



Centro de Enseñanza Técnica Industrial

Plantel Colomos

Ingeniería en Desarrollo de Software

Nombre Alumno: José Rafael Ruiz Gudiño

Registro: 20110374

Desarrollo para Dispositivos Móviles II

Práctica 10 - Servicios de red y publicación

6°P

T/M

29/11/2022

Contenido

¿Qué es comunicación punto a punto (P2P)?	3
Tipos de comunicación punto a punto	3
Redes P2P no estructuradas	3
Redes P2P estructuradas	3
Redes híbridas P2P:	3
Ventajas de la comunicación P2P	3
Desventajas de la comunicación P2P	4
Implementación	4
Permisos	4
Establecer un receptor de emisión y administrador entre pares	5
Cómo conectarse con un par	5
Proceso de publicación de aplicaciones móviles en Android	6
Cómo preparar una app para el lanzamiento	6
Configurar la aplicación para el lanzamiento	6
Compilar y firmar una versión de actualización de la app	6
Probar la versión de lanzamiento de la app	7
Actualizar los recursos de la app para el lanzamiento	7
Preparar servidores y servicios remotos de los cuales dependa la app	7
Cómo lanzar una app en Google Play	7
Preparar materiales de promoción	7
Configurar opciones y subir recursos	7
Publicar la versión de lanzamiento de una app	8
Lista de tareas para el lanzamiento	8
Conclusión	10
Referencias	11

¿Qué es comunicación punto a punto (P2P)?

La computación o redes punto a punto (P2P) es una arquitectura de aplicación distribuida que divide tareas o cargas de trabajo entre pares. Los pares son participantes igualmente privilegiados y equipotentes en la red. Se dice que forman una red de nodos peer-to-peer.

Los pares hacen que una parte de sus recursos, como la potencia de procesamiento, el almacenamiento en disco o el ancho de banda de la red, estén directamente disponibles para otros participantes de la red, sin necesidad de una coordinación central por parte de servidores o hosts estables. Los pares son tanto proveedores como consumidores de recursos, en contraste con el modelo tradicional cliente-servidor en el que se divide el consumo y la oferta de recursos.

Tipos de comunicación punto a punto

Redes P2P no estructuradas: en este tipo de red P2P, cada dispositivo puede hacer una contribución igual. Esta red es fácil de construir ya que los dispositivos se pueden conectar aleatoriamente en la red. Pero al no estar estructurado, se vuelve difícil encontrar contenido. Por ejemplo, Napster, Gnutella, etc.

Redes P2P estructuradas: Está diseñado mediante un software que crea una capa virtual para colocar los nodos en una estructura específica. Estos no son fáciles de configurar, pero pueden dar fácil acceso a los usuarios al contenido. Por ejemplo, P-Grid, Kademia, etc.

Redes híbridas P2P: combina las características de las redes P2P y la arquitectura cliente-servidor. Un ejemplo de una red de este tipo es encontrar un nodo utilizando el servidor central.

Ventajas de la comunicación P2P

- **Fácil de mantener:** La red es fácil de mantener porque cada nodo es independiente de los demás.
- **Menos costoso:** dado que cada nodo actúa como un servidor, por lo tanto, se ahorra el costo del servidor central. Por lo tanto, no hay necesidad de comprar un servidor caro.
- **Sin administrador de red:** En una red P2P dado que cada nodo administra su propia computadora, por lo tanto, no hay necesidad de un administrador de red.
- **Agregar nodos es fácil:** agregar, eliminar y reparar nodos en esta red es fácil.
- **Menos tráfico de red:** En una red P2P, hay menos tráfico de red que en una red cliente/servidor.

