RDPEUSB
- USB Device Channel Setup
- I/O

MS-RDPEVOR
- Video Remoting

MS-RDPEGFX
- Surface & Cache
- RemoteFX Codec
- Progressive Codec
- Clear Codec
- RDP8.0 Compression
- H264 Codec

MS-RDPEUDP
- Reliable/Lossy Connection
- Retransmit
- Congestion Control

MS-RDPEMT
- Build Secure Channel
- Network Auto-Detection
- Exchange DVC data

MS-RDPEI
- Touch Input
- Touch Control

MS-RDPEDISP
- Change display configuration

Microsoft

RDP Client Protocol Test Suite Family Scope

Test Target: RDP Client	Synthetic RDP Server	Align with Windows 10 documents	Test Code (C#)
➢ mstsc.exe ➢ FreeRDP ➢ Thin Client ➢ ……	➢ Test Suite as synthetic RDP server	➢ 11 common protocols ➢ 400+ test cases ➢ Positive and Negative	➢ RemoteFX codec ➢ Clear Codec ➢ Progressive Codec ➢ ……

Microsoft

RDP Client Test Suite Family Architecture
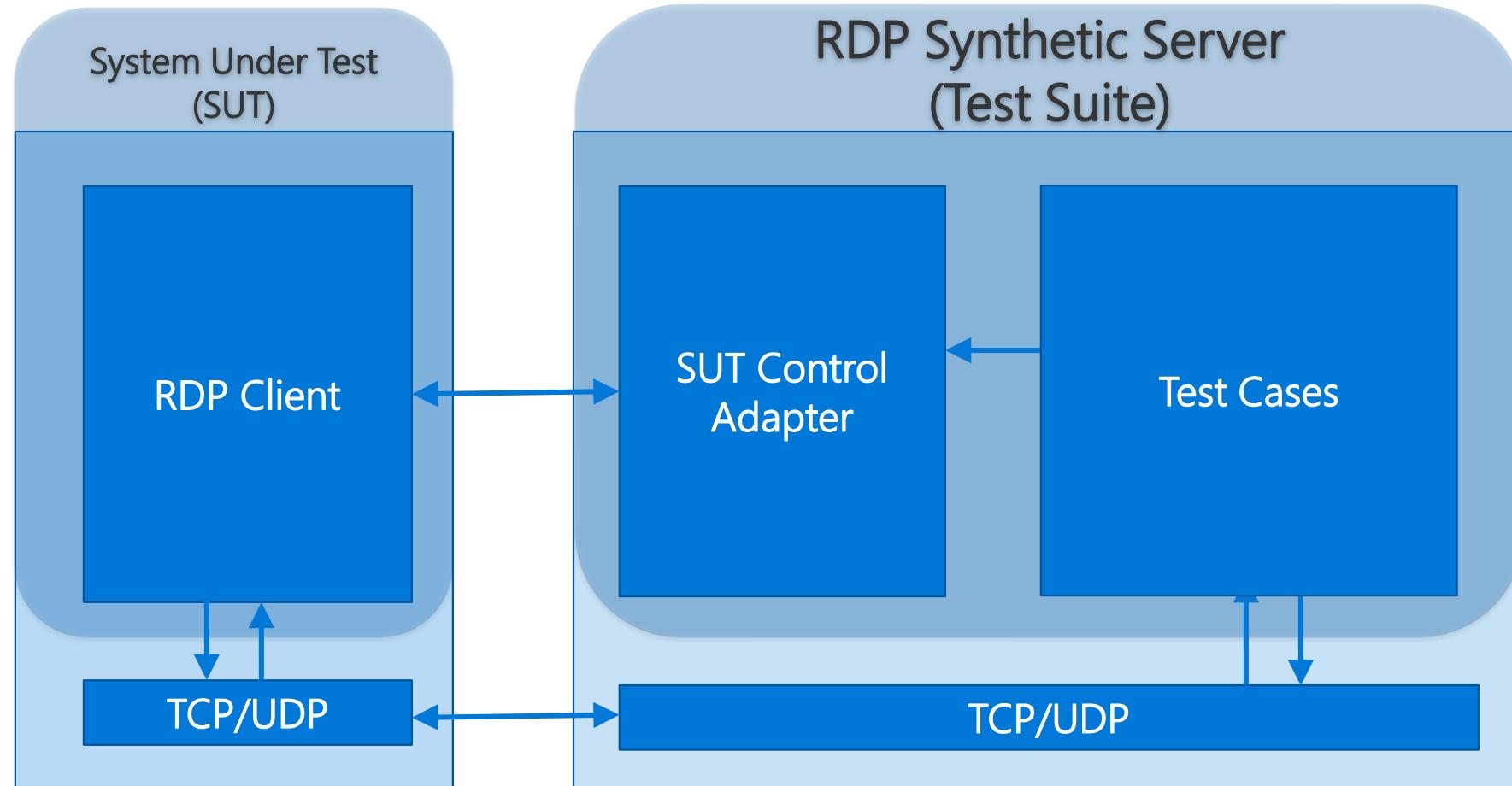
SUT Control Adapter – Protocol-based

- Self-defined a simple **protocol** to transfer control commands and result.
- User needs to implement a simple program as an **Agent** to control SUT (System Under Test)

Test Cases

RDP Protocols

Call Method

SUT Remote Control Protocol

SUT control adapter

SUT Control Request Message

SUT Control Response Message

RDP Client

Operate

Agent

Test Suite

SUT

Advantage:
- Automation on any system
- No work on test suite
- Agent is a program: flexible and powerful

Disadvantage:
- Need to develop an agent, **but workload is low**
  - ➢ Quite simple protocol
  - ➢ Familiar environment

12

Microsoft

SUT Control Adapter – Protocol-based

- Self-defined Protocol
- C# and Java implementation

· https://github.com/Microsoft/WindowsProtocolTestSuites/tree/staging/TestSuites/RDP/RDPSUTControlAgent

Command	Value	Meaning
START_RDP_CONNECTION	0x0001	Trigger SUT to start a RDP connection
CLOSE_RDP_CONNECTION	0x0002	Trigger SUT to close all RDP connection
AUTO_RECONNECT	0x0003	Trigger SUT to start an auto reconnect
BASIC_INPUT	0x0004	Trigger SUT to do basic input on RDP client
SCREEN_SHOT	0x0005	Do a screen shot on RDP client and send graphic data back
TOUCH_EVENT_SINGLE	0x0101	Ask SUT to trigger single touch event on RDP client
TOUCH_EVENT_MULTIPLE	0x0102	Ask SUT to trigger multiple touch event on RDP client
TOUCH_EVENT_DISMISS_HOVERING_CONTACT	0x0103	Ask SUT to trigger a dismiss hovering contact event on RDP client
DISPLAY_UPDATE_RESOLUTION	0x0201	Ask SUT to trigger RDPEDISP message to update display resolution
DISPLAY_UPDATE_MONITORS	0x0202	Ask SUT to trigger RDPEDISP message to update monitor setting, such as add a monitor, remove a monitor or change monitor position
DISPLAY_FULLSCREEN	0x0203	Ask SUT to change RDP client to full screen.

RDP Client Test Case Design

If RDP server sends message A to RDP client, RDP client should respond with message B.

If A is invalid, client should return error message C.

