# Networking Basics:

<https://www.softwaretestinghelp.com/networking-interview-questions-2/>

# Wifi Direct

* Wi-Fi Direct allows two devices to establish a direct Wi-Fi connection without requiring a wireless router and without internet connection.
* Mainly used for file transfer like video file.

# Test Strategy Doc Sections:

Following sections are expected in Test Strategy doc:

* List of features to be tested
* List of features not tested
* Environment/Setup details
* Escalation Procedure
* Pass/Fail criteria by Open Bug severity
* Test Entry/Exit criteria
* Suspension/Resumption criteria
* Test Deliverables/Reporting
* Bug Reporting
* Staffing/Training needs
* Assumptions/Dependencies
* Schedule and Resource Planning
* Risks/Mitigation plan

# IR Vs RF Remote

|  |  |  |
| --- | --- | --- |
|  | IR | RF |
| Coverage | @ 10 meters | @ 50 meters |
| Cost | Inexpensive | Expensive |
| Line of sight | Needs Line of Sight | No need for Line of sight |
|  |  |  |

**Apple - Video Streaming Software Quality Engineer**

* Knowledge of Video Streaming Technology
* Python
* QoS, packet traces
* HLS, HDS, or Smooth Streaming

# Wifi Security Modes

**WEP (Wired Equivalent Privacy)**

* Uses 24-bit Initialization Vectors (IVs), easy to break
* Uses 64-bit, 128-bit, 256-bit keys
* Vulnerable to attacks

**WPA (Wi-fi Protected Access)**

* Uses 128-bit TKIP (Temporal Key Integrity Protocol)
* Also called WPA-PSK (Wi-fi Protected Access Pre-Shared Key)
* With longer passphrase, its hard to break in WPA protected networks

**WPA2 (Wi-fi Protected Access)**

* Based on Advanced Encryption Standard (AES)
* 128 or 256-bit keys
* Hard to break the sufficiently large passwords

# Wifi Standards

* Standard – Bitrate – RF Frequency
* 802.11 a – 54 Mbps – 5 GHz
* 802.11 b – 11 Mbps – 2.4 GHz
* 802.11 g – 54 Mbps – 2.4 GHz
* 802.11 n – 108 Mbps – 2.4 GHz or 5GHz
* 802.11 ac – 1 Gbps – 5 GHz

# Wifi Channels/Frequencies

* 5 GHz – Faster data rates, shorter range
* 2.4 GHz – Slower data rates, longer range

# Wifi Test Matrix

Test Apps running on devices connecting to Wifi NW configured with:

* Security Modes - WEP, WPA, WPA2
* Wifi standards – 802.11 a, b, g, n, ac
* Wifi Channels – 2.4 GHz and 5 GHz

# HDMI

* HDMI is a high bandwidth standard for connecting digital devices
* HDMI also supports data protection through HDCP (high-bandwidth digital copy protection)
* HDCP is an authentication protocol
* Basically, each receiving device (HDTV) has identification data and encryption data stored on its extended display identification data (EDID) chip
* The source device, such as a [Blu-ray player](https://electronics.howstuffworks.com/blu-ray.htm), checks the authentication key of the receiving device, such as an HDTV. If both keys check out, the sending device moves on to the next step
* All HDMI-compatible devices are required to support HDCP
* HDMI is a type A connector has 19 pins

**HDMI-CEC(Consumer Electronic Channel)**

* CEC is just 1 pin on HDMI connector
* CEC allows supported devices to send instructions to each other
* For example, an [HD-DVD](https://electronics.howstuffworks.com/hd-dvd.htm) player could automatically turn on a home-theater receiver and an [HDTV](https://electronics.howstuffworks.com/hdtv.htm) when it started playing a disk
* HDMI Versions – 1.1, 1.2, 1.3, 1.4, 2.0
* HDMI 1.4 supports 4K/30 fps
* HDMI 2.0 supports 4K/60 fps
* Increase in bandwidth from previous versions
* HDMI 1.4 supports 100 Mbps

**HDMI Functions**

* A/V Data Transmission
* HDMI – CEC
* HDCP
* HPD – Hot Plug Detection

**Some Good Links:**

<https://www.lifewire.com/hdmi-facts-high-definition-multimedia-interface-1847337>

# RGB Vs YCbCr

RGB means that the picture is sent to your monitor with a Red, Green and Blue value for each pixel. Since the LCD panels all work with RGB pixels, it's the perfect way of sending a signal to an LCD screen.