Desventajas de la comunicación P2P

- Los datos son vulnerables: debido a que no hay un servidor central, los datos siempre son vulnerables a perderse debido a que no hay una copia de seguridad.
- Menos seguro: se vuelve difícil asegurar la red completa porque cada nodo es independiente.
- Rendimiento lento: en una red P2P, cada computadora es accedida por otras computadoras en la red, lo que ralentiza el rendimiento del usuario.
- Archivos difíciles de localizar: en una red P2P, los archivos no se almacenan de forma centralizada, sino que se almacenan en ordenadores individuales, lo que dificulta su localización.

Implementación

La función Wi-Fi directo (también conocida como Wi-Fi entre pares o P2P) le permite a tu aplicación encontrar rápidamente dispositivos cercanos e interactuar con ellos, en un rango que supera la capacidad de Bluetooth.

Las API de Wi-Fi entre pares (P2P) les permiten a las apps comunicarse con dispositivos cercanos sin la necesidad de conectarse a una red o un hotspot. Si tu app está diseñada para ser parte de una red segura de alcance cercano, Wi-Fi directo es una opción más adecuada que las redes Wi-Fi tradicionales ad hoc por los siguientes motivos:

Wi-Fi directo admite encriptación WPA2. (Algunas redes ad hoc solo admiten encriptación WEP).

Los dispositivos pueden transmitir los servicios que proporcionan, lo que ayuda a otros dispositivos a descubrir pares adecuados con mayor facilidad.

Cuando determina qué dispositivo debería ser el propietario del grupo en la red, Wi-Fi directo examina la administración de energía, la IU y las capacidades de servicio de cada dispositivo y usa esta información para elegir el dispositivo que pueda manejar las responsabilidades del servidor con mayor eficacia.

Android no admite el modo Wi-Fi ad hoc.

Permisos

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.nsdchat"
    ...
    <uses-permission
        android:required="true"
        android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission
        android:required="true"
        android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission
        android:required="true"
```

```

        android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission
    android:required="true"
    android:name="android.permission.INTERNET"/>
...

```

Establecer un receptor de emisión y administrador entre pares

```

private final IntentFilter intentFilter = new IntentFilter();
...
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    // Indica un cambio en el estado de Wi-Fi P2P.
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_STATE_CHANGED_ACTION);

    // Indica un cambio en la lista de pares disponibles.
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_PEERS_CHANGED_ACTION);

    // Indica que el estado de la conectividad Wi-Fi P2P ha cambiado
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_CONNECTION_CHANGED_ACTION);

    // Indica que los detalles de este dispositivo han cambiado.
    intentFilter.addAction(WifiP2pManager.WIFI_P2P_THIS_DEVICE_CHANGED_ACTION);
    ...
}

```

Al final del método `onCreate()`, obtén una instancia de `WifiP2pManager` y llama a su método `initialize()`. Este método muestra un objeto `WifiP2pManager.Channel`, que usarás más adelante para conectar la app al framework P2P de Wi-Fi.

```

Channel channel;
WifiP2pManager manager;

@Override
public void onCreate(Bundle savedInstanceState) {
    ...
    manager = (WifiP2pManager) getSystemService(Context.WIFI_P2P_SERVICE);
    channel = manager.initialize(this, getMainLooper(), null);
}

```

Cómo conectarse con un par

Para conectarte a un par, crea un objeto `WifiP2pConfig` nuevo y copia datos en este desde el `WifiP2pDevice` que representa el dispositivo al que deseas conectarte. Luego, llama al método `connect()`.

```

@Override
public void connect() {
    // Picking the first device found on the network.
    WifiP2pDevice device = peers.get(0);

    WifiP2pConfig config = new WifiP2pConfig();
    config.deviceAddress = device.deviceAddress;
}

```

```

config.wps.setup = WpsInfo.PBC;

manager.connect(channel, config, new ActionListener() {

    @Override
    public void onSuccess() {
        // WiFiDirectBroadcastReceiver notifies us. Ignore for now.
    }

    @Override
    public void onFailure(int reason) {
        Toast.makeText(WiFiDirectActivity.this, "Connect failed. Retry.",
            Toast.LENGTH_SHORT).show();
    }