Test Case 1:
1. Establish RDP connection;
2. Server.Send(*msgA*);
3. Expect client respond *msgB, and*
   - *Assert (msgB.Field1 == valid1)*
   - *Assert (msgB.Field2 == valid2)*
   - *...*
   - *Assert (msgB.FieldN == validN)*

Test Case 2:
1. Establish RDP connection;
2. Server.Send(*Invalid msgA*);
3. Expect client respond *msgC, and*
   - *Assert (msgC.Satus == ERROR)*

Microsoft

RDP Client Test Case Design: Graphic output verification

Problem

- In most cases, codec issues cannot be observed on the wire

Encode

Send by RDP protocols

Decode

Decode

Display

Screen Shot

Expected

Screen Shot

Transfer by SUT Control Adapter

Compare

Test Suite

Screen Shot

SUT
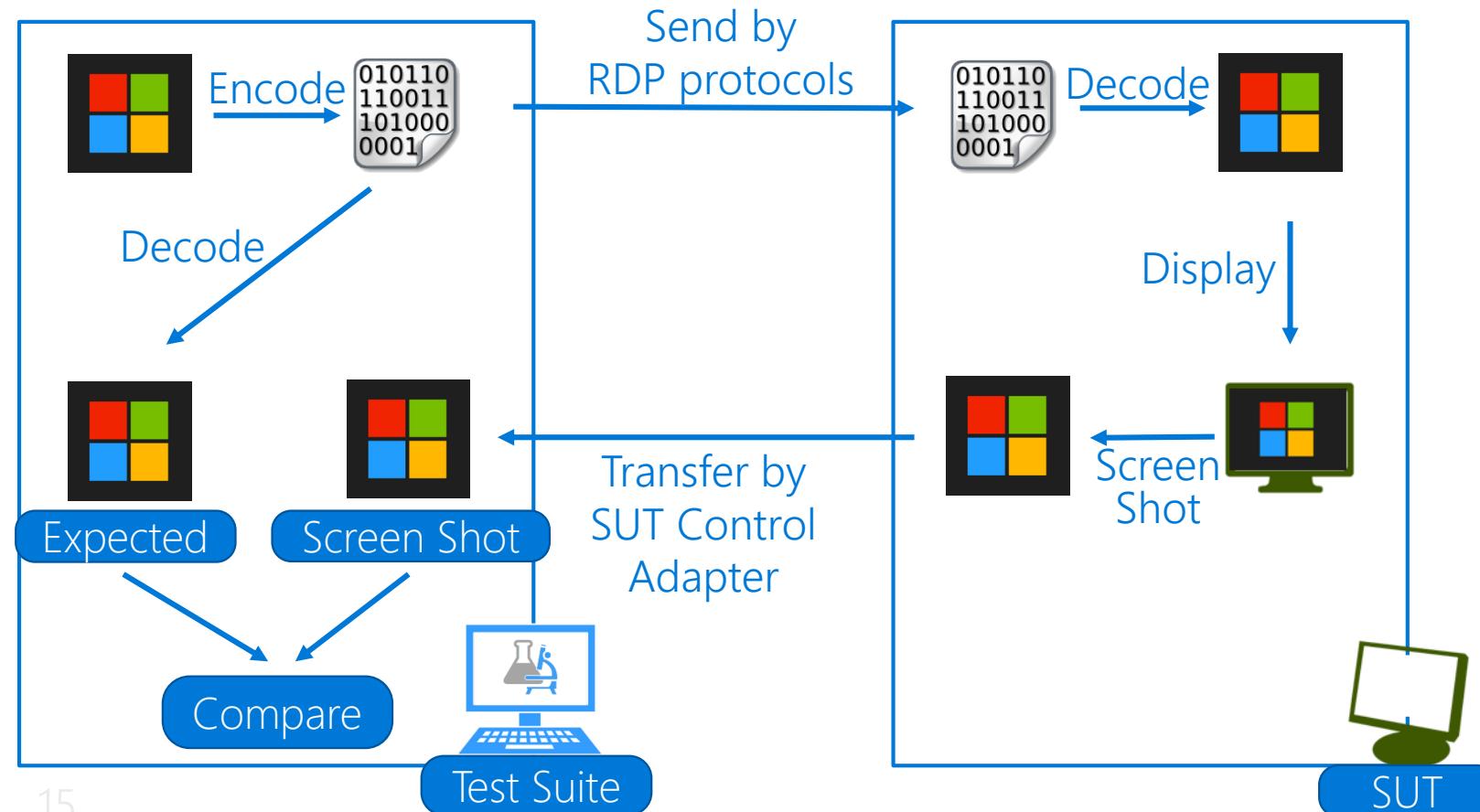
How to compare:

- Exactly the same, clear codec
- Look Similar, lossy codec
  - ➤ Image quality assessment(IQA) algorithm: MS-SSIM

Microsoft

RDP Client Test Suite Family Deployment

System Under Test

Name   RDP-Client

IP Add   192.168.0.1

Subnet Mask 255.255.255.0

Network Switch

Driver Computer

Name   RDP-Server

IP Add   192.168.0.2

Subnet Mask 255.255.255.0

RDP Client Test Suite Family
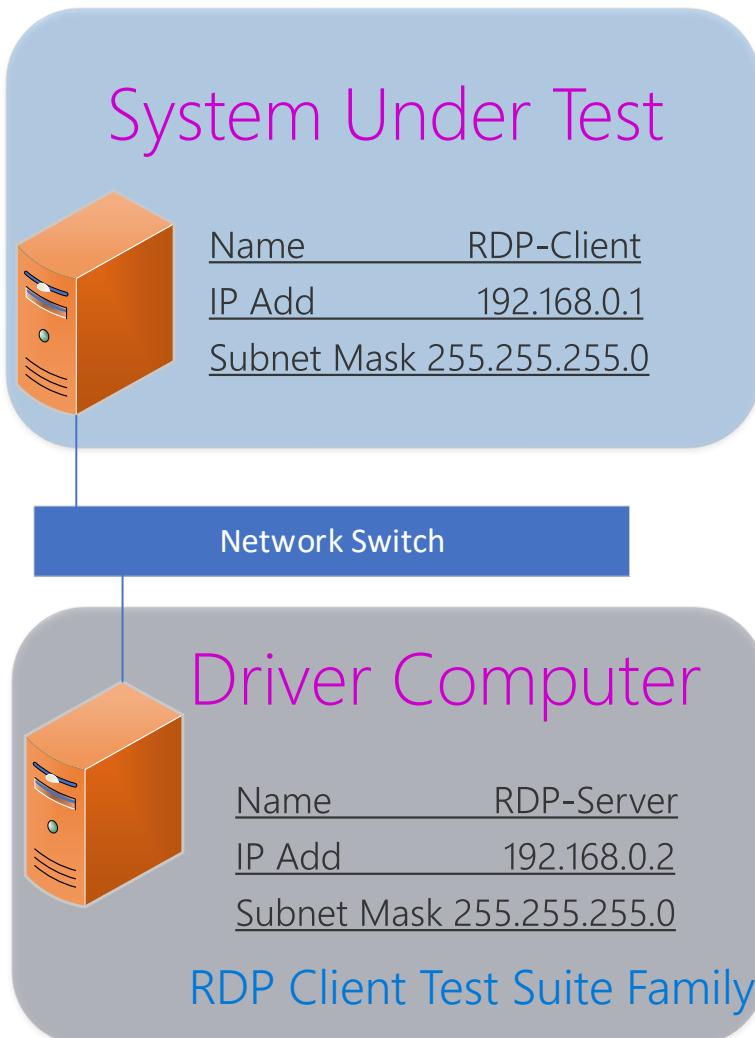
Machine role	Software/service
System Under Test (SUT)	RDP Client Implementation
Driver Computer	Microsoft® Visual Studio® 2012 Ultimate Protocol Test Framework RDP Client Test Suite Family (Synthetic RDP Server)

Common Deployment Steps:

