16 Remote Desktops für Network-Computing

Anwender können lokal an einem Computer arbeiten (lokales Computing) oder ihr Endgerät über ein Netz mit einem Computer verbinden, auf dem ihre Anwendungen ablaufen (Network Computing).

Bei lokalem Computing sind die Anwender-Peripheriegeräte Bildschirm, Maus, Tastatur, Lautsprecher, Drucker, Scanner usw. direkt an den Computer angeschlossen, der die Anwendungen ausführt.

Wenn Anwender vom Home-Office oder mobil auf zentralen Büroanwendungen arbeiten sollen, benötigen sie sichere Weitverkehrszugänge auf diese Anwendungen. Dieses Kapitel gibt einen Überblick über freie und kommerzielle Lösungen.

Beim Network Computing sind die Peripheriegeräte an ein weiteres Endgerät (ein Terminal) angeschlossen, das seinerseits seine Benutzerein- und -ausgaben über ein Netz mit einem anderen Computer austauscht, auf dem die Anwendungen laufen.

Damit Computer, auf denen Anwendungen laufen, und die Endgeräte der Benutzer ihre Ein- und Ausgabedaten austauschen können, verwenden sie ein Übertragungsprotokoll. Hier finden Sie Informationen

- über freie, standardisierte und frei zugänglich dokumentierte Protokolle aus der Unix/Linux-Welt und
- proprietäre, durch Urheberrecht geschützte und erst seit kurzem öffentlich dokumentierte Protokolle von Microsoft sowie
- über das plattformunabhängige Virtual Network Computing (VNC)

Bei Unix/Linux ist es seit Mitte der 90er Jahre des letzten Jahrhunderts selbstverständlich, dass man von einem beliebigen Rechner aus eine Desktop-Sitzung auf einem anderen PC nutzen kann. Das Massachusetts Institute of Technology (MIT) hatte schon damals auf der Basis der Vorgängerprojekte »V« und »W« das X-Window-System als grafische Benutzeroberfläche für Unix-Systeme entwickelt. Der Anwendungsserver und das Benutzer-Endgerät kommunizieren dabei über das X.11-Protokoll. Dabei sind die Begriffe Server und Client anders als gewohnt definiert, da sie den Blickwinkel der Anwenders und nicht des Rechenzentrums verwenden:

- Am Benutzer-Endgerät läuft ein X-Server, der Benutzereingaben (Maus, Tastatur) entgegennimmt und Ausgaben auf seinem Bildschirm darstellt.

- Anwendungen, sogenannte X-Clients wie Textverarbeitung, Mail etc., nutzen diese grafische Oberfläche, indem sie von ihr die Benutzereingaben erhalten und ihr Daten zur Ausgabe auf deren Grafikbildschirm schicken.

Den Datenverkehr zwischen X-Server und X-Client reduziert das Protokoll NX (s. unten) von NoMachine, einem Unternehmen des italienischen Softwarehauses Medialogic.

Microsoft machte Ende des letzten Jahrhunderts sein Serverbetriebssystem Windows NT 4 Server Terminal Server Edition mit Hilfe von Citrix transportfähig und pflegt diese Eigenschaft bei seinen Serverbetriebssystemen Windows 2000 Server, Windows 2003 Server und Windows Server 2008. Zudem hat Microsoft seine Desktop-Betriebssysteme Windows XP und Vista für je eine einzelne Sitzung ein wenig nach außen geöffnet, damit Anwender von zu Hause auf ihren XP- und Vista-Bürorechnern weiterarbeiten können, ohne dafür zusätzliche freie oder kommerzielle Software zu benötigen.

Für den Datenaustausch zwischen eigenen Terminalservern sowie nach außen geöffneten XP/Vista-Desktop-Sitzungen einerseits und den Benutzer-Endgeräten andererseits setzt Microsoft sein proprietäres Remote Desktop Protocol (RDP) ein. Dieses unterscheidet sich von dem Citrix-Protokoll Independent Computing Architecture (ICA) für die Kommunikation zwischen dessen Terminalservern (Citrix Presentation Server, heute Citrix Xen App) und Benutzer-Endgeräten.

Die Schnittstellen für X.11 und NX für Benutzer-Endgeräte sind öffentlich dokumentiert. Microsoft hat die Spezifikationen für sein RDP inzwischen auf seinem MSDN (Microsoft Developer Network) veröffentlicht.

Auch schon eine relativ lange Tradition hat das plattformunabhängige Virtual Network Computing (VNC). Auch diese ursprünglich von Olivetti und dem Oracle Research Laboratory entwickelte und inzwischen quelloffene Lösung zeigt den Bildschirm des Rechners, auf die VNC-Serversoftware läuft, auf einem Rechner, auf dem die VNC-Viewer-Software arbeitet, an und sendet Maus- und Tastatureingaben an den VNC-Server. Es ist immerhin so schnell, dass man damit auf Windows-PCs das Fernsehbild und den Ton einer mit Linux gepatchten Satellitenfernseh-D-Box genießen kann.

16.1 Überblick

Lesen Sie im Abschnitt 16.2, wie X-Anwendungen ihre Dialoge auf entfernte Endgeräte zaubern, und im Abschnitt 16.3, wie das mit Nomachine NX noch etwas schneller geht. Probefahrten mit Nomachine NX können Sie über die Adresse `http://www.nomachine.com/testdrive.php` starten. Erfahren Sie dann im Abschnitt 16.4 mehr über den Einsatz des Remote Desktop Protokol (RDP) von Microsoft und wie man damit entfernte Desktops nutzen kann. Microsofts arbeitet stetig an seinem Remote Desktop Protocol weiter. Mit Windows Vista wurden die Versionen RDP 6.0 und 6.1

eingeführt. Diese bieten mehr Farbtiefe und mehr Funktionen als ihre Vorgänger und verschlüsseln den Datenverkehr anders als diese. Bei Redaktionsschluss unterstützten nur Client-Programme für Windows XP und Vista alle Funktionen dieser Version des RDP-Protokolls.

Schließlich vermittelt Abschnitt 16.5 eine Idee vom plattformunabhängigen Virtual Network Computing (VNC).

16.2 X.11-Programme im Remote-Betrieb

Theoretisch ist es X-Programmen völlig egal, ob sie ihre Daten lokal oder remote über ein Netz übertragen, also ob der X-Server auf dem gleichen PC läuft oder auf einem weit entfernten. Bei X-Programmen tauschen der X-Client und der X-Server über das Protokoll X.11 Fragen und Antworten miteinander aus und warten dabei stets höflich auf alle Antworten auf jede Frage. So eine Frage mit Antwort heißt auch Roundtrip.

Ein- und Ausgaben können dabei viele oder wenige Dialoge erfordern. So kann der X-Client für ein animiert aufklappendes Menü

- den X-Server jeweils eine Pixelzeile schreiben lassen und ihn dann fragen, ob er das auch wirklich getan hat, und dann auf die positive Antwort warten, bevor er ihm die nächste Pixelzeile schickt oder
- den X-Server pro 500stel Sekunde oder in noch kürzeren Abständen eine Zeile schreiben lassen und nach dem Aufbau des gesamten Menüs fragen, ob er damit fertig ist. Das erfordert nur einen Roundtrip.

Führt man diese beiden Menüvarianten lokal auf einem PC aus, wird man kaum Unterschiede feststellen, da hier die Unix Domain Sockets die Wege zwischen X-Client und X-Server überbrücken. Über ein schmalbandiges WAN (Wide Area Network) werden Benutzer hingegen bei der ersten Version unnötig lange auf die Übertragung der vielen Frage-Antwort-Spielchen zwischen X-Client und X-Server warten müssen. Testen Programmierer ihre Anwendungen nur lokal, merken sie nicht, wenn ein X-Programm viele solcher Roundtrips enthält. Dabei sind die Programmierer daran nicht allein schuld, da viele dieser Roundtrips eine Folge der Verwendung von Programmbibliotheken sind, die für lokale Anwendungen optimiert sind und sich nicht darum scheren, wenn bei Ausführung über ein WAN der Mauszeiger zuckelt und die Animationen ruckeln. Durch das sinnlose Warten schrecken sie leider Anwender ab.