});
}

```

Proceso de publicación de aplicaciones móviles en Android

La publicación es el proceso general por el cual las apps de Android se ponen a disposición de los usuarios. Cuando se publica una aplicación para Android, se realizan dos tareas principales:

- **Se prepara la app para su lanzamiento:** Durante el paso de preparación, se compila una versión de lanzamiento de la aplicación que los usuarios pueden descargar e instalar en sus dispositivos Android.
- **Se realiza el lanzamiento de la app para los usuarios:** Durante el paso de lanzamiento, se publicita, vende y distribuye la versión de actualización de la aplicación para los usuarios.

Cómo preparar una app para el lanzamiento

La preparación de una aplicación para su lanzamiento es un proceso de varios pasos que implica las siguientes tareas:

Configurar la aplicación para el lanzamiento

Como mínimo, se debe quitar las llamadas de Log y el atributo android:debuggable del archivo de manifiesto. También se debe configurar la información de versión de la app. Asimismo, es posible que se deba configurar varios parámetros más para cumplir con los requisitos de Google Play o adaptar cualquier método que se use para lanzar la aplicación.

Si se usan los archivos de compilación de Gradle, se puede emplear el tipo de compilación de lanzamiento a fin de configurar los ajustes de compilación para la versión publicada de la app.

Compilar y firmar una versión de actualización de la app

Se pueden usar los archivos de compilación de Gradle con el tipo de compilación release a fin de compilar y firmar una versión de actualización de la app.

Probar la versión de lanzamiento de la app

Antes de distribuir la app, debes probar por completo la versión de actualización en al menos un teléfono celular y una tablet de destino.

Actualizar los recursos de la app para el lanzamiento

Asegurarse de que todos los recursos de la app, como los archivos y gráficos multimedia, estén actualizados y se incluyan con la app o por etapas en los servidores de producción correspondientes.

Preparar servidores y servicios remotos de los cuales dependa la app

Si la app depende de servidores o servicios externos, se debe controlar que sean seguros y estén listos para la producción.

Es posible que se deban realizar varias tareas como parte del proceso de preparación. Por ejemplo, se deberá obtener una clave privada para firmar la aplicación, crear un ícono y también es conveniente que se prepare un contrato de licencia para el usuario final (CLUF) a fin de proteger a la persona, organización y propiedad intelectual.

Cuando se termine de preparar una aplicación para el lanzamiento, se contará con un archivo .apk firmado que se puede distribuir a los usuarios.

Cómo lanzar una app en Google Play

Google Play es una plataforma de publicación sólida que permite publicitar, vender y distribuir aplicaciones para Android a usuarios de todo el mundo. Cuando se lanzan aplicaciones a través de Google Play, se tiene acceso a un paquete de herramientas de desarrollo que permiten analizar las ventas, identificar las tendencias de mercado y controlar los destinatarios a los cuales se distribuyen tus aplicaciones. También tienes acceso a varias funciones que aumentan las ganancias, como la facturación integrada y las licencias para aplicaciones. La amplia matriz de herramientas y funciones, junto con varias características de la comunidad de usuarios finales, convierten a Google Play en el principal mercado de compra y venta de aplicaciones para Android.

Lanzar una aplicación en Google Play es un proceso simple que consta de tres pasos básicos:

Preparar materiales de promoción

Para aprovechar al máximo las capacidades de mercado y publicidad de Google Play, se deben crear materiales de promoción para la aplicación, como capturas de pantalla, videos, gráficos y texto promocional.

Configurar opciones y subir recursos

Google Play permite orientar una app hacia un grupo de usuarios y dispositivos de todo el mundo. Cuando se configuran varios ajustes de Google Play, se puede elegir los países a los que se desea llegar, los idiomas que se quiere usar y el precio que