1. Deployment RDP client on SUT
2. Install Test Suite on Driver Computer
3. Configure Test Suite on Driver Computer

Microsoft

How to Run Test Suite

1. Batch File
2. VS
3. PTM
   - GUI
   - Cmd line

RDP Server Test Suite

System Under Test

Name   RDP-Server
IP Add   192.168.0.1
Subnet Mask 255.255.255.0

Network Switch

Driver Computer

Name   RDP-Client
IP Add   192.168.0.2
Subnet Mask 255.255.255.0

RDP Server Test Suite

MS-RDPBCGR

Connection
AutoReconnect
Input
Output
StaticVirtualChannel
AutoDetect
MultitransportBootstrapping
HealthMonitoring

Machine role	Software/service
System Under Test (SUT)	RDP Server Implementation
Driver Computer	Microsoft® Visual Studio® 2012 Ultimate Protocol Test Framework RDP Server Test Suite (Synthetic RDP Client)

Microsoft

Agenda

Open Source Status

RDP Test Suites Overview

RDP Test Suites Updates

Demo

Updates to RDP Client Test Suite Family

- Support RDP version 10.4
- Support AVC444v2
- Support RDSTLS
- Update batch files
- Fix known issues from partners
- MMA updates to RDP parsers(1.44)

AVC444v2 Bitmap Stream Format

- New format for RDPGFX_WIRE_TO_SURFACE_PDU_1 command
- Introduced in RDP 10.3 and TD v20170316
- Similar to AVC444...
  - Pack one image frame into two YUV420p frames
  - Apply Transforming at macroblock level
  - Main view keeps the same
- What's different...
  - A new codec ID RDPGFX_CODECID_AVC444V2(0x000F)
  - auxiliary view changes

AVC444v2 Encoding Process at Server

1. Original image is transformed into two views.

Original
Image
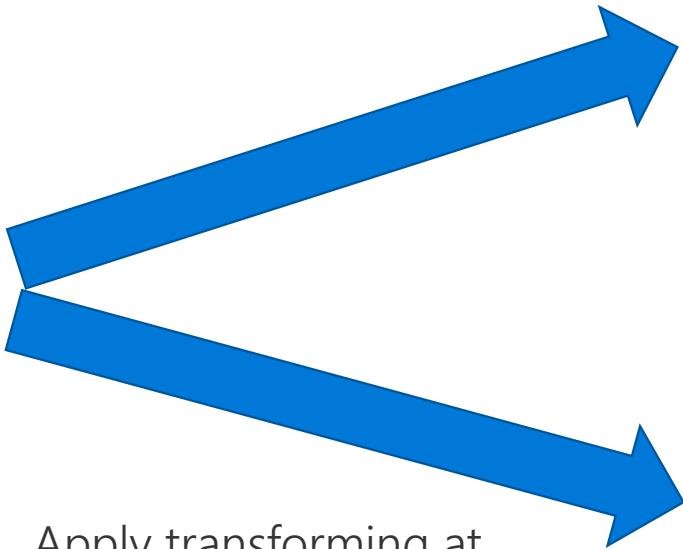
Apply transforming at
macroblock level

Main View

Auxiliary View

AVC444v2 Encoding Process at Server

2. Encoded two views into bit streams utilizing H.264 encoder.
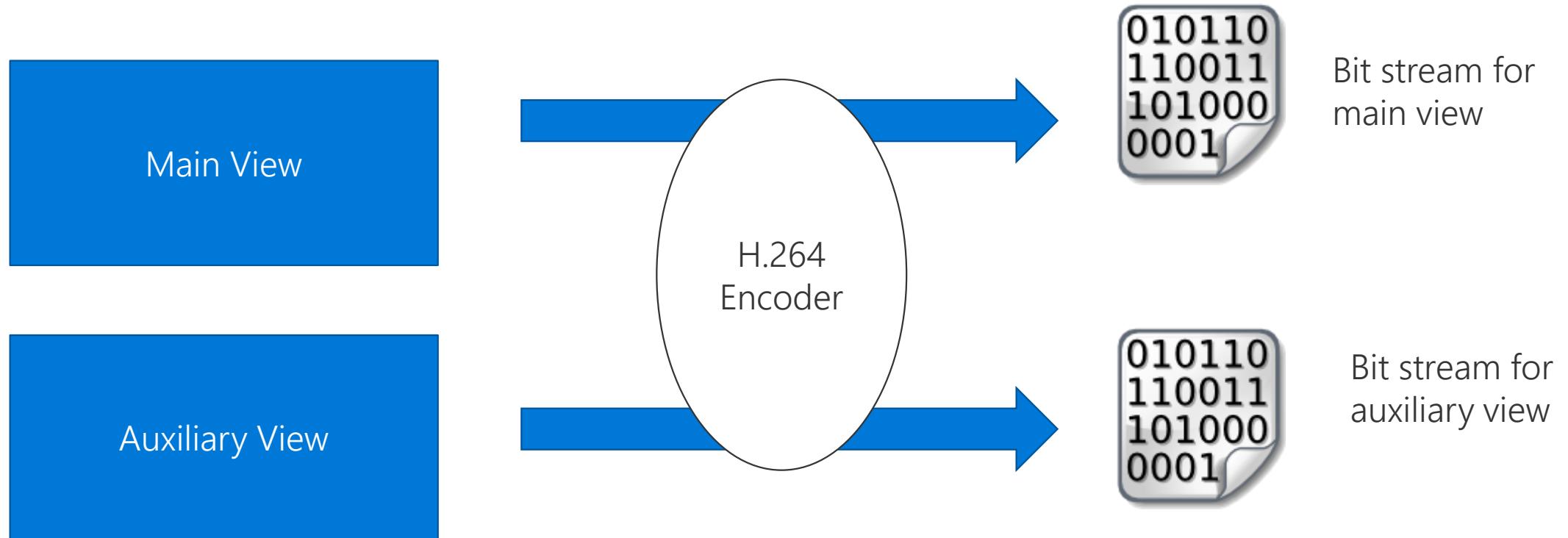
AVC444v2 Encoding Process at Server