Die folgenden Abschnitte zeigen Linux-Einsteigern und Administratoren, dass sie keinen zweiten Bildschirm brauchen, um mehrere Desktops gleichzeitig zu sehen und wie sie zeitraubende Administration und Betreuung von Servern und Desktops vor Ort reduzieren können.

16.2.1 Remote X.11

Dieser Abschnitt vermittelt Ihnen einen Überblick zu X.11-Anwendungen im LAN und WAN. Das an der Eliteuniversität Massachusetts Institute of Technology (MIT) Ende der 1980er Jahre entwickelte X.11-Protokoll zielte von Anfang an auf Netzwerktransparenz: Es sollte die grafischen Ausgaben entfernter Anwendungen im Netz auf der eigenen grafischen Konsole darstellen können. X.11 kann hierzu Daten über so genannte UNIX Domain Sockets lokal oder entfernt über ein Netzwerk ausgeben.

X-Programme kommunizieren über das flexible X.11-Protokoll mit einem X-Server, um ihre grafische Ausgabe auf einem Bildschirm darzustellen und von ihm die Maus- und Tastatureingaben der jeweiligen Nutzer entgegenzunehmen. Der X-Server übernimmt die Kommunikation mit der Hardware der darstellenden Maschine.

Es gibt zwei Methoden, um X.11 über ein Netz zu nutzen:

- Sie stellen einzelne Anwendungen einer entfernten Maschine auf Ihrem X.11-Desktop lokal dar. Dieses gelingt auf zwei Wegen: Einerseits durch das direkte Ansprechen Ihres lokalen X-Servers und durch das Setzen der `DISPLAY`-Variablen in der Shell-Umgebung auf der entfernten Maschine, an der Sie angemeldet sind. Alternativ übertragen Sie die Aufgabe an die Secure Shell SSH, mit dem Sie sich entfernt angemeldet haben. Diese baut beim Setzen der `-X` oder `-Y` Option einen gesicherten Tunnel auf.
- Der Hintergrunddienst *Displaymanager* übernimmt den Export eines kompletten grafischen Unix-Desktops. Unter SUSE stehen hierzu die Programme `kdm`, `gdm` oder `xdm` bereit.

16.2.2 Einzelne Applikationen exportieren

Die klassische Methode besteht darin, dass eine entfernte Applikation direkt Ihren ersten X-Server anspricht, der hierzu auf Port TCP 6000 lauscht und Verbindungen entgegennimmt. Dies erfordert jedoch, wie im nächsten Abschnitt gezeigt, eine gewisse Vorbereitung. Auf einer Firewall muss der genannte Port offen sein. Ein Beispiel könnte so aussehen, dass Sie zuerst die Maschine freigeben, von der grafische Applikationen kommen. Anschließend loggen Sie sich auf dieser ein und setzen die Umgebungsvariable `DISPLAY`, damit jede nun gestartete Applikation ihre Ausgabe umlenkt:

```
alkalde@lokale_maschine:~> xhost entfernte_maschine
alkalde@lokale_maschine:~> ssh debacher@entfernte_maschine
Last login: Sat Oct  4 19:41:02 2008 from
lokale_maschine.mydomain.site
Have a lot of fun...
debacher@entfernte_maschine:~> export DISPLAY=lokale_maschine:0.0
debacher@entfernte_maschine:~> xterm &
```

Für eine gesicherte SSH-Verbindung brauchen Sie keine Freigabe für Port 6000. Sie melden sich einfach mit der Option `-X` auf der entfernten Maschine an und starten das gewünschte Programm:

```
alkalde@lokale_maschine:~ > ssh -X -l root entfernte_maschine
Password:
Last login: Wed Oct  1 16:00:08 2008 from
lokale_maschine.mydomain.site
Have a lot of fun...
entfernte_maschine:~ # yast2 &
```

Diese kleine Befehlsabfolge können Sie dazu nutzen, um sich als Systemadministrator mit einem entfernten Linux-PC zu verbinden und auf ihm YaST2 zu starten. Als Ergebnis zeigt Ihr lokaler Desktop die grafische Oberfläche dieses Programms als normales Fenster an. Sie können nun mit dem Programm YaST2 auf `entfernte_maschine` genauso arbeiten, als würden Sie direkt vor diesem PC sitzen. So können Sie diesen Rechner bequem grafisch administrieren, auch wenn er beispielsweise in einem gesicherten Serverraum steht.

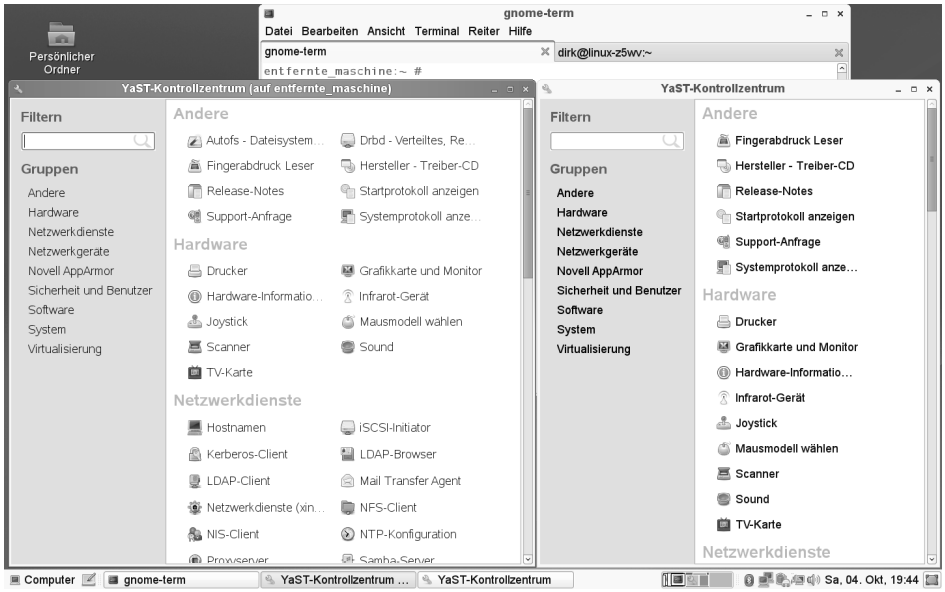

Abbildung 15.21: Das entfernt aufgerufene Yast erkennen Sie an der Titelzeile.

Hinweis: Nutzen Sie den ungesicherten X.11-Export nur in vertrauenswürdigen Netzen und vergessen Sie nicht die Freigabe in der Firewall.

16.2.3 Displaymanager nutzen

Sie können nicht nur die grafische Ausgabe einzelner Applikationen von der entfernten auf Ihre Maschine exportieren, sondern ebenso komplette Desktops. Hierzu müssen Sie einige Details vorbereiten, wenn noch kein PC in Ihrem Netzwerk einen Displaymanager offen anbietet. Andernfalls laufen die Kontaktversuche Ihres X-Servers ins Leere. Unter OpenSUSE 11.0 konfigurieren Sie das Verhalten des Displaymanagers in der Datei `/etc/sysconfig/displaymanager`:

```
...
DISPLAYMANAGER="gdm"
...
DISPLAYMANAGER_REMOTE_ACCESS="no"
...
DISPLAYMANAGER_ROOT_LOGIN_REMOTE="no"
...
DISPLAYMANAGER_XSERVER_TCP_PORT_6000_OPEN="no"
...
```

Die erste gezeigte Variable bestimmt, welcher Displaymanager durch das Startskript `/etc/init.d/xdm` aufgerufen wird. Sinnvolle Werte sind hier `gdm`, `kdm`, `xdm`. Sie sollten sicherstellen, dass Sie für `gdm` die Gnome-Basispakete und für `kdm` die KDE-Basispakete installiert haben. Die folgende Zeile bestimmt, ob Displaymanager entfernte Anfragen für den Export von X.11-Sessions zulassen. Dieses benötigen Sie, wenn Sie für X-Terminals unter LTSP (siehe Kapitel 17) den Zugriff auf Ihren Server einrichten wollen. Der Default steht auf `no`, um Exporte zu erlauben, tragen Sie hier `yes` ein. Damit öffnet der Displaymanager den Port 177 UDP, den Sie deshalb ebenfalls in Ihrer Firewall freigeben müssen. In der vorletzten gezeigten Zeile können Sie festlegen, ob sich der Systemadministrator anmelden darf. Wenn nur eine ungesicherte Verbindung zur Verfügung steht, sollten Sie von einer Freigabe absehen. Die letzte Zeile erlaubt Ihnen den Import von entfernten X.11-Ausgaben auf Ihrem Desktop.