se pretende imponer en cada región. También se puede configurar detalles de la ficha, como el tipo de aplicación, la categoría y la clasificación del contenido. Cuando se termina de configurar las opciones, se puede subir el material de promoción y la aplicación como borrador (sin publicar).

Publicar la versión de lanzamiento de una app

Si se sabe que los ajustes de la publicación están configurados correctamente y que la aplicación subida está lista para lanzarse al público, solo se hace clic en la opción Publicar de Play Console, y estará disponible para la descarga en todo el mundo en algunos minutos.

Lista de tareas para el lanzamiento

Comprende las Políticas del Programa para programadores. Las Políticas del programa para desarrolladores se diseñaron para garantizar que Play Store sea un recurso confiable para los usuarios de Android. Revisa atentamente las políticas, ya que el incumplimiento tendrá consecuencias.

Prepara una cuenta de desarrollador. Regístrate para obtener una cuenta de desarrollador y verifica que los detalles de la cuenta sean correctos. Si tienes pensado vender productos, configura tu cuenta de comerciante.

Planifica la localización. Si deseas incluir copias localizadas de tu app durante el lanzamiento, comienza a planearlas con anticipación y sigue la Lista de tareas para la localización.

Planifica las publicaciones simultáneas. La publicación de tu app en varias plataformas y dispositivos maximiza tus actividades de promoción y cantidad de instalaciones, por lo que debes planearla con anticipación. Si no puedes lanzar tu app en todas las plataformas de manera simultánea, pídeles a los usuarios información de contacto para informarles cuando esté lista la app.

Prueba los lineamientos de calidad. Los lineamientos de calidad de todas las apps y los criterios específicos para tablets, TVs y apps de Auto brindan plantillas de prueba. Puedes usarlas para verificar que tu app cuente con el diseño de IU y las funciones que esperan los usuarios de Android.

Segmenta el lanzamiento a un nivel de API reciente. Google Play exige que las apps nuevas se orienten como mínimo a Android 9.0 (nivel de API 28), y el mismo requisito es obligatorio para las actualizaciones a partir del 1 noviembre de 2019.

Compila tu Android App Bundle. Cuando estés listo para publicar la app, ya sea en versión de prueba o como producto final, crea tu Android App Bundle en Android Studio. Google Play usará el paquete de aplicación para generar, firmar y entregar APK optimizados para el dispositivo de cada usuario, lo que creará una app más pequeña para descargar e instalar.

Ejecuta pruebas internas. Usa el segmento de pruebas internas para lanzar tu app para un máximo de 100 verificadores internos y recibir comentarios antes de publicarla para usuarios externos en los segmentos de pruebas cerradas, abiertas o de producción. Usa varios segmentos de prueba cerrada para diferentes versiones de la app antes de lanzarla a los de prueba abierta o producción.

Planifica la ficha de Play Store de la app. Prepara las descripciones, los gráficos promocionales, las capturas de pantalla y los videos que agregarás a la página de Play Store de tu app. Si es necesario, asegúrate de incluir un vínculo a tu política de privacidad. Localiza la ficha de Play Store a todos los idiomas disponibles en tu app y para todos los países a los que quieras segmentarla.

Genera interés en tu app o juego con el registro previo y configura una página personalizada de la ficha de registro previo.

Sube el Android App Bundle al segmento de pruebas cerradas o abiertas. Los segmentos de pruebas cerradas y abiertas pueden ayudar de manera considerable a descubrir problemas con tu app y te dan la oportunidad de resolverlos para mejorar la calidad de la publicación inicial. Obtén información sobre cómo subir un Android App Bundle y consulta las prácticas recomendadas para las pruebas abiertas.

Define los dispositivos compatibles con tu app. Permite que Play Store sepa en qué versiones de Android y tamaños de pantalla funcionará tu app.

