1. **Giới thiệu :**

Bộ đệm khung là một phần của bộ nhớ truy cập ngẫu nhiên.Nó là một bộ đệm nhớ chứa các dữ liệu đại diện cho tất cả các pixel.

RFB là bộ đệm khung từ xa, là giao thức đơn giản để truy cập từ xa vào giao diện đồ họa của máy chủ . Áp dụng được tất cả loại cửa số X11, Window,macintosh.

Điểm cuối ở từ xa, nơi người dùng ngồi gọi là máy khách

Điểm cuối bắt nguồn thay đổi từ bộ đệm khung gọi là máy chủ RFB

Máy khách ngắt kết nối và sau đó , kết nối lại trạng thái giao diện vẫn được giữ nguyên

Nhiều máy khách có thể kết nối cùng 1 lúc, và giao diện các máy khách luôn luôn di động theo trạng thái giao diện của máy chủ RFB

1. **Giao diện :**

Mặt hiển thị đồ họa nguyên thủy (đặt hình chữ nhật chứa dữ liệu pixel tại vị trí x,y cho trước )

Một chuỗi các hinh chữ nhật này tạo cập nhật bộ đệm khung.Bản cật nhật thể hiện sự thay đổi từ bộ đệm khung này, sang bộ đệm khung khác Các hình chữ nhật trong bản cập nhật thường rời rac.

Một bản cập nhật từ máy chủ sang máy khách để phản hồi yêu cầu rõ ràng

Máy khách và mạng chậm thì tỉ lệ cập nhật thấp

1. **Mô hình màn hình :**

Ở dạng đơn giản giao thức sử dụng bộ đệm hình chữ nhật.Các bản cập nhật được chứa trong vùng đệm này, không có ở bên ngoài.Ứng dụng khách chỉ cần trình bày vùng đệm này cho người dùng.Máy chủ và khách có khả năng mở rộng mô hình bằng cách thêm nhiều màn hình. Nhiều màn hình có thể chồng lên nhau một phần hoặc toàn phần

Máy khách phải theo dõi nội dung toàn bộ bộ đệm khung

Khách hàng có thể yêu cầu thay đổi kích thước bộ đệm khung và bố cục màn hình

Máy chủ có thể tự do phê duyệt hoặc từ chối

Chỉ thay đổi bố cục màn hình không ảnh hưởng đến nội dung bộ đệm khung.Do đó máy khách phải theo dõi các kích thước nội dung hiện tại và so sánh nó với kích thước nhận hình chữ nhật, nếu có sự thay đổi thì mới có thể loại bỏ nội dung bộ đệm khung

1. **Giao thức đầu vào :**

Dựa trên mô hình máy trạm tiêu chuẩn của bàn phím và thiết bị trỏ nhiều nút

Các sự kiện đầu vào khi được người dùng nhấn một phím hoặc nút chuột hoặc chuột di chuyển từ máy khách gửi đến máy chủ

Nếu không phù hợp tiêu chuẩn thì chúng ta có thể mở rộng ở phần **General Input Interface**

1. **Trình bày dữ liệu pixel:**

Sự tương tác giữa Client-Server liên quan đến thương lượng về định dạng và mã hóa mã dữ liệu Pixel được gửi.

Máy chủ luôn cung cấp dữ liệu Pixel ở dạng máy khách muốn, nếu máy khách có thể đáp ứng nhiều loại mã hóa khác nhau, nó có thể chọn loại mã hóa phù hợp nhất

Mã hóa đề cập đến cách dữ liệu pixel của 1 hình chữ nhật sẽ được gửi.

1. **Phân giao thức mở rộng :**

Một số cách mà giao thức có thể được mở rộng :

Các bảng mã mới :

Một kiểu mã hóa mới có thể được thêm vào giao thức tương tối dễ dàng trong khi vẫn duy trì khả năng tương thích với các chủ và máy khách hiện tại

Mã hóa giả :

Ngoài các mã hóa đặc trưng, máy khách có thể yêu cầu mã hóa giả để khai báo với máy chủ rằng nó hỗ trợ một phần mở rộng nhất cho giao thức, giả sử nếu máy chủ không có tiện ích mở rộng mã hóa giả, sẽ bỏ quan phần mã hóa giả

Các loại bảo mật mới :

Việc thêm một loại bảo mật mới mang lại sự linh hoạt tối ưu trong việc của đổi hành vin của giao thức mà k làm khả năng tương thích với các máy khách và máy chủ hiện có .

1. **Chuỗi mã hóa :**

Mã hóa được sử dụng cho các chuỗi trong giao thức trướic đây thường là không xác định, vậy nên có rất nhiều cách triển khai các loại mã hóa khác nhau

RFP khuyến khích sử dụng bộ mã hóa UTF-8

1. **Thông điệp từ giao thức :**

Nó được sử dụng qua kết nối TCP/IP

Có 3 giao đoạn

Giai đoạn bắt tay mục đích của nó là đồng ý về phiên bản giao thức và loại bảo mật

Giai đoạn thứ hai là giai đoạn khởi tạo nơi máy khách và máy chủ thông điệp ClientInit và ServerInit