3. Pack two bit streams into RDPGFX_WIRE_TO_SURFACE_PDU_1.

Bit stream for
main view

Bit stream for
auxiliary view

RDPGFX_CODECID_AVC444V2

avc420EncodedBitstream1

avc420EncodedBitstream2

RDPGFX_WIRE_TO_SURFACE_PDU_1
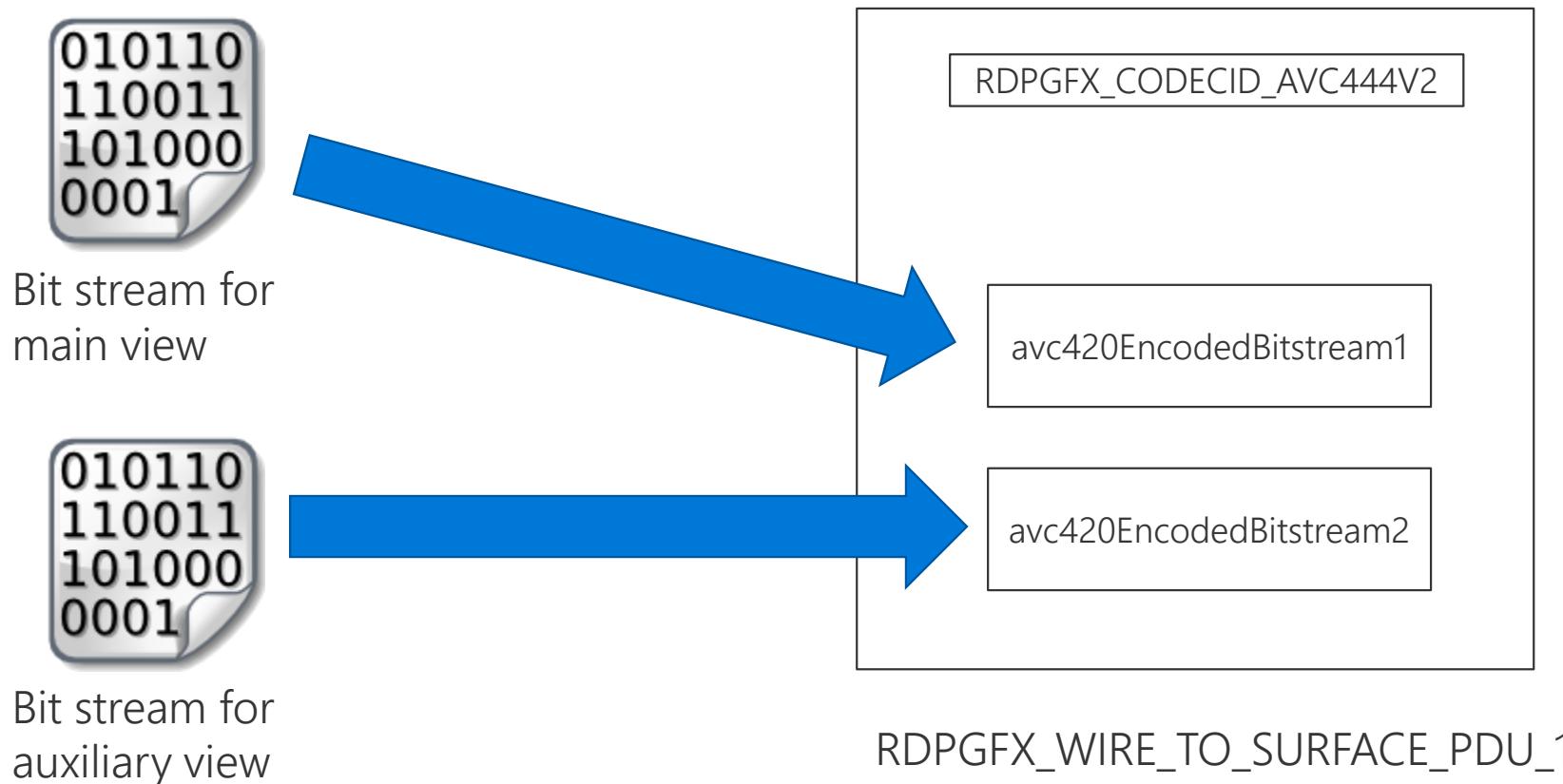
AVC444v2 Encoding Process at Server

4. Send PDU to terminal client through RDPEGFX channel.

RDPGFX_WIRE_TO_SURFACE_PDU_1
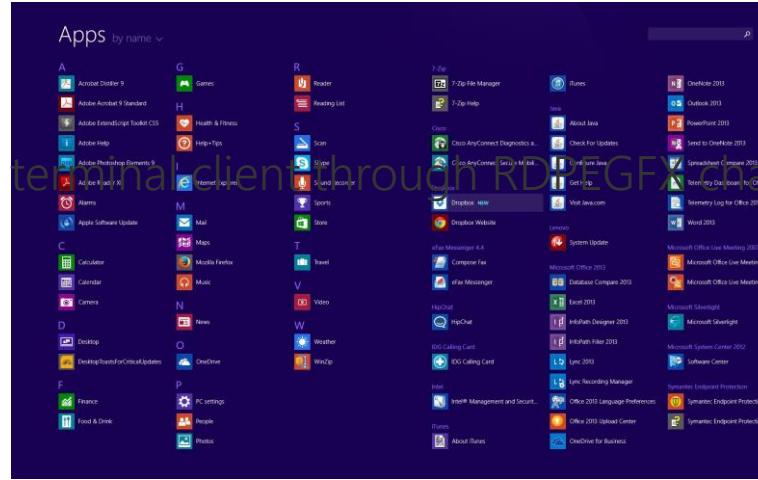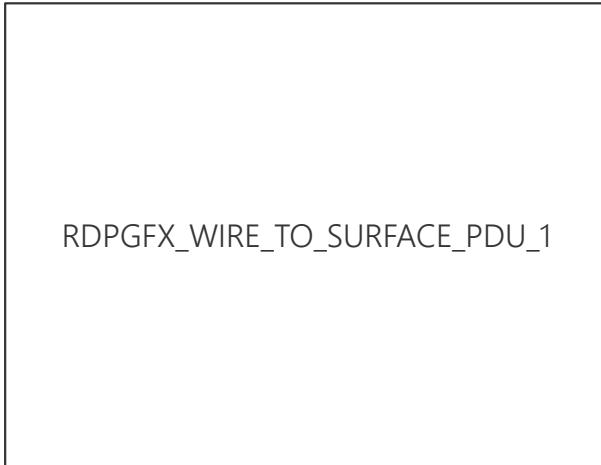
Transforming Detail

1. Original image is split into 16x16 macroblocks.

mb0   mb1   mb2   ...
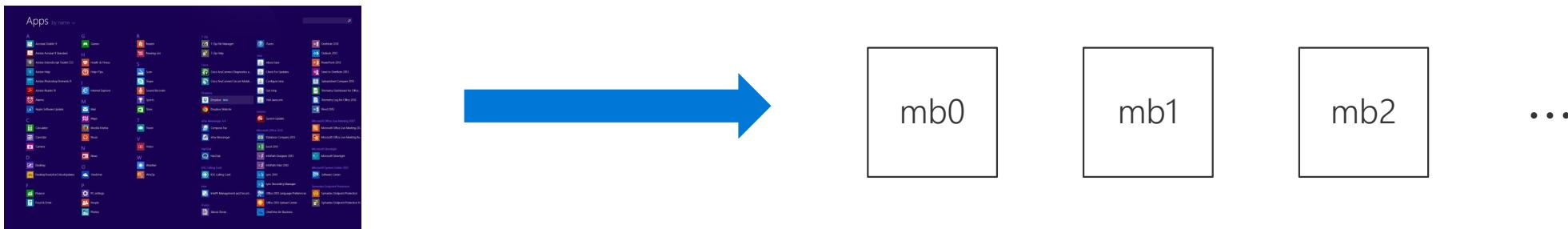
Transforming Detail

2. For each macroblock, do color conversion to represent into YUV plane.

Y_{444}

mb0 →

U_{444}

V_{444}
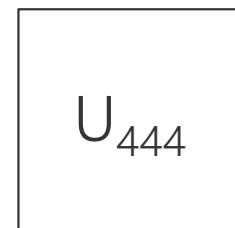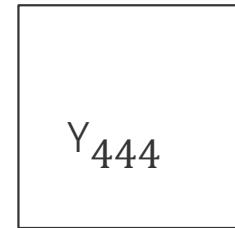
Transforming Detail