YCbCr is a different video signal type where each pixel gets a *luminance* value (from complete dark to full bright), and two *chroma* values. Those two *chroma* values are representing the colors.

Afaik, YCbCr with 4:4:4 *should* display the same thing as Full RGB, but it's unnecessary to have a double conversion (and potential loss due to rounding) when it's eventually going to be displayed on a RGB matrix.

YCbCr 4:4:4 means that the image has a full channel (1920x1080 in your case) of luminance and 2 full channels of chroma.  
YCbCr 4:2:2 means that the image has a full channel of luminance, but the 2 chroma channels are **halved** (**960**x1080).

# Network Parameter Basics:

[**Packet loss**](https://en.wikipedia.org/wiki/Packet_loss)

The network may fail to deliver (drop) some packets due to [network congestion](https://en.wikipedia.org/wiki/Network_congestion). The receiving application may ask for this information to be retransmitted, possibly resulting in [congestive collapse](https://en.wikipedia.org/wiki/Congestive_collapse) or unacceptable delays in the overall transmission.

[**Latency**](https://en.wikipedia.org/wiki/Network_delay)

It might take a long time for each packet to reach its destination because it gets held up in long queues, or it takes a less direct route to avoid congestion. In some cases, excessive latency can render an application such as VoIP or online gaming unusable.

**Jitter** - is the variation in delay of a streaming application

Caused due to Congestion in routers and switches start dropping packets because they are coming in/out faster that what can be processed

**QoS - Quality of service** is a **measurement of the overall performance of a service**, such as a [telephony](https://en.wikipedia.org/wiki/Telephony) or [computer network](https://en.wikipedia.org/wiki/Computer_network) or a [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) service, particularly the performance seen by the users of the network. To quantitatively measure quality of service, several related aspects of the network service are often considered, such as [packet loss](https://en.wikipedia.org/wiki/Packet_loss), [bit rate](https://en.wikipedia.org/wiki/Bit_rate), [throughput](https://en.wikipedia.org/wiki/Throughput), [transmission delay](https://en.wikipedia.org/wiki/Transmission_delay), [availability](https://en.wikipedia.org/wiki/Availability), [jitter](https://en.wikipedia.org/wiki/Jitter).

For a Video Streaming Service, QoS cannot accurately characterize the users’ perception. Therefore, the concept of Quality of Experience (QoE) is what makes more sense for a Video Streaming Service.

**QoE of a Video Streaming service**

**Video Streaming Related Parameters**

* Startup Time
* Startup bitrate
* Frequency of Buffering
* % of time spent in Buffering
* Overall Playback time
* Number of Bitrate switches
* Frequency of Bitrate switches
* Overall Bitrate for complete streaming session
* Perceived Video/Audio Quality of Streaming session

**Other Parameters**

* Look and feel of App
* Ease of use
* Organization of Content/catalog
* How easy it is to find an asset in catalog
* Quality of Recommendations
* Subscription Model
* Collection of assets
* Navigation experience
* Player features – For ex: Frame Images during FF/REW
* Device on which App is supported

**HTTP Commands**

* **GET**: A client can use the GET request to get a web resource from the server.
* **HEAD**: HEAD request is similar to GET request. However, the server returns only the response header without the response body
* **POST**: Used to post data up to the web server. Example: In many Internet applications, the users are required to submit information to the server (e.g., the name, address, the search keywords). Once they fill in the requested data and hit the submit button, the browser packs the form data and submits them to the server, using either a GET request or a POST request. On receiving these details, the server can take some action.
* **PUT**: Ask the server to store the data. The PUT method requests that the enclosed entity be stored under the supplied [URI](https://en.wikipedia.org/wiki/URI). If the URI refers to an already existing resource, it is modified; otherwise create operation should happen if Request-URI is a valid resource URI.
* **DELETE**: Ask the server to delete the data.