Giai đoạn cuối là tương tác giao thức bình thường

* 1. **Thông điệp bắt tay :**

1. **Protocol Version :**

Máy chủ gửi -> máy khách một thông điệp Protocol Version , điều này cho biết số phiên bản giao thức RFP của máy chủ hỗ trợ, và trả lời lại bằng thông báo tương tự

Máy khách không được yêu cầu phiên bản giao thức cao hơn máy chủ cung cấp

Protocol Version gồm 12Byte RFP xxx.yyy\n

1. **Bảo mật :**

Khi phiên bản được quyết định , máy chỉ và máy khách phải đồng về cách loại bảo mật

Phiên bản 3.7 Máy chủ liệt kê kiểu bảo mật hợp lệ, máy chủ sẽ gửi lại một byte duy nhất cho biết kiểu dữ hiệu hợp lệ, ngược lại nếu Number of securyti bằng 0 thì sẽ không hỗ trợ kết nối

1. **Kết quả bảo mật :**

Máy chủ sẽ gửi một thông báo cho máy khách biết bắt tay bảo mật có thành công hay không

Phiên bản 3.8 trở : Sẽ gửi lí do thất bại

Phiên bản 3.3 và 3.7 nếu không thành công , máy chủ sẽ đóng kết nối

* 1. **Các loại bảo mật :**

a.**Không có :**

Không cần xác thực và dữ liệu giao thức sẽ được gửi mà không được mã hóa

3.8 trở đi Giao thức tiếp tục với thông báo Sercurity Result

3.3 và 3.7 Giao thức chuyển sang giai đoạn khởi tạo

b. **Xác thực VNC :**

Xác thực VNC sẽ được sử dụng và dữ liệu giao thức sẽ được gửi đi không được mã hóa.

Máy chủ gửi 16byte Challenge, máy khách mã hóa Challenge bằng DES

Giao thức tiếp tục thông báo với Security Result

c.Loại bảo mật chặt chẽ :

Là một phần mở rộng giao thức chung cho phép thực hiện 3 điều :

Đường hầm dữ liệu

Xác thực

Khả năng của máy chủ : Cho phép máy chủ thông báo cho máy khách về các khả năng của máy chủ

Sau loại máy bảo mật chặt chẽ, máy chủ sẽ gửi các đường hầm được hỗ trợ

1. **Vencrypt :**

Kiểu bảo mật này là một phương pháp xác thực chung gồm nhiều kiểu con xác thực

Máy chủ gửi tới phiên bản cao nhất nó có thể hỗ trợ, máy khách chủ cũng gửi lại phiên bản nó có thể hỗ trợ cao nhất[

* 1. **Thông báo khởi tạo :**

Khi máy khách và máy chủ đã thỏa thuận với nhau.Thì giao thức sẽ chuyển qua giai đoạn khởi tạo,máy khách gửi thông điệp ClientInit và máy chủ sẽ gửi thông điệp ServerInit

1. ClientInit : Shared-flag có giá trị khác 0 nếu máy chủ cố gắng kết nối đến các máy khác, giá trị là zero nếu máy chủ cấp quyền truy cập độc quyền cho máy khách
2. ServerInit :

Sau khi nhận đc Client máy chủ sẽ gửi 1 thông báo ServerInit.

Thông báo này cho biết chiều rộng và chiều cao của bộ đệm khung của máy chủ

* 1. Thông báo từ máy khách đến máy chủ :

Loạt thông báo mà máy khách gửi đến máy chủ phải hỗ trợ :

SetPixelFormat,SetEncodings,FrameBufferUpdateRequest,KeyEvent,PointerEvent,ClientCutText,

1. SetpixelFormat : Giá trị Pixel được gửi trong thông báo FrameBufferUpdate,nếu máy khách k yêu cầu thì máy chủ sẽ gửi các giá trị pixel ở dạng tự nhiên được chỉ định trong thông báo của ServerInit
2. SetEncodings :

Đặt kiểu mã hóa mà máy chủ có thể gửi dữ liệu pixel. Thứ tự của các kiểu mã hóa này là một gợi í, máy chủ có chọn hoặc không. Dữ liệu pixel có thể luôn được gửi ở dạng mã hóa thô .

Ngoài các mã hóa phổ biến máy khách có thể yêu cầu mã hóa giả

1. Framebuffer Update Request :

FUR được xác định bởi X-vị trí y-vị trí, chiều rộng và chiều cao

Máy chủ thường phản hồi FUR bằng cách gửi FramebuffferUdate duy nhất và có thể gửi cho nhiều FUR

1. KeyEvent :

Có 2 kiểu sự kiện : Nhấn là FlagDown khác 0, thả là bằng 0

1. PointerEvent :

Cho biết chuyển động của con trỏ hoặc nhấn hoặc thả nút con trỏ .

Con trỏ hiện tại là (x- vị trí, y-vịtrí) trạng thái của nút 1 đến 8 được đại diện bởi bit 0 đến 7 của button-mask

Trên chuột thông thường nút 1, 2 và 3 tương ứng với các nút trái giữa phải, trên bánh xe 4 có nghĩa là lên , nút 5 có nghĩa là xuống, nút 6 có nghĩa là trái, 7 có nghĩa là phải .

1. ClientCuxtext :

Máy khách có văn bản ISO 8859-1 (Latin-1) mới trong bộ đệm cắt của nó.

1. Enable Continuous Update :