2. Map pixels

Y_{444}

U_{444}

V_{444}

Y_{420}

U

V
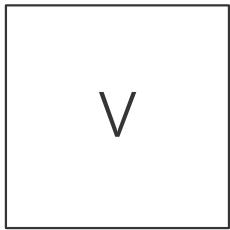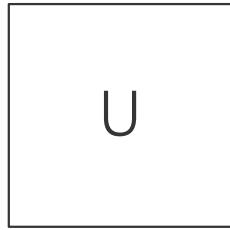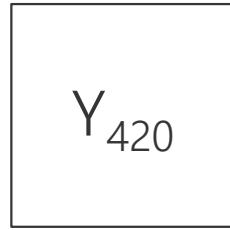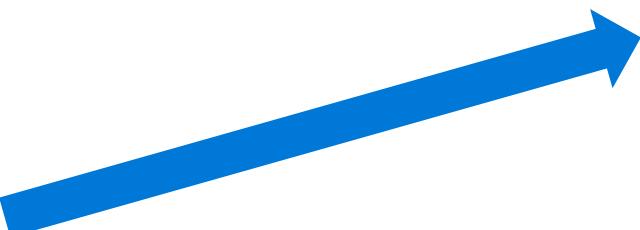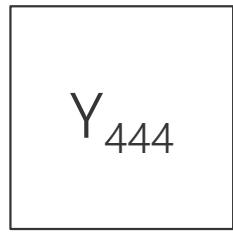
# Auxiliary view of AVC444 for each macroblock

Y Frame

U Frame

V Frame

Original Image
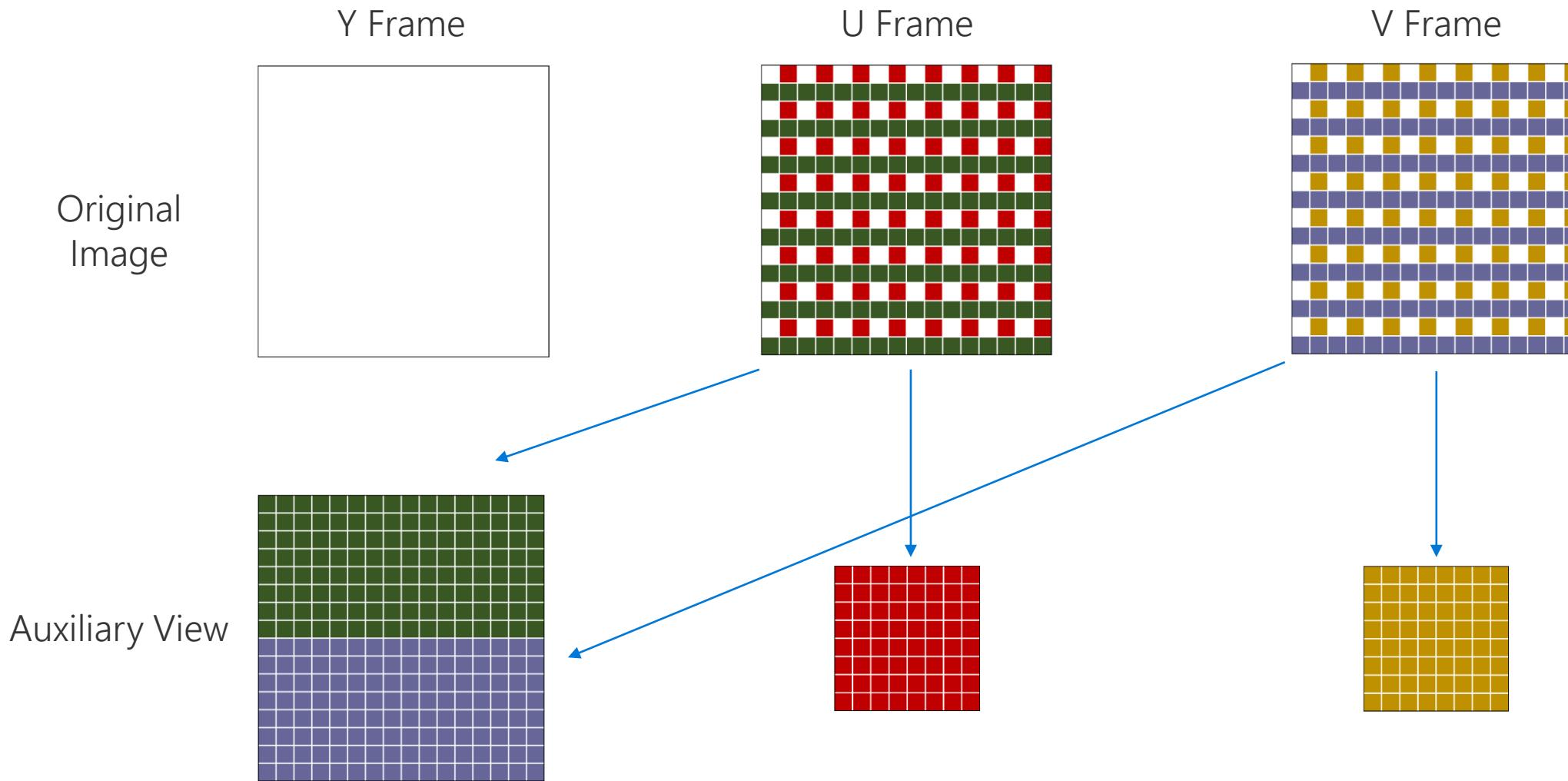
Auxiliary View

Auxiliary view of AVC444v2 for each macroblock
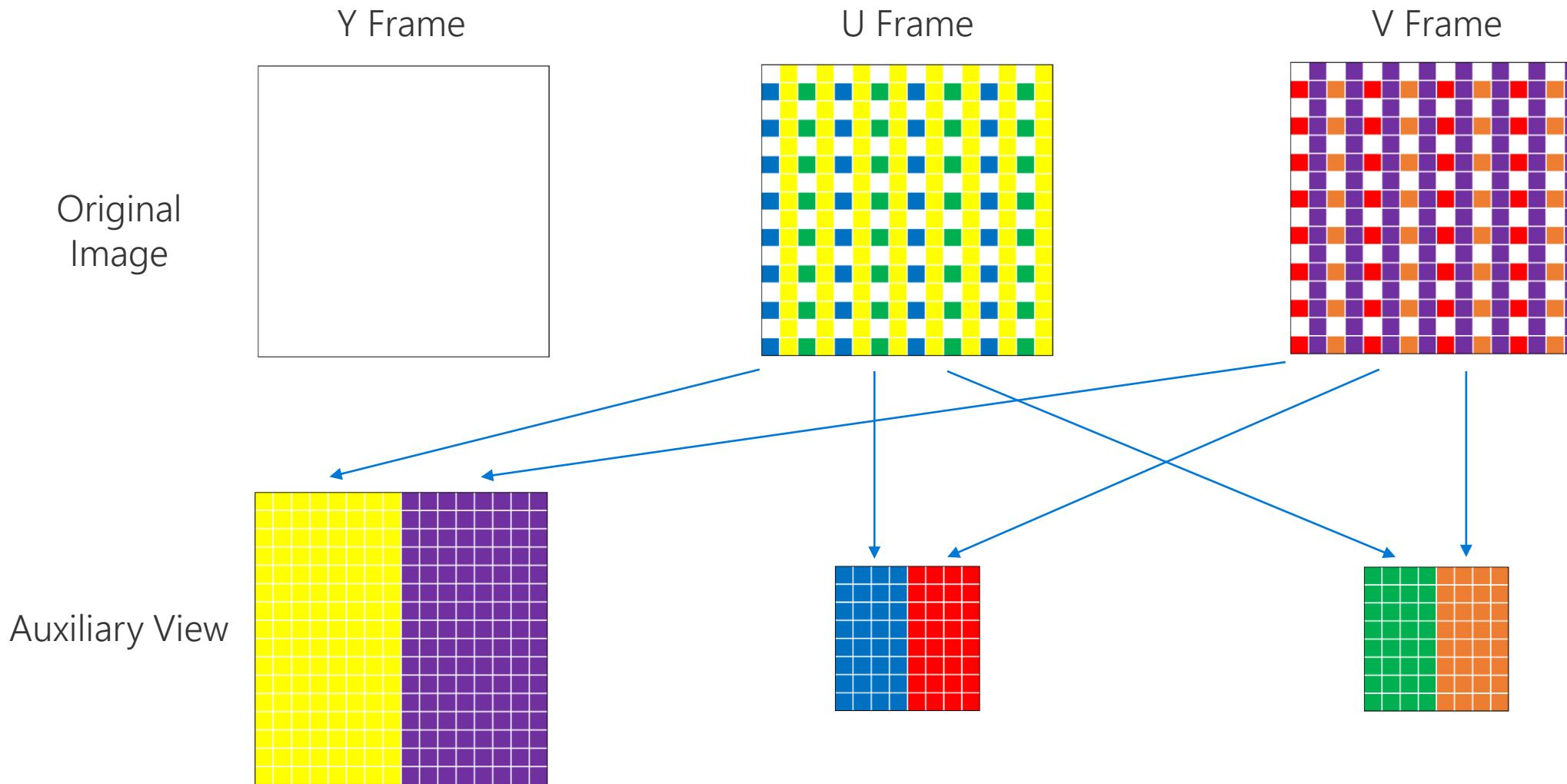
AVC444v2 Bitmap Stream Format