Hinweis: Nach dem Ändern dieser Datei müssen Sie die SUSE-Konfiguration erneuern und den Displaymanager neu starten: `SuSEconfig; rcxdm restart`

Nach der Freigabe für den entfernten Displaymanager-Zugriff können Sie dieses nun ausprobieren. Hierzu eignet sich das Programm `Xnest`:

```
Xnest :1 -query localhost
```

Dieses ist ein spezieller X-Server, der als Fenster Ihres grafischen Desktops läuft. Da Ihr Hauptserver bereits auf `:0` aktiv ist, geben Sie als Option zusätzlich `:1` an.

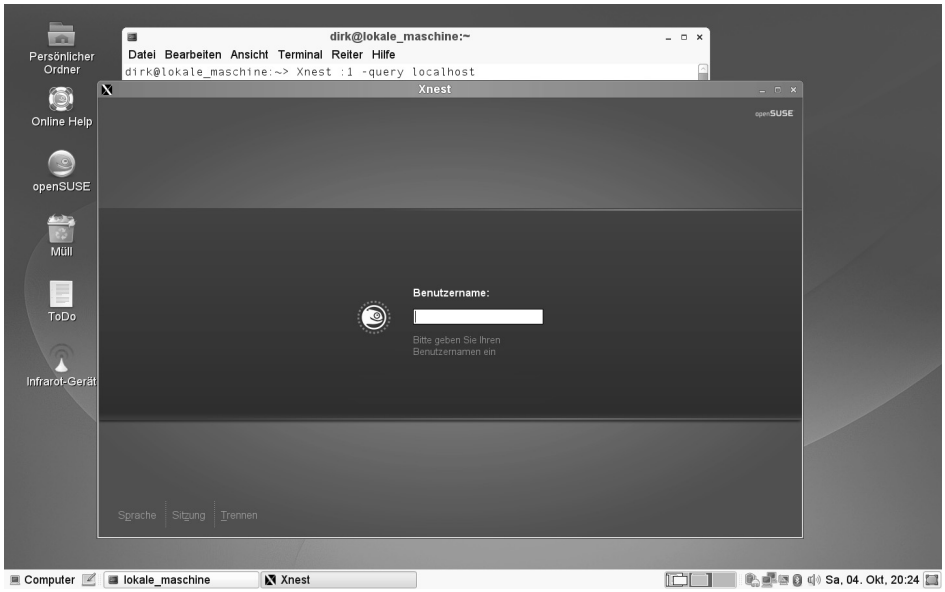

Abbildung 16.1: Xnest auf einem Linux-Desktop im Einsatz

Auf einem anderen Gerät, auf dem Sie die Grafikausgabe sehen wollen, –beispielsweise einem X-Terminal – starten Sie den X-Server in eben dieser Weise, nur dass Sie statt `localhost` den Namen des entfernten Servers eintragen:

```
X -query ihr_server
```

Wenn Sie den ersten antwortenden Server ohne eine Präferenz ansprechen möchten, können Sie auch einfach `X -broadcast` aufrufen. Möchten Sie Ihren Usern eine Auswahlliste anbieten, kann Ihnen dieses der Server mittels:

```
X -indirect ihr_server
```

anbieten.

16.3 Nomachine NX

Bei X-Programmen tauschen der X-Client und X-Server über das Protokoll X.11 Fragen und Antworten miteinander aus und warten dabei auf alle Antworten auf jede Frage. So eine Frage mit Antwort heißt auch Roundtrip . Bereits oben wurde am Beispiel eines animiert aufklappenden Menüs erklärt, dass ein- und dieselbe Ausgaben viele oder wenige Dialoge erfordern kann.

X-Programme wissen bei ihrer Ausführung auf einem X-Client nicht, wo sich der zugehörige X-Server befindet. Sie können bei einem über ein WAN erreichbaren X-Server nicht selbständig Roundtrips zwischen X-Server und X-Client einsparen. Auf dieses

Problem zielt die Erweiterung NX des X.11-Protokolls der Division Nomachine von Medialogic S.p.A..

NX beschleunigt den Aufbau von X-Bildschirmen durch

- Cachen (Speichern und aus dem Speicher wieder verwenden) bereits einmal übertragener Daten und differentielle Übertragung,
- Einsparen der meisten X-Roundtrips sowie
- Verdichten der verbleibenden X-Kommunikation.

Durch diese Effizienzverbesserung können Nutzer mit Network-Computing zufriedener sein als bei Technologien wie reinem X.11. Nomachine NX unterstützt Linux-, Solaris- und Windows-Anwendungsserver sowie Windows-, Solaris-, Linux- und MacOS/X- Clients.

Nomachine setzt auf dem in der Unix-Welt bewährten Protokoll X.11 auf. Es schaltet je einen Stellvertreter (Proxy) zwischen das X-Programm auf dem X-Client und den X-Server auf dem Benutzer-Endgerät. Es spart in mehreren Schritten überflüssigen X-Verkehr:

- Mit dem X-Programm auf dem Linux-Anwendungsserver (X-Client) spricht der NX-Proxy die Sprache von X.11 und täuscht ihm so vor, er sei der zuständige X-Server. Das Programm *nxagent* übersetzt auf dem NX-Proxy des Anwendungsservers als bildschirmloser Schatten-X-Server.das X-Protokoll in das NX-Protokoll.
- Der NX-Proxy speichert alle Nachrichten zwischen (Caching) und überträgt nur Unterschiede. Er eliminiert unnötige Frage-Antwort-Spiele und komprimiert den restlichen Datenverkehr. Der NX-Proxy auf dem Linux-Anwendungsserver leitet dann alle empfangenen Daten des X-Programms per NX-Protokoll an den NX-Proxy auf dem Endbenutzer-PC weiter. Auf dem Verbindungsabschnitt vom NX-Proxy auf dem Linux-Terminalserver zum NX-Proxy auf dem Benutzer-Endgerät braucht NX durch die drei Sparmaßnahmen sehr wenig Bandbreite.
- Der NX-Proxy auf dem Benutzer-Endgerät benimmt sich gegenüber dem dort laufenden X-Server wie eine ganz normale X.11-Anwendung: der NX-Proxy und der X-Server sprechen wieder das ganz normale X-Protokoll über schnelle UNIX Domain Sockets. Der X-Server stellt Benutzern die Ausgaben des X-Programms auf dem Bildschirm des Client-PCs dar und nimmt Eingaben entgegen.

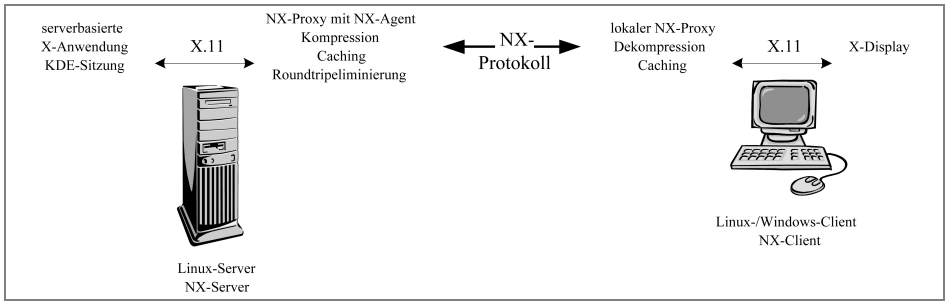

Abbildung 16.2: Komponenten von NX am Anwendungsserver und am Benutzerarbeitsplatz

16.4 Microsofts Remote Desktop Protocol

Microsoft verwendet sein Remote Desktop Protocol (RDP) für Remote Desktop Connections (RDC) sowohl bei den Terminaldiensten seiner Terminalserver als auch beim Fernzugriff auf XP-und Vista-Arbeitsplatzrechner. Die IT-Industrie nutzt letzteren Fernzugriff auf Arbeitsplatzrechner, um Anwendersitzungen auf Desktop-Rechnern zentral und automatisiert aus Rechenzentren bereitzustellen.