**Performance Testing of Video Streaming Service**

**Some important parameters to consider for Load Testing**

* Different devices
* Different Resolutions – Consider most popular resolutions
* Different Bandwidth conditions
* Test from several regions/states
* Playback from beginning, from Pause Point

**Useful KPIs to Monitor When Load Testing**

* Percentage of HTTP request errors
* Average loading time of playlist for HLS, DASH (Manifest file)
* Average loading time of chunks for HLS, DASH (Actual AV segments/Payloads)
* Average Bitrate
* Average loading time of starting stream
* Lag time

**Network Sniffers**

* Wireshark
* Fiddler
* Any Other tools

**HLS file format for Live Profile:**

<https://developer.apple.com/documentation/http_live_streaming/example_playlists_for_http_live_streaming/live_playlist_sliding_window_construction>

**Http Response codes:**

<https://www.restapitutorial.com/httpstatuscodes.html>

**How does DRM Work:**

<https://www.brightcove.com/en/blog/2018/07/dealing-drm-understanding-drm-and-how-produce-protected-content>

<https://bitmovin.com/what-is-drm/>

**Python Resources & Coding Exercises:**

<https://www.pythonmorsels.com/resources/?__s=wzehb8rarpzj5tc6jsgd>

Python Loops

<https://treyhunner.com/2016/04/how-to-loop-with-indexes-in-python/>

List Comprehension:

<https://treyhunner.com/2015/12/python-list-comprehensions-now-in-color/>

**To Prepare for Apple Interview:**

* Python Coding Conventions
* Wireshark usage
* Python Programs
* HLS Manifest file reading

**Programs to refresh**

**Python Linked List**

[Remove Nth Node From End of List](https://leetcode.com/problems/remove-nth-node-from-end-of-list)

[Merge Two Sorted Lists](https://leetcode.com/problems/merge-two-sorted-lists)

[Reverse Linked List](https://leetcode.com/problems/reverse-linked-list)

[Split Linked List in Parts](https://leetcode.com/problems/split-linked-list-in-parts)

[Linked List Cycle](https://leetcode.com/problems/linked-list-cycle)

[Add Two Numbers](https://leetcode.com/problems/add-two-numbers)

[Middle of the Linked List](https://leetcode.com/problems/middle-of-the-linked-list)

[Rotate List](https://leetcode.com/problems/rotate-list)

[Remove Duplicates from Sorted List](https://leetcode.com/problems/remove-duplicates-from-sorted-list)

[Sort List](https://leetcode.com/problems/sort-list)

[Palindrome Linked List](https://leetcode.com/problems/palindrome-linked-list)

[Intersection of Two Linked Lists](https://leetcode.com/problems/intersection-of-two-linked-lists)

Create Binary Search Tree from list

How to test using PyTest?

**Python List**

**Reverse Python List**

**Solution-1**

**def** reverse\_list(input\_list):  
 first = 0  
 last = len(input\_list)-1  
  
 **while** first < last:  
 temp = input\_list[first]  
 input\_list[first] = input\_list[last]  
 input\_list[last] = temp  
 first += 1  
 last -= 1  
 **return** input\_list  
  
*#Driver Code*test\_list = [9,8,7,6,5,4,3,2,1]  
print(reverse\_list(test\_list))

**Solution-2**

**def** reverse\_list(input\_list):

**return**(input\_list[::-1])

[**Fibonacci Number**](https://leetcode.com/problems/fibonacci-number)

**def** fib(N):  
 *"""* **:type** *N: int* **:rtype***: int  
 """* **if** N == 0:  
 **return** 0  
 **elif** N == 1:  
 **return** 1  
 **else**:  
 **return** fib(N-1) + fib(N-2)