Consulta los reportes previos al lanzamiento. Cuando subas un Android App Bundle al segmento de pruebas cerradas o abiertas, recibirás un informe previo al lanzamiento. De esta manera, podrás identificar los problemas que se detecten cuando se pruebe la app de forma automática en una gran variedad de dispositivos reales con diferentes versiones de Android.

Configura el precio y los países de distribución de la app. Una vez que hayas determinado tu modelo de monetización, configura la app como gratuita o pagada, y selecciona los países donde se distribuirá.

Habilita las opciones de distribución adecuadas. En la página Precios y distribución, habilita los dispositivos y programas específicos, como Wear OS by Google, Android TV y Designed for Families. De esta manera, Google Play puede revisar tu app y, cuando la apruebe, hacerla más detectable para los usuarios.

Configura tus productos integrados en la aplicación y suscripciones. Confirma los países en los que puedes vender tu app, así como la moneda y los impuestos que debes considerar. Luego, configura tu cuenta del comerciante y agrega información y precios para los productos integrados en la aplicación y las suscripciones.

Determina la clasificación del contenido de la app. Brindar una calificación apropiada para tu app es requisito de las Políticas del Programa para Desarrolladores, pero también garantiza que los públicos etarios apropiados vean tu app.

Pasa a la revisión final y publicación. En primer lugar, retrocede y verifica nuevamente si completaste todas las tareas de esta lista. Luego de este proceso, podrás publicar la app en el canal de producción. Si publicarás una actualización, usa un lanzamiento en etapas para hacerlo de manera progresiva a una creciente cantidad de usuarios. De esta manera, podrás detener la actualización si se detecta un problema, y la cantidad de usuarios afectados será menor.

Promociona la app. Comienza a promocionar tu app con la insignia de Google Play. Si quieres aumentar tu público, prueba el resto de nuestras herramientas de marketing y consulta las prácticas recomendadas para el crecimiento.

Explora y responde las opiniones de los usuarios. Descubre qué opinan los usuarios sobre tu app y respóndeles para solucionar los problemas que no se detectaron durante las pruebas.

Consulta Android vitals. Esta función te muestra información importante basada en la ejecución de tu app en dispositivos reales para ayudarte a comprender y solucionar problemas de estabilidad y rendimiento. Si mejoras las estadísticas de Android vitals, optimizarás la experiencia del usuario y obtendrás una mayor participación y mejores calificaciones.

Conclusión

Las conexiones punto a punto se utilizan normalmente para conectar entre sí dos sistemas (los cuales pueden ser dispositivos móviles) dentro de una red de área amplia (WAN). Una conexión punto a punto es muy útil ya que sirve para llevar los datos del sistema local a un sistema remoto o bien de una red local a una red remota.

Hay diferentes formas para realizar una conexión punto a punto, pero una de las recomendadas por la documentación de Android es a través de WiFi-direct ya que permite a una aplicación encontrar rápidamente dispositivos cercanos e interactuar con ellos, en un rango que supera incluso la capacidad de Bluetooth, todo esto a través de WifiP2pManager.

En cuanto a la publicación de una app móvil a través de una tienda (en este caso Google Play), pude observar que se tiene que tomar en cuenta muchos puntos como puede ser la localización, licencias, facturación, publicidad, etc. También hay que tener una cuenta de desarrollador, además de hacer las suficientes pruebas de compilación y depuración para asegurar que la app se funcional en un dispositivo móvil Android, por lo que a través de esta investigación pude comprender como funciona el lanzamiento de las apps Android.

Referencias

Cómo crear conexiones P2P con Wi-Fi directo | Desarrolladores de Android |.

(s. f.). Android Developers. <https://developer.android.com/training/connect-devices-wirelessly/wifi-direct?hl=es-419>

GeeksforGeeks. (2022, 21 septiembre). *What is P2P(Peer-to-Peer Process)?*

<https://www.geeksforgeeks.org/what-is-p2ppeer-to-peer-process/>

Google Play |. (s. f.). Android Developers.

<https://developer.android.com/distribute/best-practices/launch/launch-checklist?hl=es-419>