16.4.1 Einsatzgebiete von RDP/RDC

Mit dem Remote Desktop Protocol liefert Microsoft seit Ende des letzten Jahrtausends seine Terminaldienst-Sitzungen von den Terminalserver-Versionen seines Serverbetriebssystems an Benutzer-Endgeräte. Um von einem Server aus mehrere Benutzer-Sitzungen bereitstellen zu können, hatte Microsoft dazu die sogenannte Multiwin-Techonologie von Citrix lizenziert, nicht aber deren Übertragungsprotokoll Independent Computing Architecture (ICA).

Die Multiwin-Technologie erlaubt es, auf Windows-Servern getrennte Benutzersitzungen ablaufen zu lassen und deren Benutzer-Dialoge aufzusplitten.

Microsoft bietet die Terminaldienste seiner Windows-Server für drei Anforderungen von Unternehmen an:

- Fernwartung der Windows-Server,
- Mit wenig Aufwand zu wartende und sicherere Anwendersitzungen als Antwort auf die ausufernden Gesamtkosten von Anwender-PCs und dabei
- Benutzerbetreuung durch gleichzeitiges Darstellen und Steuern der Anwendersitzungen auf einem Endgerät im Helpdesk.

Die Terminaldienste erlauben, viele populäre Windows-Programme auf Windows-Servern ablaufen zu lassen und über das RDP-Protokoll an Benutzer-Endgeräte zur Darstellung zu übermitteln. Da Windows-Anwendungen aber als Ein-Benutzer-Pro-

gramme entwickelt sind und teils direkt auf PC-Hardware zugreifen wollen, lassen sich etliche Windows-Programme selbst mit erheblichem Programmier-Aufwand nicht über Terminaldienste bereitstellen. Als Alternative zu Terminaldiensten bietet sich inzwischen der Fernzugriff auf XP/Vista-Rechner an.

Seit Windows XP verwendet Microsoft sein Remote Desktop Protocol auch für den Fernzugriff (Remote Desktop Connection – RDC) auf Anwender-PCs. Damit können Anwender vom zu Hause aus ihrem Home Office oder von unterwegs auf offene Sitzungen ihres Büro-PCs zugreifen, ohne wegen der Einschränkungen von Terminaldiensten auf eine Auswahl der auf Terminalservern lauffähigen Windows-Programme angewiesen zu sein. Dieser Fernzugang wird inzwischen mehr und mehr dazu genutzt, um Desktops mit XP und Vista zentral aus Rechenzentren bereitzustellen.

16.4.2 Technik von RDP

RDP basiert auf dem Protokoll ITU-T T.128 der International Telecommunications Union (ITU) für Echtzeit-Datenverbindungen, auch T-Share genannt. Es verwendet den Port 3389 des Netzwerk-Transportprotokolls Transmission Control Protocol/Internet Protocol (TCP/IP) für das Übermitteln von Daten. Im Open Systems Interconnection Reference Model (OSI-Schichtenmodell) deckt es die Ebenen 4 (Transportschicht), 5 (Sitzungsschicht), 6 (Darstellungsschicht) und 7 (Anwendungsschicht) ab.

Die Version 6.0 von RDP hat Microsoft im Jahr 2007 mit Windows Vista eingeführt, die Vorgänger 5.2 mit dem Service Pack 2 von Windows XP und 5.1 zuvor mit dem Service Pack 1 von Windows XP. Bei Redaktionsschluss aktuell ist RDP 6.1. Gegenüber der Fünfer-Version hat Microsoft u. a. die Farbtiefe des Bildschirms erhöht und das Verschlüsselungsverfahren geändert.

16.4.3 RDC-Clients für RDP

Den vollen Funktionsumfang von RDP nutzen die Client-Programme von Microsoft für ihre eigenen Desktop-Betriebssysteme Windows XP und Windows Vista sowie deren funktional leicht eingeschränkte Embedded-Versionen für Terminals, nämlich XP Pro und Vista Business/Ultimate für Embedded Systems:

- Die Windows XP Remote Desktop Connection-Software `msrdpcli.exe` mit dem Funktionsumfang von RDP 5.2 unterstützt Clients ab Windows 95.
- Terminal Services Clients 6.1 für Windows XP laufen auf Clients mit den Service Packs 2 oder 3 von Windows XP.
- Im Service-Pack 1 für Vista ist ein Terminal Services Client 6.1 für RDC enthalten.

Freie Entwickler (wie von *rdesktop*) und Softwarehäuser, die Client-Software für RDP entwickeln (wie HOB), können inzwischen dank der massiven Intervention durch die Europäische Union auf eine Dokumentation des Protokolls von Microsoft zugreifen

und arbeiten daran, möglichst viele Merkmale dieses ausgesprochen komplexen Protokolls zu unterstützen. Für Linux–Server bietet HOB auch einen Formatwandler von X.11 auf RDP.

16.4.4 Remote-Desktop-Verbindung

Erste Erfahrungen mit Remote-Verbindungen können Sie mit Windows-Terminaldiensten von Windows-Servern oder mit XP- oder Vista-Desktops sammeln.

Microsoft nennt den Fernzugriff von einem anderen Arbeitsplatz auf einen Windows-Desktop-PC eine Remote-Desktop-Verbindung.

Microsoft gestattet solche Remote-Desktop-Verbindungen nur auf die teureren Versionen seiner Desktop-Betriebssysteme Windows XP Professional und Windows Vista Ultimate und Enterprise.

Lesen Sie hier zuerst, was Sie auf dem Windows-PC mit Vista Ultimate tun müssen, um eine Remote-Verbindung auf ihn zuzulassen und dann, wie Sie aus der Ferne mit einem Windows- Endgerät eine Remote-Sitzung darauf ausführen.

16.4.5 Einrichten des Remotezugang auf den Windows-Desktop

Um eine Remote-Zugang zu einem Vista-Ultimate-PC einzurichten, muss man

- sicherstellen, dass dieser PC eingeschaltet bleibt und sich nicht in einen Stromsparmodus schlafen legt, während jemand darauf aus der Ferne zugreifen will,
- Remote-Desktop-Verbindungen zulassen,
- festlegen, welche Benutzer die Verbindung verwenden können und
- gegebenenfalls in einer Brandmauer einen TCP-Port freischalten.

Zunächst geht es mit einer nur sechsschrittigen Befehlsfolge zu den Dialogen für die Remote-Einstellungen:

Start • *Einstellungen* • *Systemsteuerung* • *System und Wartung* • *System* • *Remoteeinstellungen*

Um dahin zu kommen, müssen Sie auf dem Weg dahin bei der Benutzerkontensteuerung die Option *Fortsetzen* wählen. Schließlich kommen Sie dann hoffentlich zum Dialogfenster *Systemeigenschaften.* Hier interessiert nur der Reiter *Remote.*

Abbildung 16.3: Dialog zum Freigeben von Remoteverbindungen

Im Reiter *Remote* geht es um zwei ganz verschiedene Remote-Verbindungen: die *Remoteunterstützung* erlaubt gegenseitigen Support und der *Remotedesktop* das Arbeiten an diesem Computer aus der Ferne.

In der Voreinstellung ist die Option *Remoteunterstützungsverbindungen mit diesem Computer zulassen* aktiv; *Remotedesktop-Verbindungen* sind nicht zugelassen.

Um Letzeres freizugeben, müssen Sie sich entscheiden, ob Sie die Remote-Verbindung

- nur mit Endgeräten, auf denen eine aktuelle Version von Microsofts Remote Desktop Connection für Windows XP oder Windows Vista läuft oder
- auch mit freien RDP-Clients wie *rdesktop*, mit RDP-Clients anderer Hersteller oder mit RDP-Clients für ältere Windows-Versionen

nutzen wollen.