<https://www.geeksforgeeks.org/dictionary-counter-python-find-winner-election/>

votes = [**"john"**, **"johnny"**, **"jackie"**, **"johnny"**, **"john"**, **"jackie"**, **"jamie"**, **"jamie"**, **"john"**, **"johnny"**, **"jamie"**, **"johnny"**, **"john"**]

vote\_set = set(votes)  
winner = [(votes[0], votes.count(votes[0]))]  
  
**for** name **in** vote\_set:  
 **if** name **is** winner[0][0]:  
 **continue  
 else**:  
 **if** votes.count(name) > winner[0][1]:  
 winner.clear()  
 winner.append((name, votes.count(name)))  
 **elif** votes.count(name) == winner[0][1]:  
 winner.append((name, votes.count(name)))

[Flipping an Image](https://leetcode.com/problems/flipping-an-image)

[Transpose Matrix](https://leetcode.com/problems/transpose-matrix)

**def** transpose1(A):  
 row = len(A)  
 col = len(A[0])  
 result = [[**None**] \* row **for** \_ **in** range(col)]  
 **for** i **in** range(row):  
 **for** j **in** range(col):  
 result[j][i] = A[i][j]  
 **return** result  
  
**def** transpose2(A):  
 **return** [[A[i][j] **for** i **in** range(0,len(A))] **for** j **in** range(0,len(A[0]))]  
  
test\_matrix = [[1,2,3],[4,5,6]]  
print(transpose1(test\_matrix))  
print(transpose2(test\_matrix))

Merge Sort

**def** merge(list1, list2):  
 result\_list = []  
  
 **while** list1 **and** list2:  
 **if** list1[0] < list2[0]:  
 result\_list.append(list1[0])  
 list1.pop(0)  
 **else**:  
 result\_list.append(list2[0])  
 list2.pop(0)  
  
 **if** list1:  
 result\_list.extend(list1)  
 **if** list2:  
 result\_list.extend(list2)  
  
 **return** result\_list  
  
  
**def** merge\_sort(input\_list):  
 **if** len(input\_list) < 2:  
 **return** input\_list  
 **else**:  
 mid = len(input\_list) // 2  
 **return** (merge(merge\_sort(input\_list[:mid]), merge\_sort(input\_list[mid:])))  
  
*#Driver Code*input\_list = [9,8,7,6,5,4,3,2,1]  
print(merge\_sort(input\_list))

**Recursion:**

**Reverse a String**

def reverse\_string(in\_str):

if len(in\_str) == 0:

return in\_str

else:

return (reverse\_string(in\_str[1:]) + in\_str[0])

print(reverse\_string("Hello"))

**Reverse a List of characters:**

def reverse\_list(in\_str\_list):

if len(in\_str\_list) == 0:

return in\_str\_list

else:

return(reverse\_list(in\_str\_list[1:]) + list(in\_str\_list[0]))

print(reverse\_list(["h","e","l","l","o"]))

**Reverse a List of characters:**

def reverse\_list(in\_str\_list):

return in\_str\_list[::-1]

**Divide 2 integers without using Division operator:**

<https://stackoverflow.com/questions/5386377/division-without-using>

var divide = function (dividend, divisor) {

// Handle 0 divisor

if (divisor === 0) {

return NaN;

}

// Handle negative numbers

var isNegative = false;

if (dividend < 0) {

// Change sign

dividend = ~dividend+1;

isNegative = !isNegative;

}

if (divisor < 0) {

// Change sign

divisor = ~divisor+1;

isNegative = !isNegative;

}

/\*\*

\* Main algorithm

\*/

var result = 1;

var denominator = divisor;

// Double denominator value with bitwise shift until bigger than dividend

while (dividend > denominator) {

denominator <<= 1;

result <<= 1;

}

// Subtract divisor value until denominator is smaller than dividend

while (denominator > dividend) {

denominator -= divisor;

result -= 1;

}

// If one of dividend or divisor was negative, change sign of result

if (isNegative) {

result = ~result+1;

}

return result;

}

Binary Search

[Max Consecutive Ones](https://leetcode.com/problems/max-consecutive-ones)

[Majority Element](https://leetcode.com/problems/majority-element)