For RDP Client Test Suite,

- One new test case added(RDPEGFX_H264Codec_PositiveTest_AVC444v2)
  - Deserialize test data
  - Pack and send PDU
  - Check response
- Codec tool is used to generate test data in advance

RDSTLS Authentication

A variation of Enhanced RDP Security which is

- Primarily used in redirection scenarios
- Leveraging the TLS security protocol
  - Identity authentication
  - Encryption/decryption
- Client will check the identification of redirected server
  - With the certificate given by connection broker
- User authentication directly follows TLS handshake
  - Through the new RDSTLS PDUs

Roles of Server Redirection with RDSTLS Authentication

connect to

RD Connection Broker

Terminal Client

redirected to

RD Session Host
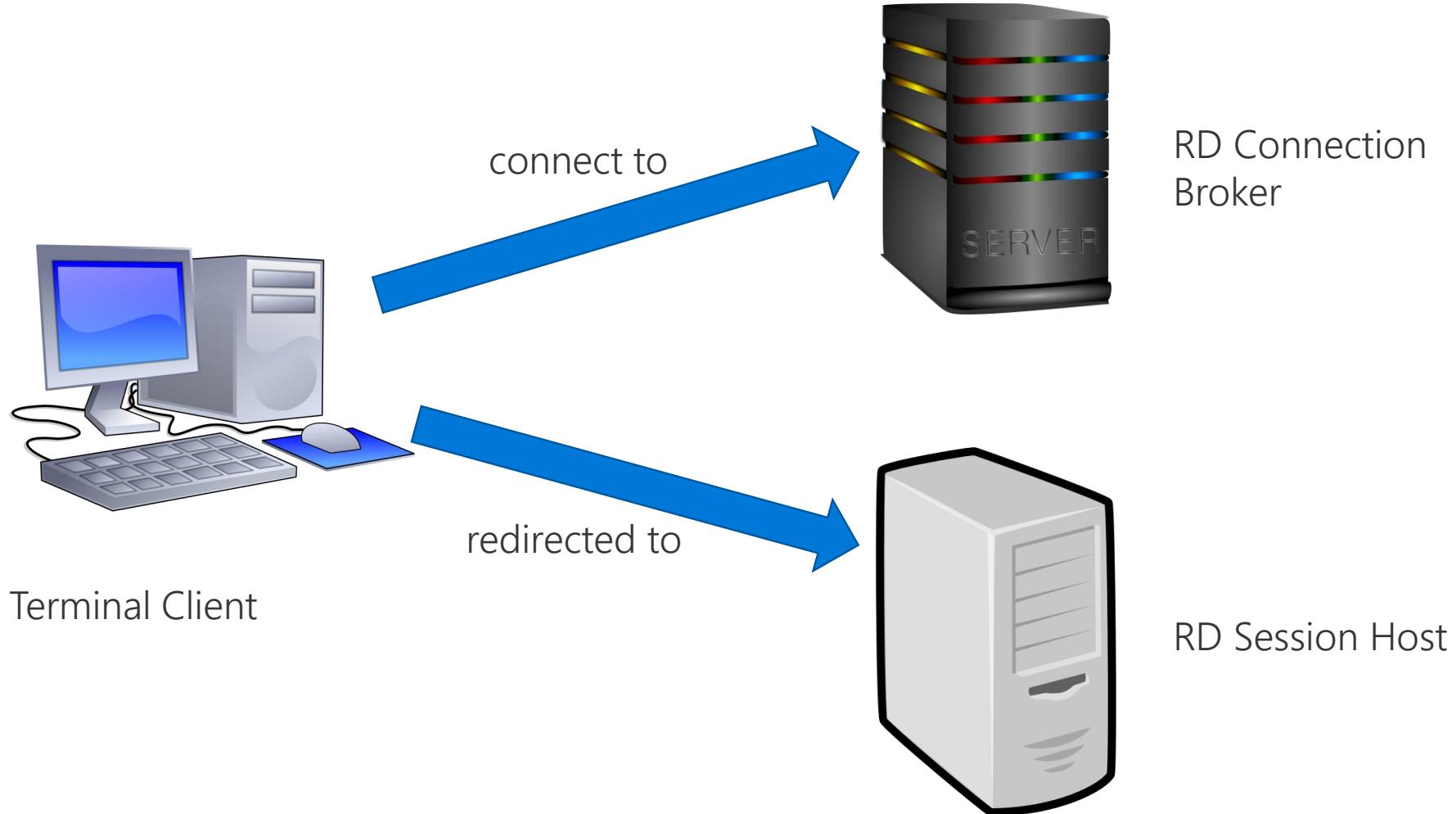
Sequence of Server Redirection with RDSTLS Authentication

1. Terminal client accomplishes the connection and authentication with RD connection broker.

Terminal Client

RD Connection
Broker

Sequence of Server Redirection with RDSTLS Authentication

2. RD connection broker sends RDP_SERVER_REDIRECTION_PACKET to terminal client with the address, certificate and credential information of RD session host.

RDP_SERVER_REDIRECTION_PACKET

Terminal Client

RD Connection Broker
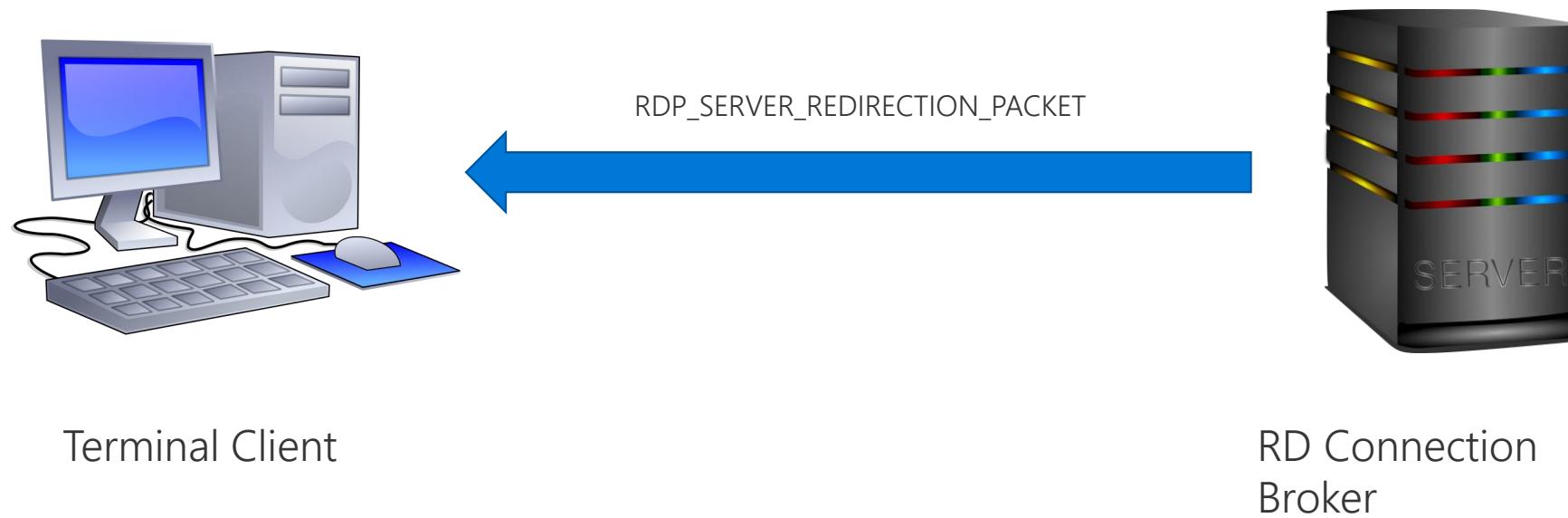
Sequence of Server Redirection with RDSTLS Authentication