Im ersten Fall wählen Sie im Reiterfeld *Remotedesktop* den dritten Radioknopf und im zweiten Fall den zweiten Radioknopf *Verbindungen von Computern zulassen, auf denen eine beliebige Version von Remotedesktop ausgeführt wird.*

Wenn Sie auf dem Desktop-PC, auf dem Sie eine Remote-Verbindung zulassen wollen, eine Energiesparoption erlaubt haben, warnt Sie Vista jetzt in einem neuen Dialogfenster *Remotedesktop* und zeigt Ihnen gleich den Weg, um das zu ändern.

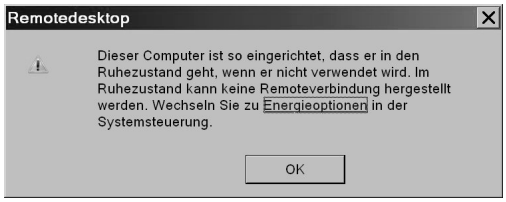

Abbildung 16.4:
Vista warnt vorm Energiesparen

Klicken Sie hier auf die Schaltfläche *Energieoptionen*, um direkt in das Dialogfenster *Systemsteuerung\Hardware* und *Sound\Energieoptionen* zu kommen.

Entscheiden Sie sich hier links für die Option *Energiesparmodus ändern*. Dies bringt Sie in das Dialogfenster *Energiesparplaneinstellungen bearbeiten*. Klappen Sie hier die Liste von *Energiesparmodus nach* auf und entscheiden Sie sich für die Option *Niemals* und klicken Sie die Schaltfläche *Änderungen speichern*. Jetzt sollte Ihr Vista Ultimate-PC immer für Remote-Sitzungen bereit sein und ungehindert weiter an Ihrem Hauptarbeitsplatz Strom ziehen, egal ob jemand daran direkt oder aus der Ferne arbeitet oder nicht.

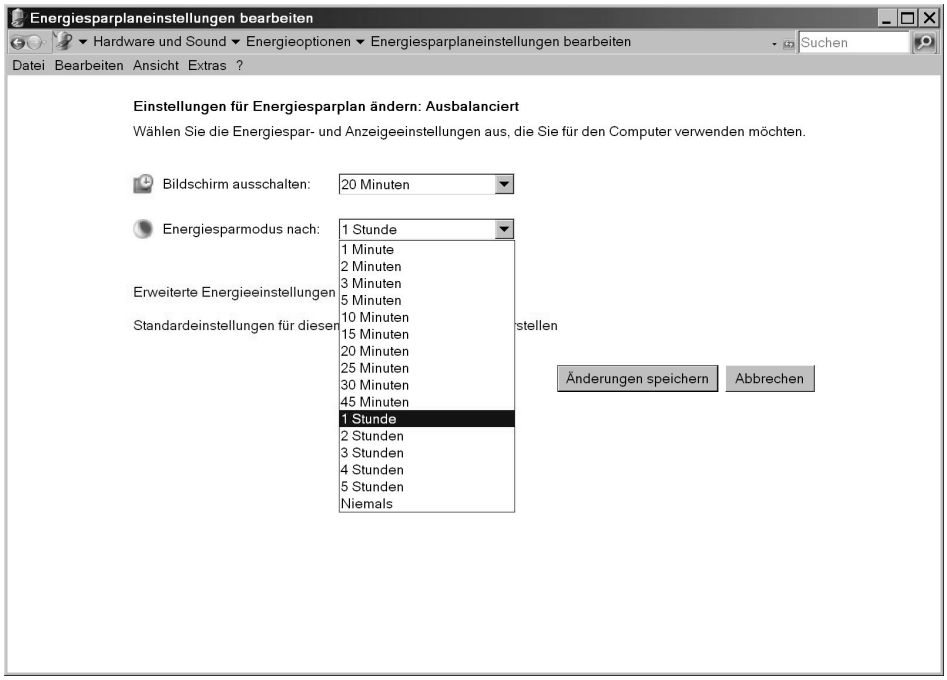

Abbildung 16.5: Für Remotezugang darf der Vista-Desktop nicht schlafen oder Strom sparen

Um jetzt Benutzern den Remote-Zugriff auf den Vista Ultimate-Desktop zu erlauben, müssen Sie zurück in den Reiter *Remote* der *Systemeigenschaften* und dort im Bereich *Remotedesktop* zur Schaltfläche *Benutzer auswählen* …. Das bringt Sie in das Fenster

Remotedesktopbenutzer. Dieses zeigt, wer schon Remote-Verbindungen nutzen darf und erlaubt Ihnen, weitere Benutzer dafür freizuschalten..

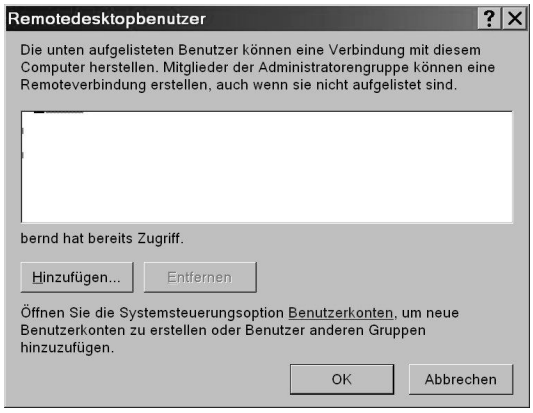

Abbildung 16.6: Hier eingetragene Benutzer dürfen aus der Ferne zugreifen

Über die Schaltfläche *Hinzufügen…* können Sie weiteren Benutzern den Remote-Zugriff auf diesen PC erlauben. Aus diesem Fenster heraus können Sie auch einen Ausflug zur Benutzerverwaltung unternehmen und dort weitere Benutzer anlegen, bevor Sie ihnen den Remote-Zugriff erlauben. Dieser Ausflug ist auch sinnvoll, wenn die Remote-Benutzer noch kein Passwort haben sollten. Das sollten Sie unbedingt einrichten. Zurück im Reiter *Remote* des Dialogs *Systemeigenschaften* bestätigen Sie mit den Schaltflächen *Übernehmen* und dann mit *OK* Ihre Entscheidungen.

Bevor Sie den Remote-Zugang testen, sollten Sie die Bildschirmauflösung des PCs, auf den Sie gleich zugreifen wollen, so niedrig einstellen, dass sie nicht größer ist als die des zugreifenden Geräts, da Sie sonst den Desktop nicht immer richtig sehen und steuern können.

16.4.6 Auf Remote-Sitzung zugreifen

Wenn Sie das alles eingerichtet haben, können Sie den Remote-Zugriff in Ihrem lokalen Netz von einem anderen Gerät, sei es einem Thin Client (siehe nächstes Kapitel) oder einem Windows- oder Linux PC testen.

Für den Test ist es unerheblich, was für ein Endgerät mit welchem Betriebssystem Sie verwenden, so lange es eine Client-Software für das Remote Desktop Protocol mitbringt. Haben Sie allerdings zuvor den Zugriff auf XP/Vista beschränkt, müssen Sie bei einem XP-Endgerät eventuell noch die aktuelle Remote Desktop Connection Software von Microsofts Webseite laden und einrichten. Mit anderen Endgeräte-Clients wird mit dieser Beschränkung der Zugriff nicht klappen.

Um mit einem Endgerät mit Windows XP auf eine Remote-Verbindung zuzugreifen, hangeln Sie sich in drei Schritten durch den XP-Befehlsbaum mit

Start • Zubehör • Remotedesktopverbindung.

Windows XP öffnet dann das kleine Fenster *Remotedesktopverbindung*. Hier können Sie den Namen oder die IP-Adresse des Remote-Computers angeben, mit dem Sie Ihren lokalen PC verbinden wollen.

Abbildung 16.7:
Remotecomputer angeben

Ein Klick auf die Schaltfläche *Optionen* rechts unten im Fenster vergrößert das Fenster und zeigt sechs Reiter, in denen Sie viele Verbindungsdaten erfassen und gestalten können.

Abbildung 16.8:
Anmeldereiter Allgemein des Dialogfensters Remotedesktopverbindung

Unter anderem können Sie im Reiter *Erweitert* die aktuelle Übertragungsrate einstellen.