[Best Time to Buy and Sell Stock](https://leetcode.com/problems/best-time-to-buy-and-sell-stock)

[Missing Number](https://leetcode.com/problems/missing-number)

[Plus One](https://leetcode.com/problems/plus-one)

[Search Insert Position](https://leetcode.com/problems/search-insert-position)

[Largest Number At Least Twice of Others](https://leetcode.com/problems/largest-number-at-least-twice-of-others)

[Remove Duplicates from Sorted Array](https://leetcode.com/problems/remove-duplicates-from-sorted-array)

[Merge Sorted Array](https://leetcode.com/problems/merge-sorted-array)

[Rotate Array](https://leetcode.com/problems/rotate-array)

[Subsets](https://leetcode.com/problems/subsets)

[Rotate Image](https://leetcode.com/problems/rotate-image)

**Python Strings**

**Apple Interview Program**

**File Operations**

**Python CheatSheet**

**Data Structures**

**Tries -** [**https://medium.com/basecs/trying-to-understand-tries-3ec6bede0014**](https://medium.com/basecs/trying-to-understand-tries-3ec6bede0014)

**REST API Testing**

**HTTP CRUD**

POST Create a new resource.

GET Read a resource.

PUT Update an existing resource.

DELETE Delete a resource.

**Testing Terminologies**

* **Test strategy** is a high level document which defines the approach for software testing. It is basically derived from the Business Requirement document. Test strategy is developed by project manager or business analyst. It is kind of static document which sets the standards for testing so not updated often.
  + **Scope and objective**
  + **Business Issues**: How much is the budget of the project, how much time is required for testing, how much resources are needed etc
  + **Testing approach**: What type of testing is needed (performance, load, stress, functional etc.) and whether the testing is only manual or automation or both
  + **Test deliverables**: What are the documents required from the testing team, how they would keep the record of the testing cycles etc. will be included here.
  + **Defect tracking approach**: Which tool will be used for tracking the defects and how will the testing team communicate with the development team and how the flow would go for defects are decided at this point in test strategy.
  + **Training**: If there is some complex or new tool is introduced in the business then it is helpful if the team members are given proper training. What type of training and the responsible person to conduct such training is defined here.
  + **Automation**: If the project or business needs automation testing then the script language, tool used, reporting and code maintained is planned in test strategy.
  + **Risks**:
* **Test plan** is derived from SRS (*Software Requirement Specification*) which is prepared by test lead or manager. The main goal of test plan is to include all the details related to testing such as what to test, when to test, how to test and who will be the tester. Test plan is often not updated but if there is some new feature or change is introduced then it has to be updated accordingly.
  + **Test plan ID**: This is a unique ID which defines the test plan. It can be a number or name or mix of both, as per the convenience.
  + **Test environment**: This section defines what kind of environment is needed for the testing to carry out. For e.g. in device testing, usually a virtual set up is made to test emergency calling.
  + **Features to be tested/Not tested**: This will have all the details about the features which tester needs to test and what are the feature which are not tested (may be because it is not yet implemented or not tested for that particular release).
  + **Entry/Exit criteria**: These are the terms which define when to start or stop the testing. Standards will be defined under test strategy and followed by testers in test plan.
  + **Suspension Criteria and Resumption Requirements:**
  + **Test Deliverables** - Test Plan (this document itself), Test Cases, Test Scripts, Defect/Enhancement Logs, Test Reports
  + **Estimate and Schedule**
  + **Staffing and Training needs**
  + **Risks/Mitigation plans**
  + **Assumptions and Dependencies**
  + **Types** **of** **testing**: The types of testing required such as regression, functional, non-functional, stress etc. are defined and then executed by the respective tester.