3. Terminal client disconnect with RD connection broker.

Terminal Client

RD Connection
Broker

Sequence of Server Redirection with RDSTLS Authentication

4. Terminal client initiates connection with RD session host and negotiates with security protocol set to PROTOCOL_RDSTLS.

Client X.224 Connection Request PDU with requestedProtocols including PROTOCOL_RDSTLS

Server X.224 Connection Confirm PDU with selectedProtocol set to PROTOCOL_RDSTLS

Terminal Client
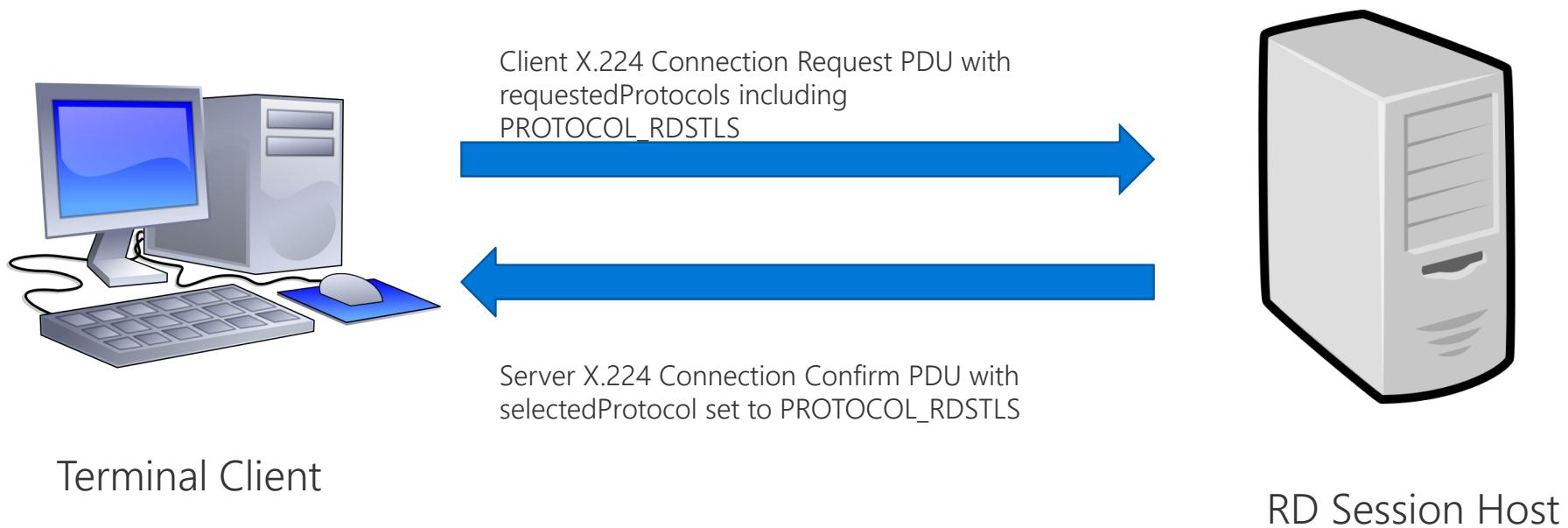
RD Session Host

Sequence of Server Redirection with RDSTLS Authentication

5. Terminal client initiates TLS handshake with RD session host and authenticates whether the certificate provided by RD session host is identical to that in RDP_SERVER_REDIRECTION_PACKET.

Terminal Client

RD Session Host

Sequence of Server Redirection with RDSTLS Authentication

6. RD session host authenticates terminal client by RDSTLS authentication.

RDSTLS Capabilities PDU

RDSTLS Authentication Request
PDU with Password Credentials

RDSTLS Authentication
Response PDU

Terminal Client
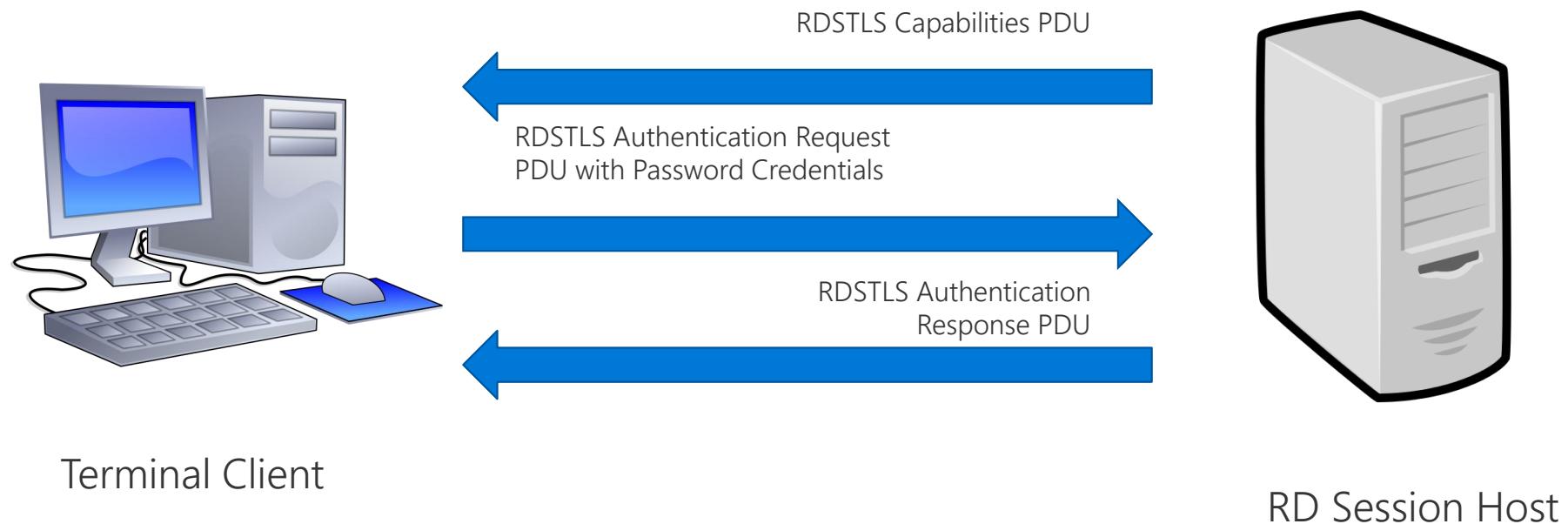
RD Session Host

Sequence of Server Redirection with RDSTLS Authentication

7. Terminal client continues the normal connection sequence with RD session host.

Terminal Client

RD Session Host

RDSTLS Authentication

For RDP Client Test Suite

- One new test case added(BVT_ServerRedirection_PositiveTest_RDSTLSAuthenticationWithPasswordCredentials)
  - server redirection through RDSTLS authentication
- Expects SUT
  - Handles new RDP_SERVER_REDIRECTION_PACKET
  - Reconnects using RDSTLS
  - Setting fields correctly
- Driver computer acts as...
  - Connection broker
  - Session host.