Abbildung 16.**9:** Einstellen der aktuellen Übertragungsrate

Sobald Sie die Rechner- und Benutzernamen richtig erfasst und die Schaltfläche *Verbinden* angeklickt haben, kommen Sie in den Anmeldebildschirm des Remote-Computers. Oben sehen Sie mittig in einem in die Titelleiste gequetschten Bereich den Namen des Remote-Rechners und rechts Schaltflächen zum Steuern der Fenstergröße und zum Ausloggen.

Abbildung 16.**10:** Anmeldefenster für Remote-Verbindung

Wenn Sie wie hier im Beispiel mit einem Windows-Client mit einer Bildschirmauflösung von 1024 * 786 Punkten auf einen Remote-Rechner mit einer Auflösung von 1920 * 1200 Bildpunkten zugreifen, müssen Sie mit Darstellungsfehlern wie im obigen Bild rechnen.

Wählen Sie den Benutzernamen aus, rollen Sie dann, falls nötig, mit dem rechten Rollbalken ein wenig nach unten, damit Sie den Passwortdialog sehen und geben Sie Ihr Passwort für den Remote-Computer ein. Wenn das alles geklappt haben sollte, sehen Sie wie hier ein Vollbild des Windows Vista Remote-Computers auf dem Bildschirm Ihres darauf zugreifenden Windows XP-Rechners.

Abbildung 16.11: Remote-Vollbild

Mit den hinein gequetschten Schaltflächen oben rechts steuern Sie die Größe des Remote-Bildschirms. Die linke Schaltfläche verkleinert ihn zum Icon und die mittlere verkleinert sein Fenster.

Abbildung 16.12: Remotebildschirmfenster im XP-Desktop

Die Remote-Sitzung schließen Sie wieder, indem Sie oben rechts bei den in das Remote-Fenster hinein gequetschten Schaltflächen *Schließen* anklicken und dann im Fenster *Terminaldienstesitzung trennen* mit der Schaltfläche *OK* Ihre Absicht bestätigen.

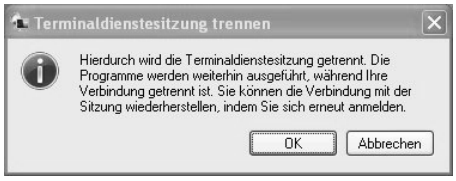

Abbildung 16.13: Verbindung wirklich trennen

Das gleiche erreichen Sie, wenn Sie sich wieder direkt am Remote-Computer anmelden. Hier sehen Sie bei dem remote eingeloggten Benutzer im Anmeldebildschirm den Hinweis *logged in*.

Hat sich auf dem Remote-Computer jemand während einer Remote-Sitzung angemeldet, erfahren Sie dies auf dem aus der Ferne zugreifenden Endgerät durch das Dialogfenster *Remotedesktop getrennt*.

Abbildung 16.14: Die Remotedesktopsitzung wurde durch ein Login auf dem Remotecomputer beendet.

Wollen Sie solche Remote-Sitzungen nicht nur gelegentlich verwenden, sondern zum automatischen Ausrollen und Managen von Windows-Sitzungen aus gehosteten virtuellen XP- oder Vista-PCs einsetzen, können Sie inzwischen vielfältige kommerzielle Angebote, u. a. der Virtual Desktop Infrastructure Alliance, nutzen.

16.5 Virtual Network Computing (VNC)

Das plattformunabhängige Protokoll Virtual Network Computing (VNC) lässt sich für viele Anwendungen nutzen:

- Benutzer bitten Sie aus der Ferne um Hilfe an ihren Desktops.
- Sie wollen einen Windows-Desktop zusätzlich auf Ihrem Linux-PC nutzen oder umgekehrt.
- Sie wollen das Fernsehbild und den Ton Ihrer mit Linux gepatchten Nokia-D-Box auf ihrem Windows- oder Linux-Desktop sehen und hören.

Virtual Network Computing stammt aus dem Jahre 1998. Ursprünglich wurde es an den AT&T Laboratories in Cambridge entwickelt und 2002 in ein ausgegründetes Unternehmen namens RealVNC eingebracht. Neben der kommerziellen Variante, die sich beispielsweise in Fernsteuerlösungen für Serverhardware (Keyboard, Video, Maus) findet, existieren einige freie Weiterentwicklungen und Verbesserungen, wie UltraVNC oder TightVNC. Letzteres ist in der Linux-Welt verbreiteter Standard und steht unter der GPL-Lizenz für freie Software.

VNC erlaubt den Zugriff auf einen gemeinsamen Desktop von verschiedenen Rechnern, die unterschiedliche Betriebssysteme verwenden können. VNC löst das Problem anders als X.11. Es stellt den kompletten Desktop des einen Rechners in einem Fenster auf einem weiteren Rechner dar. VNC ist auf die optimale Nutzung schmaler Bandbreiten getrimmt und hat damit dem einfachen Remote-X.11 einiges voraus. Die benötigte Bandbreite für flüssiges Arbeiten hängt von der CPU (ein etwas älterer Pentium-III reicht aus), von der Bildauflösung sowie der Farbtiefe ab. Passiert nichts, fallen keine Daten an. Je größer jedoch das zu transportierende Bild und je besser die Farbtiefe, desto mehr Daten müssen bei jeder Änderung des Desktop-Inhalts übermittelt werden. Ganz extrem sind Videosequenzen.

Das VNC-Paket besteht aus zwei Komponenten, dem *Server* und dem Client, in der VNC-Sprache *Viewer*. Der Server greift die Daten vom laufenden Desktop ab und kann diese an einen oder mehrere Clients über das Netzwerk weiterreichen. Clients verbinden sich auf laufende VNC-Server und sorgen für die lokale Darstellung des entfernten Desktops in einem Fenster der grafischen Oberfläche, von der sie gestartet wurden.

Während unter Windows genau eine Möglichkeit zur Verfügung steht, gibt es bei Linux zwei Varianten, um einen VNC-Server zu betreiben:

- Entweder Sie starten den Server als eigenen Prozess mit dem Kommando `vncserver`. Dieser Server öffnet keinen grafischen Desktop auf der Maschine, auf der er gestartet wurde. Damit können Sie mehr als einen VNC-Export anbieten. Es bedeutet aber auch, dass eine aktive X.11-Session auf diesem Wege nicht exportiert werden kann.
- Oder Sie schalten das `x11vnc`-Modul in Ihrer X-Server-Konfiguration ein und erlauben damit den entfernten Zugriff auf einen lokal laufenden Desktop. Dieses entspricht dem von Windows gewohnten Verhalten.

Wenn Sie eine dieser VNC-Betriebsvarianten auf Ihrem Linux-Server betreiben wollen, installieren Sie bitte mit YaST die Pakete `tightvnc`, `xorg-x11-Xvnc` und `x11vnc`.

Im Gegensatz zu einer X.11-Sitzung, die mit dem Abschalten des Remote-Displays automatisch alle darin gestarteten Prozesse beendet, kann eine VNC-Sitzung ewig weiterleben. Der Client hängt sich lediglich vom Server ab, veranlasst jedoch nicht dessen Herunterfahren. Damit bleibt der VNC-Server aktiv, auch wenn sich alle VNC-Clients abgemeldet haben und alle in der Session gestarteten Programme laufen weiter. So kommt man bei der erneuten Verbindung zu exakt dem Zustand zurück, an dem der Client beendet wurde. Trotzdem sollten aus Sicherheitsgründen alle offenen Dateien gesichert werden, bevor der Client geschlossen wird.

Einen Nachteil kann das Konzept jedoch haben: Bieten Sie Ihren Nutzern einen gemeinsamen VNC-Login-Server an – eine Maschine, die viele VNC-Server hostet –, benötigt die Maschine unter Umständen sehr viel Speicher und einiges an Ressourcen. Beenden Ihre Benutzer ihre grafischen Sitzungen nicht, liegen durchaus etliche Megabyte wartend im Speicher und einige offene Applikationen verbrauchen sogar noch Rechenzeit.