**Verification Vs Validation**

**Verification** is the process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.

**Validation** is the process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements

**QA Vs QC**

SOFTWARE QUALITY ASSURANCE**(SQA)** is a set of activities for ensuring quality in software engineering processes (that ultimately result in the quality of software products). It includes:

* + Software Development Methodology
  + Project Management
  + Configuration Management
  + Requirements Development/Management
  + Estimation
  + Software Design
  + Testing
  + etc

**SQC** is a set of activities for ensuring quality in software products. The activities focus on identifying defects in the actual products produced. It includes:

* + *Reviews*
    - Requirement Review
    - Design Review
    - Code Review
    - Deployment Plan Review
    - Test Plan Review
    - Test Cases Review
  + *Testing*
    - [Unit Testing](http://softwaretestingfundamentals.com/unit-testing/)
    - [Integration Testing](http://softwaretestingfundamentals.com/integration-testing/)
    - [System Testing](http://softwaretestingfundamentals.com/system-testing/)
    - [Acceptance Testing](http://softwaretestingfundamentals.com/acceptance-testing/)

# Testing Smart TV

* Power On/OFF from Remote and Physical Switch
* Channel Scan after connecting RF cable
* Channel Zapping (CH +/-), Vol+/-,
* Media Button Opens Smart TV Menu option
* Network
  + Acquiring DHCP IP
  + Setting Static IP
  + Wifi
* Firmware Upgrade
* Remote Control
* Video Input
  + Check TV gets input from HDMI/Component/Composite/etc
* Resolutions
  + Set TV Display to specific Resolution and Play different Resolution Video and see if TV handles/Scales properly
* Audio/Video Streaming
* Media Playback from USB media device (Photos/Music/Video)
* Audio/HDMI Out to Home Theatre system
* HDMI/Component/Composite Video Out testing

To Apply:

<https://www.telenav.com/>

<https://job-openings.monster.com/sr-software-qa-engineer-santa-clara-ca-us-telenav-inc/205641639?jvs=cf,can-6342,can,0&WT.mc_n=olm_sk_feed_ziprecruiter>

Quantenna Communications: <https://www.linkedin.com/jobs/view/1130135096>

Senior Software QA Engineers  
<https://www.glassdoor.com/partner/jobListing.htm?pos=126&ao=352789&s=230&guid=00000169285aa90e8be80e4033112cb7&src=GD_JOB_AD&t=SR&extid=1&exst=EO&ist=L&ast=EOL&slr=true&cs=1_71be1807&cb=1551160224444&jobListingId=3120217579&ugo=d1d1b5c9-3a3c-4562-9ee5-d6cb6d0fdcb7&uido=4B1CAECACE48DE8901BAA6090ACD2CCE>

QA Associate / Engineer - STB / IPTV / QAM  
<https://www.glassdoor.com/partner/jobListing.htm?pos=120&ao=4321&s=230&guid=00000169285aa90e8be80e4033112cb7&src=GD_JOB_AD&t=SR&extid=1&exst=EO&ist=L&ast=EOL&slr=true&aa=1&ea=1&cs=1_21663a5f&cb=1551160224419&jobListingId=3066313680&ugo=d1d1b5c9-3a3c-4562-9ee5-d6cb6d0fdcb7&uido=4B1CAECACE48DE8901BAA6090ACD2CCE>

QA Manager  
<https://www.glassdoor.com/partner/jobListing.htm?pos=110&ao=8095&s=230&guid=00000169285aa90e8be80e4033112cb7&src=GD_JOB_AD&t=SR&extid=1&exst=EO&ist=L&ast=EOL&slr=true&aa=1&ea=1&cs=1_2db24aa5&cb=1551160224355&jobListingId=3117319773&ugo=d1d1b5c9-3a3c-4562-9ee5-d6cb6d0fdcb7&uido=4B1CAECACE48DE8901BAA6090ACD2CCE>

Senior Software QA Engineers (Multiple Openings)  
<https://www.glassdoor.com/partner/jobListing.htm?pos=102&ao=4128&s=230&guid=00000169285aa90e8be80e4033112cb7&src=GD_JOB_AD&t=SR&extid=1&exst=EO&ist=L&ast=EOL&slr=true&cs=1_2cfad46e&cb=1551160224313&jobListingId=3110874931&ugo=d1d1b5c9-3a3c-4562-9ee5-d6cb6d0fdcb7&uido=4B1CAECACE48DE8901BAA6090ACD2CCE>