Agenda

Open Source Status

RDP Test Suites Overview

RDP Test Suites Updates

Demo

Demo

Run Test Suite with PTM
Protocol Based Control Agent
Codec Tool

# Reference: MS-SSIM: Multi-Scale Structural Similarity

- Method for measuring the similarity between two images.
- Calculated value from -1 to 1
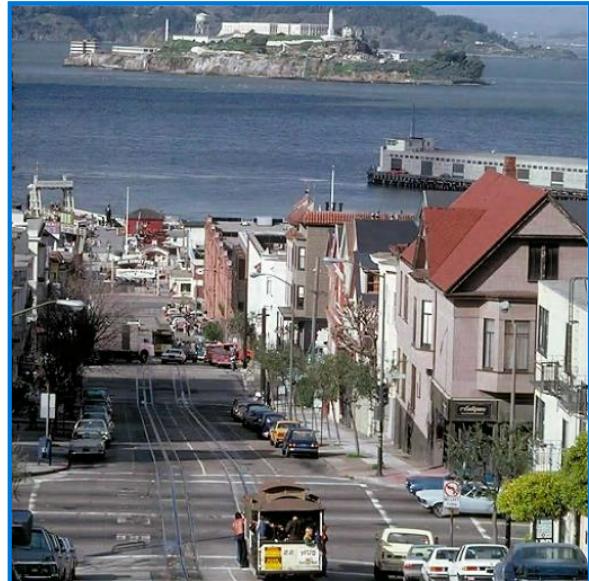
- Reference Image
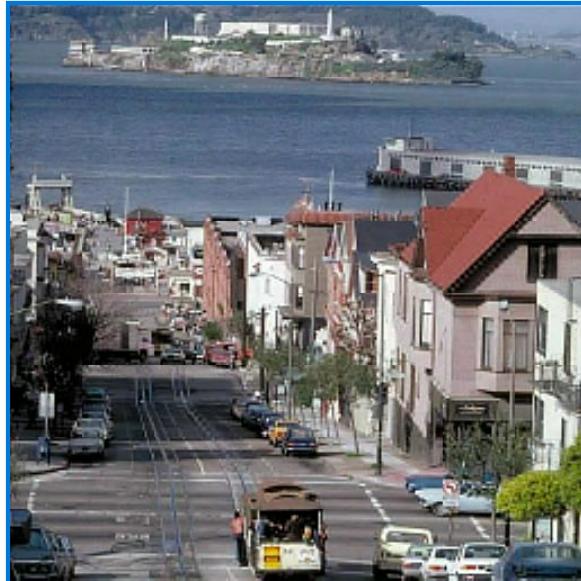
- Loss of high frequency data
- MS-SSIM = **0.974**
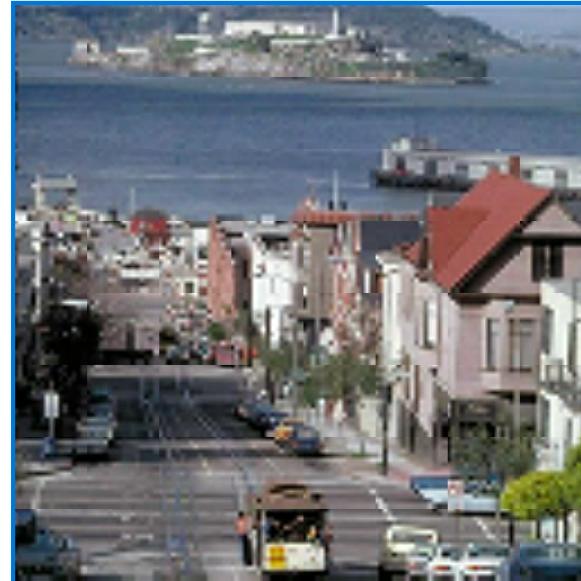
- Severely loss of high frequency data
- MS-SSIM = **0.882**
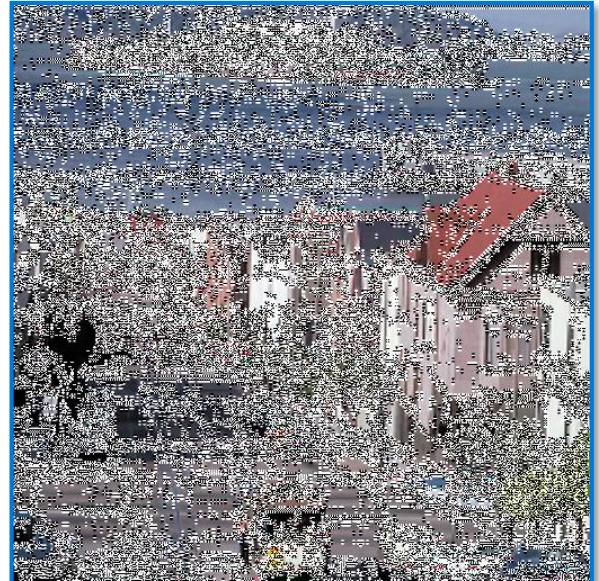
- Decoder error
- MS-SSIM = **0.175**

Reference: PTM

```
ptmcli.exe <-p|-profile profileName> [-s|-selected] [-r|-report reportfile]
[-outcome pass,fail,inconclusive,notrun] [-sortby name|outcome]
[-separator comma|space]
```

- ## PTM Capability

  - Guide: Embedded deployment guide illustration

  - Detect: SUT capability auto detection

  - Filter: Cases filtered by features and SUT capability

  - Configure: Test suite configuration recommendation

  - Run: GUI test case run w/o VS dependency

  - Investigate: Integrated test run logs

  - Rerun: Rerun and profile supported

  - Cmd: Command line supported

-p | -profile profileName   Specifies the file name of profile of the test
            suite to run.

-s | -selected   Turn this switch on to run only the test cases selected in the Run Page of PTM. Otherwise, run all the test cases in the profile.

-r | -report reportfile   Specifies the file name of the plain text report.

-outcome   Specifies the outcome of the test cases to be
            included in the plain text report.
Saparate using comma without space.
Possible values: pass,fail,inconclusive,notrun
Default value: pass,fail.

-sortby   Specifies the way to sort the test cases in the report.
Possible values: name outcome.
Default value: name

-separator   Specifies the separator in the plain text report.

Possible values: space comma
Default value: space

Reference: Resources

- Windows protocol Test Suite Open Source

  - https://github.com/Microsoft/WindowsProtocolTestSuites

- Protocol Test Framework Open Source

  - https://github.com/Microsoft/protocoltestframework

- PTM and Plugin Open Source

  - https://github.com/Microsoft/WindowsProtocolTestSuites/tree/4bdee2811d6e7e1d90b329d88ff2fc7646301cfd/ProtocolTestManager

- Windows Protocol documents for the RTM release on MSDN

 - https://msdn.microsoft.com/en-us/library/cc216517.aspx

Q&A

Helen Lu   Vivian Tian   Jiajun Wang   Huiren Jiang