16.5.1 Unabhängiger VNC-Server

Nach der Installation der oben genannten Pakete können Sie sofort loslegen. Hierzu starten Sie als erstes `vncserver`. `vncserver` ist ein sogenanntes Wrapper-Skript, das im Hintergrund den eigentlichen Server `Xvnc` aufruft:

```
linux:~ #  vncserver
You will require a password to access your desktops.
```

```
Password:
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Verify:

New 'X' desktop is linux:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/linux:1.log
```

Diese Kommandos zum Start eines VNC-Servers können nicht nur System-Administratoren, sondern alle Benutzer ausführen, da VNC keine privilegierten Ports belegt. Einen VNC-Server für Desktop 1 beenden Sie mit `vncserver -kill :1`, für weitere Server zählen Sie die Desktop-Nummer entsprechend hoch. Das funktioniert als Nicht-Systemadministrator natürlich nur für die eigenen Sessions. Wollen Sie alle aktuell laufenden VNC-Server auf einmal beenden, hilft ein `killall Xvnc`.

Der Aufruf von `vncserver` legt im Heimatverzeichnis des Benutzers, der ihn gestartet hat, im versteckten Verzeichnis `.vnc/xstartup` eine Standardkonfiguration für die im VNC laufende X-Session an:

```
#!/bin/sh
xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
twm &
```

Diese Vorgabe werden Sie in den meisten Fällen wegen des voreingestellten `twm` ändern wollen. Der Tab Window Manager (TWM) ist zwar sehr kompakt, schnell und beim X.11-Paket immer dabei, aber alles andere als benutzerfreundlich.

Nach dem erfolgreichen Start steht auf dem Linux-Server ein VNC-Server in seiner Voreinstellung zur Verfügung. Dabei beachten Sie bei jedem Start die Ausgabe des `vncserver: linux:1`. Das Skript wirft eine Display-Nummer aus (`:1`, `:2`, ...), die für den späteren Zugriff auf den Server benötigt wird. Auf einem Linux-PC mit laufender grafischer Oberfläche beginnt die Zählung mit 1; läuft kein X.11, stünde hier eine 0. Die Portnummerierungen der VNC-Server selbst beginnen mit 5800 für den Browser-Java-Applet-Zugriff und bei 5900 für den Zugriff per `vncviewer`. Zu dieser Basiszahl zählen Sie jeweils die Display-Nummer hinzu. Damit können Sie sehr einfach von fast überall eine Desktop-Sitzung auf einer entfernten Maschine einleiten bzw. wiederaufnehmen. Unter Windows steht ein gestarteter VNC-Server üblicherweise unter Port 5900 zur Verfügung.

Haben Sie keine DNS-Namensauflösung konfiguriert, kann statt des Rechnernamens auch die IP angesprochen werden. Aus Sicherheitsgründen sollten Sie immer ein Passwort für den Zugriff festlegen: Beim Erststart fragt Sie `vncserver` selbst nach dem

Passwort. Später können Sie zum Ändern und Neusetzen das Kommando `vncpasswd` verwenden. Das VNC-Passwort ist dabei komplett unabhängig von irgendwelchen System- oder Login-Passwörtern. Es wird verschlüsselt in der Datei `.vnc/passwd` im jeweiligen Heimatverzeichnis des aufrufenden Benutzers abgelegt.

Die Netzwerkanforderungen des klassischen VNC sind recht hoch, Implementierungen wie Ultra- oder TightVNC arbeiten aber mit Kompression und können damit auch über sehr geringe Bandbreiten (wie ISDN-Verbindungen) noch akzeptabel funktionieren.

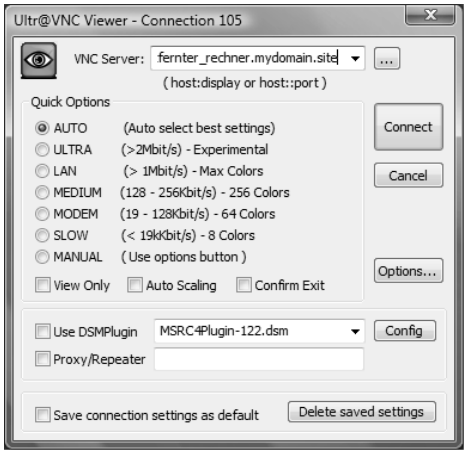

Abbildung 16.15: Konfiguration der Kompression des Windows-UVNC-Clients

Sind die Reaktionszeiten nicht akzeptabel, können Sie durch die Verwendung eines sehr kleinen Desktops (etwa 600x400 Punkte) und nur 8 Bit Farbtiefe die notwendige Bandbreite stark reduzieren. Als Voreinstellung benutzt der Tight-VNC-Viewer die Kompressions- und Übertragungstechniken, die mit `copyrect tight hextile zlib corre rre raw` benannt sind. Die letzte Methode ist eine Fallback-Option ohne jegliche Kompression. Nicht alle VNC-Server unterstützen alle Kodierungs- und Kompressionsarten. Eine weitere Option ist `-compresslevel N`, der von 0 bis 9 reicht. Dabei steht die 1 für schwache Kompression bei geringer CPU-Belastung und 9 für die stärkste bei hoher CPU-Last. Bei einem schnellen LAN lohnt es sich nicht, die CPUs der beteiligten Maschinen viel rechnen zu lassen. Für ISDN-Verbindungen hingegen opfert man lieber Rechenzeit, um einen flüssigeren Desktop-Aufbau zu erhalten. Ebenfalls möglich ist die Verbindung mit JPEG-Kompression. Dieses stellt `-quality N` ein. Level 0 erreicht beachtliche Kompression bei schlechterer Bildqualität. Mit 9 erreicht man nur noch schwache Kompression, aber eine gute Qualität der Darstellung. Die Option `-nojpeg` schaltet die JPEG-Kompression völlig ab. Die Hilfeseiten (`man vncviewer`) nennen weitere Optionen und erklären die verschiedenen Kodierungsverfahren.

16.5.2 X.11-VNC-Erweiterung

Der traditionelle VNC-Server unter Linux setzt auf seinen eigenen, lokal nicht sichtbaren X-Server. Das ist sinnvoll für die Remote-Administration von Linux-Servern oder für VNC-Login-Server, die irgendwo ohne Monitor im Rack montiert sind. Das gilt ebenfalls für Benutzer von Windows-Maschinen, die einen Linux-Desktop nicht missen wollen und sich diesen von einem Zweitrechner holen.

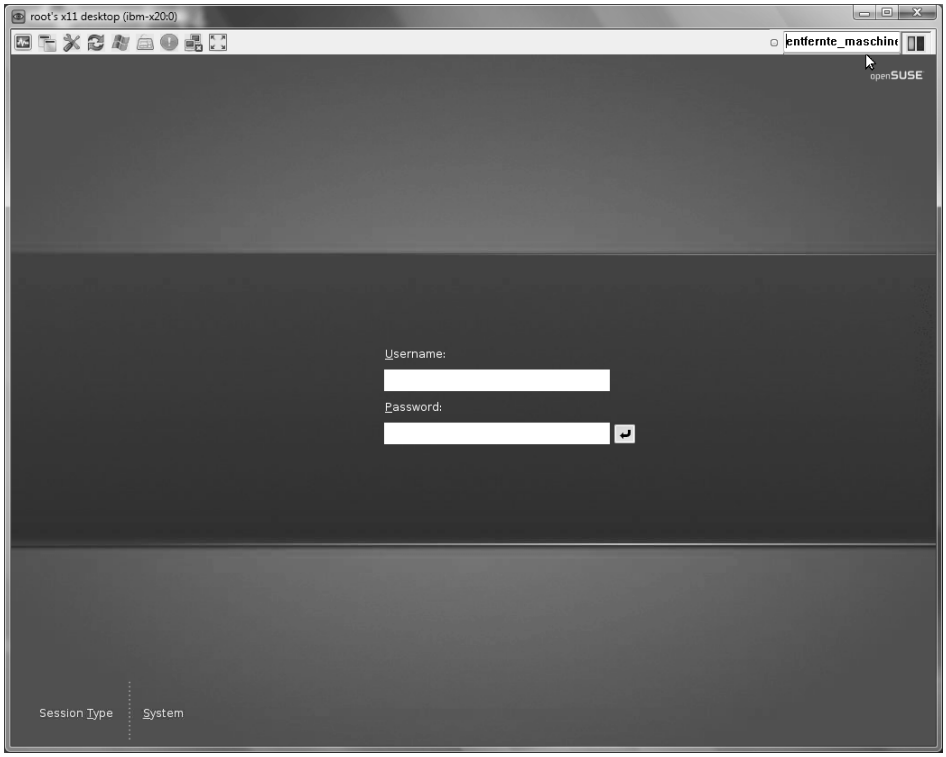

Abbildung 16.16: Zugriff von Windows auf den X.11-VNC-Server unter Linux

Wollen Sie aber PCs gemeinsam mit lokalen Benutzern administrieren oder diesen Benutzern helfen, brauchen Sie eine Möglichkeit, den laufenden Linux-Desktop zu klonen. Diese Fähigkeit von X.11 stellt das Modul `vnc.so` zur Verfügung. Sie passen hierzu die `/etc/X11/xorg.conf` an und tragen eine weitere Zeile in der Module-Section hinzu:

```
Section "Module"
  Load     "vnc"
  ...
Section "Device"
  # This tells X where to locate the VNC password file
```

```
  Option      "usevnc"
  Option      "rfbauth"     "/etc/X11/vncpasswd"
  ...
```

Der X-Server lädt so das zusätzliche Modul `vnc`. In der Device-Section sagen Sie dem VNC-Modul, dass es den VNC-Server einschalten soll und wo es seine Datei mit den Passwörtern für den entfernten Zugriff findet. Diese Datei füllen Sie mit `vncpasswd /etc/X11/vncpasswd`. Diese Erweiterung können Sie bei OpenSLX-Clients (lesen Sie hierzu Kapitel 17) durch das Plugin `x11vnc` bequem hinzufügen. Das Plugin richtet dann alle benötigten Komponenten ein.

16.5.3 VNC unter Windows

Für Windows sind verschiedene VNC-Lösungen im Angebot: Sehr gut ist das kostenlose UltraVNC (`http://www.uvnc.com`), das unter der GPL steht. Sie laden das Paket aus dem Bereich *Download* und installieren es anschließend per Doppelklick. Der Installer legt sowohl für den VNC-Server als auch den Client ein Desktop-Icon an, sofern Sie ihm dies nicht untersagen. Ein Doppelklick auf das Server-Icon startet einen Konfigurationsdialog, in dem Sie das Zugriffspasswort festlegen.

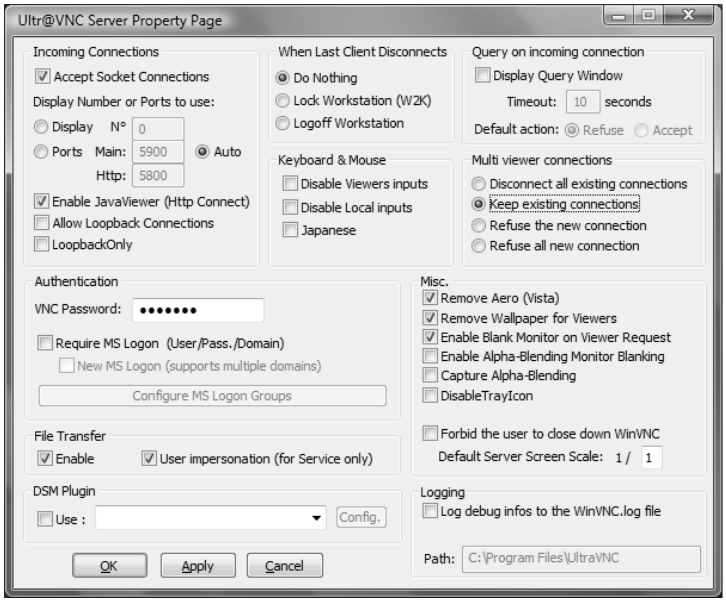

Abbildung 16.17: Legen Sie hier das Passwort für Ihren Windows-VNC-Server fest

Nun können Sie von einer anderen Maschine aus mit einem beliebigen VNC-Client auf den Windows-Desktop zugreifen. Sie sehen dann den Abzug des aktuellen Desktops.

Abbildung 16.18: Zugriff mit einem VNC-Client auf den Windows-Desktop

16.5.4 Client-Zugriffe

Für einen ersten Test des Client-Zugriffs unter Linux greifen Sie am besten lokal auf die soeben gestartete VNC-Sitzung zu: Dazu setzen Sie an der Konsole das Kommando `vncviewer localhost:1` ab, wobei Sie `:1` eventuell durch eine andere Display-Nummer ersetzen müssen. Dann erscheint nach korrekter Eingabe des gesetzten VNC-Passworts der neue Desktop in einem eigenen Fenster. Der VNC-Client steht zusätzlich als Java-Applet zur Verfügung, das im Browserfenster ablaufen kann. Sie können Ihren Browser auf eine URL `http://localhost:5800` richten und bekommen dann die Ausgabe wie im VNC-Viewer im Browser-Fenster angezeigt.

Im Gegensatz zu X11 kann VNC mehrere Clients auf einen Server zulassen. Am einfachsten erreichen Sie dieses, indem Sie den Server in der Betriebsart *always shared* starten. Diese geben Sie beim Kommandoaufruf an, da sonst automatisch ein anderer Client beim Start des neuen beendet werden würde: `vnc server -geometry 1200x1000 -depth 16 -alwaysshared`. Dieser Aufruf richtet eine Auflösung des Serverdesktops von 1200x1000 Bildpunkten ein sowie eine Farbtiefe von 16 Bit und die Möglichkeit, dass sich mehrere Clients gleichzeitig verbinden können. Soll sich immer nur ein Client verbinden, erreicht dieses die Option `nevershared`. Sie sehen hier, dass Sie sich an keine typische Bildschirmauflösung halten müssen, wenn Sie einen VNC-Server starten. Die gewählte Bildauflösung kann völlig flexibel gewählt werden. Wenn Sie auf dem Zielsystem mit dem `vncclient` zwei nebeneinander liegende Bildschirme von 1280x1024er Auflösung füllen wollen, können Sie den Server auch mit `-geometry 2400x1000` starten.

Sie können VNC zwar durch ein Passwort vor unberechtigtem Verbindungsaufbau schützen, jedoch überträgt VNC seine Daten unverschlüsselt. Da dies für die meisten Netze nicht akzeptabel ist, sollte man geschützte Tunnel verwenden.

Hierzu wird der VNC-Server mit der zusätzlichen Option `-localhost` gestartet, welche verhindert, dass Verbindungen von anderen Rechnern aufgebaut werden dürfen. Damit nun ein Zugriff von einem entfernten Rechner erfolgen kann, wird der VNC-Port über die Secure Shell (`ssh`) getunnelt. Hierzu benötigen Sie die Portnummer des VNC-Servers, wie im vorherigen Abschnitt erklärt. Der Verbindungsport auf Ihrer Seite darf dabei noch nicht belegt sein. Wenn der VNC-Server `entfernte_maschine` heißt, lautet das notwendige SSH-Kommando auf dem Client wie folgt:

```
ssh -l user -L 5901:localhost:5901 entfernte_maschine
vncserver :1 -localhost
```

SSH fragt Sie wie üblich nach dem Passwort des Benutzers (`-l user`). Dieser kann nach dem Login den VNC-Server starten. Anschließend starten Sie lokal den VNC-Viewer durch `vncviewer localhost:1`, der sich an das Tunnelende auf dem Client hängt. Der Viewer sieht dieses Tunnelende als lokal laufenden VNC-Server und spricht dazu Port 5901 an. Das Ganze geht auch einfacher durch den Aufruf von: `vncviewer -via [user@]entfernter_rechner :1`

```
lokale_maschine # vncviewer -via alkalde@entfernte_maschine :1
Password:
Connected to RFB server, using protocol version 3.7

Enabling TightVNC protocol extensions
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "alkalde's X desktop (linux:1)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8
blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8
blue 0
ShmCleanup called
Using shared memory PutImage
Tunneling active: preferring tight encoding
```

Hierbei sollten Sie jedoch beachten, dass erst die Nachfrage nach dem SSH-Passwort des Users und dann die Abfrage des VNC-Passwortes erfolgt. Alles Weitere verläuft dann wieder wie gewohnt, wobei die Performance wegen der zwischengeschalteten Verschlüsselung auf schwächeren Maschinen etwas niedriger liegen kann.